Multicast Tree Rearrangement to Recover Node Failures in Overlay Multicast Networks

Hee K. Cho and Chae Y. Lee
Dept. of Industrial Engineering, KAIST,
373-1 Kusung Dong, Taejon, Korea

Abstract

Overlay multicast makes use of the Internet as a low level infrastructure to provide multicast service to end hosts. The strategy of overlay multicast slides over most of the basic deployment issues associated with IP multicast, such as end-to-end reliability, flow and congestion control, and assignment of an unique address for each multicasting group.

Since each multicast member is responsible for forwarding multicast packets, overlay multicast protocols suffer from multicast node failures. To cope with node failures in the overlay multicast networks, the employment of multicast service nodes (MSNs) is considered which allows relatively high processing performance to cover the disconnected nodes. We are interested in minimizing the cost of both the MSNs and additional links when a node failure occurs.

Overlay multicast tree rearrangement to connect multicast members is discussed and formulated as a binary integer programming problem. The tree rearrangement problem is solved by a heuristic based on the Lagrangean relaxation. The performance of the proposed algorithm is investigated by carrying out experiments in 50 and 100 node problems. The employment of MSNs is illustrated to be dependent on the end-to-end delay bound in overlay networks and the degree constraint of member nodes.

1. Introduction

Multicasting provides an efficient way of transmitting data from a sender to a group of receivers. Instead of sending a separate copy of the data to each individual group member, a source node sends identical messages simultaneously to multiple destination nodes. An underlying multicast routing algorithm determines a multicast tree connecting the source and group members. Data generated by the source flows through the multicast tree, traversing each tree edge exactly once. As a result, multicast is more resource-efficient and is well suited to applications such as teleconferencing, video-on-demand (VOD) service, electronic newspapers, cyber education and medical images. However, despite the conceptual simplicity of IP multicast and its obvious benefits, its deployment is difficult due to the complexity of IP multicast technology and lack of applications.

To cope with the increasing traffic of multimedia contents, the study on multicasting will become more active. Recent efforts to provide multicast delivery have thus shifted to overlay multicast that builds a transport-layer overlay network among members of a multicast group. Recent topics related to overlay multicast includes [1, 2, 3, 4, 5, 6, 7, 8, 11] as an alternative to IP multicast. Overlay multicast uses the Internet as a low level infrastructure to provide multicast service to end hosts. Many basic deployment issues such as end-to-end reliability, congestion control, and assignment of an unique address for each multicasting group that are associated with IP multicast can be solved with overlay multicast.

Current overlay multicast projects can be classified into two catalogs according to the structure: end-to-end overlay [1, 3, 5] and proxy-based overlay [2, 4]. In end-to-end overlay, every member in the multicasting group shares the responsibility to forward data to other members. End hosts organize themselves into a multicasting tree. We call these end hosts multicast nodes in the end-to-end overlay case. Scattercast [2] and Overcast [4] are typical

examples of proxy-based overlay structure that form a hierarchical structure compared to end-to-end overlay. The multicasting service is performed with the help of proxy nodes, which can duplicate data and forward them to end hosts with predefined routing algorithm. In proxy-based overlay, proxy is the multicast node and end hosts just receive the multicast data from the corresponding proxies.

Representative research [7] on overlay multicast protocol includes Scattercast, Overcast, Narda [3] and ALMI [5]. Each protocol has different design objectives, which leads to different properties. Narda and Scattercast intend to minimize the delay from a multicast source to each member. ALMI strives to minimize the multicast tree cost, where the cost of each link is defined as the round-trip delay between group members. Overcast, on the other hand, maximizes available bandwidth for each member. In the tree construction process, Narda and Scattercast use a mesh-first approach. That is, group members are first connected into a mesh and then the multicast tree is built on top of it. On the other hand, Overcast and ALMI use a direct approach. The step to build the mesh is bypassed and the multicast tree is formed directly.

In overlay multicast, since multicast members are responsible for forwarding multicast packets, they suffer from multicast node failures. When a multicast node fails, rapid multicast tree recovery is essential to distribute multicast data to disconnected nodes. However, many researchers have focused their attention mainly on constructing the initial overlay multicast tree [1, 2, 3, 4, 5, 6, 8, 11]. In this paper, we are interested in solving the tree rearrangement problem by establishing new connections for the multicast members when a multicast node failure occurs.

2. Packet Delivery and Node Failures in Overlay Multicast Networks

Overlay multicast avoids the deployment hurdles of IP multicast at the cost of data delivery latency due to its inefficient multicast data distribution. This inefficiency is illustrated in the

example network of Figure 1. In Figure 1, nodes A, B, C, D, and E are members of a multicast group, nodes R1, R2, and R3 are routers in the network, and the dashed lines connecting the nodes are physical links. The flows in Figure 1 (a) illustrate how IP multicast forwards data sent by A to the other members of the group. Figure 1 (b) shows a possible overlay multicast traffic flow. Figure 1 (c) shows the overlay without the underlying physical network. Comparing Figure 1 (a) and (b), it is clear that data delivery using overlay multicast experiences longer delay than traditional IP multicast delivery. This example shows that latency between the source and the multicast members in an overlay multicast largely depends on the design of the overlay route. Thus, in this paper, we assume that end-to-end delay constraint is required to avoid excessive delay.

Note in Figure 1 (c) that each link in the overlay multicast tree represents an independent unicast session. Node C of Figure 1 (c) handles three unicast sessions. However, each multicast node has a limit in the number of unicast sessions that can be handled simultaneously. It is due to the CPU performance, network interface card capacity, buffer size and others. Thus we consider the degree constraint of a multicast node in the overlay multicast tree. The degree constraint represents the maximum number of unicast sessions that a multicast node can handle depending on the capacity.

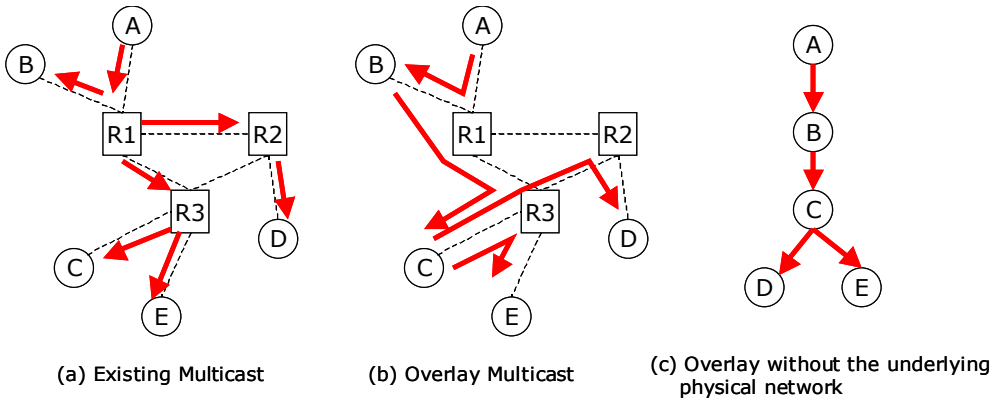


Figure 1. Comparison of packet delivery

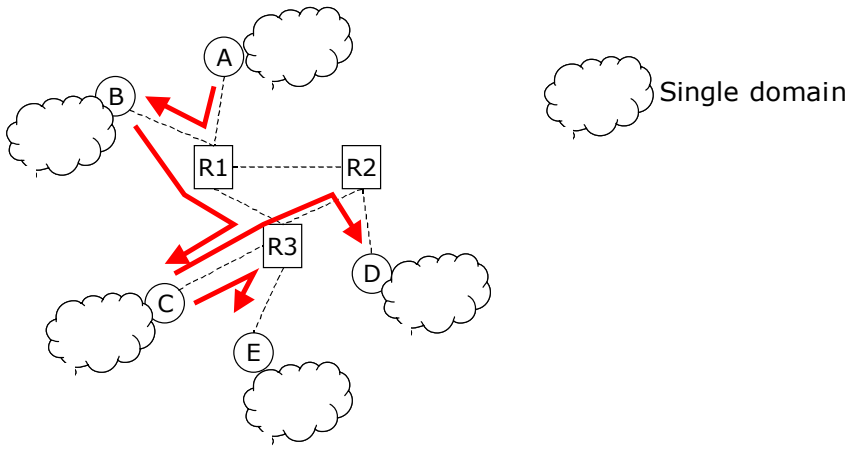


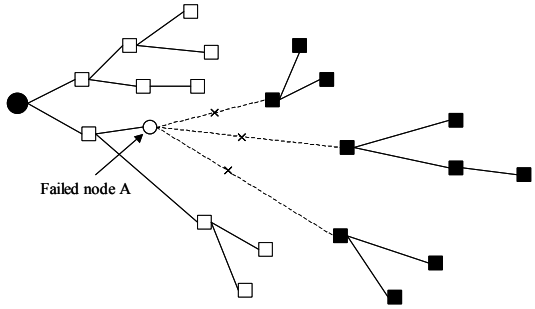
Figure 2. Overlay network for inter domain multicast

Overlay multicast can also be used in inter-domain multicast. Only a host from each domain receives the multicast data stream from the remote sender via the overlay multicast. The other receivers in the domain receive the application data stream from the host as illustrated in Figure 2. The domain in this paper represents a domain that is managed by a common multicast protocol.

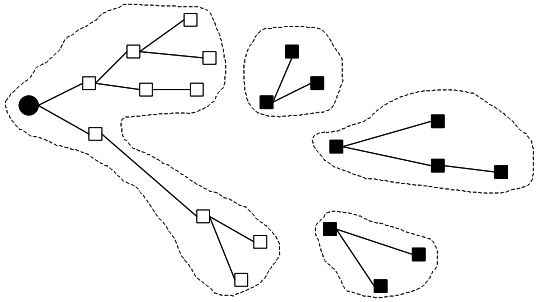
Now, consider a tree rearrangement in case of a node failure. In Figure 3, suppose that node A has failed in the overlay multicast tree. Thus, the children of node A have to rejoin the multicast session. The overlay multicast tree is divided into several fragments as shown in Figure 3 (b). Figure 3 (c) shows the case where the end-to-end delay constraint is not satisfied by some nodes after the multicast tree rearrangement, due to the degree bound of the node. The end-to-end delay can be reduced by the employment of a node with a higher degree in the overlay multicast tree [8]. Thus, to satisfy the end-to-end delay bound, we consider the employment of the multicast service nodes (MSNs) which allow relatively high processing performance by covering all the disconnected nodes.

The employment of an MSN requires additional links to connect the disconnected multicast nodes. The overlay multicast tree rearrangement, therefore, causes expenses for the MSNs and the links. For efficient tree rearrangement, special consideration is required in the selection of the MSNs and additional links. In this paper, we are interested in minimizing the cost of both

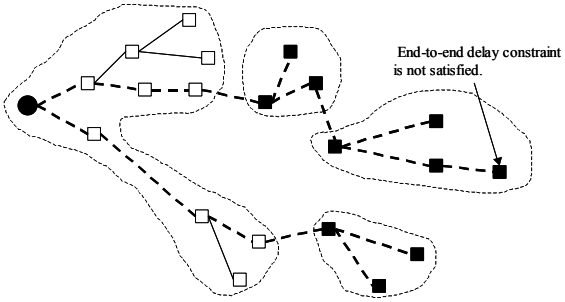
the MSNs and additional links when a node failure occurs.



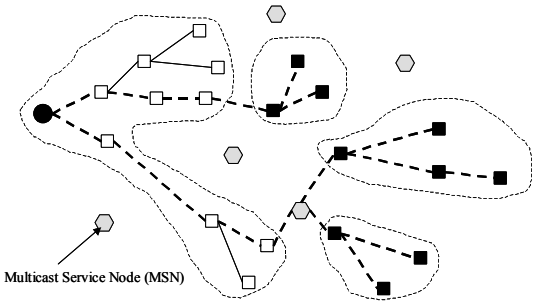
(a) Single multicast node failure



(b) Separation of multicast tree



(c) Tree rearrangement without the delay constraint



(d) Tree rearrangement with the delay constraint

Figure 3. Node failure and tree rearrangement

3. Multicast Tree Rearrangement in Overlay Multicast Networks

Given a graph $G = (V, E)$ where V is the set of nodes and E is the set of links, consider an overlay multicast tree, $T = (N, E1)$ where $N \subset V$ is the set of multicast nodes and $E1 \subset E$ is the set of links in T. Let $R \subset N$ be the set of multicast nodes to rejoin the multicast tree due to a node failure. The set of the black rectangle nodes in Figure 3 corresponds to R.

For a rejoin node $m \in R$, a single path is required to connect the node to the multicast source. If no path exists between the multicast source and node m, the tree rearrangement is impossible. However, by assuming at least one path between the multicast source and node m, the failed tree is rearranged to connect every multicast node. Let x_{ij}^{m} be a binary variable to represent a link (i,j) that is employed for a path between the source and rejoin multicast member m. If link (i,j) is employed to connect the source and node m, then $x_{ij}^{m} = 1$. Otherwise, $x_{ij}^{m} = 0$. The dotted lines of Figure 3 (c) and (d) represent links with $x_{ij}^{m} = 1$. Then the following relation holds for every node including the source node s.

$$\sum_{j \neq i} x_{ij}^{m} - \sum_{k \neq i} x_{ki}^{m} = 1 \qquad for\ i = s\ and\ all\ m \in R$$

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

$$\sum_{j \neq i} x_{ij}^{m} - \sum_{k \neq i} x_{ki}^{m} = -1 \qquad for\ i = m\ and\ all\ m \in R$$

In the above constraints, notice that several paths between the source and rejoin nodes may possibly traverse through link (i,j). Since the multicast tree rearrangement problem considers the cost of each additional link, we introduce a constant M to reflect the multiple new paths that share the same link. Let y_{ij} represent the adoption of link (i,j) for the new paths, then the following equation holds.

$$\sum_{m} x_{ij}^{m} + \sum_{m} x_{ji}^{m} \leq M y_{ij} \qquad for\ i < j$$

Note that the above equation holds for any link in the overlay multicast tree.

Now, as discussed in the previous section, data delivery in the overlay multicast tree is performed by independent unicast sessions. Because each multicast node including the MSN has a capacity limit in the number of simultaneous unicast sessions, we consider the degree constraint of a multicast node. The constraint represents the number of unicast sessions a multicast node can handle simultaneously. By letting w_i be the degree constraint of node i, we have

$$\sum_{j \neq i} y_{ij} + \sum_{k \neq i} y_{ki} \leq w_i z_i \qquad \text{for all } i$$

In addition to the above constraints, we need to consider the end-to-end delay between the source node and multicast members. In this problem, we assume that every multicast member has a delay bound. In other words, every path between the source and a rejoin node has to satisfy the delay bound constraint. By letting d_{ij} be the link delay and D be the end-to-end delay bound, we have

$$\sum_{(i,j) \in E} d_{ij} x_{ij}^{\ m} \leq D \qquad \text{for all } m \in R$$

Now, we need to minimize the cost of the newly installed MSNs and that of the newly included links. Let u_i and c_{ij} respectively be the cost of the newly installed nodes and links to rearrange the multicast tree. Then the total cost is given by $\sum_{i \in L} u_i z_i + \sum_{\substack{(i,j) \in E2 \\ i<j}} c_{ij} y_{ij}$ where $L \subset V$ is a set of candidate MSNs and $E2 \subset E$ is a set of links newly included into the overlay multicast tree. From the above discussion, the multicast tree rearrangement problem can now be formulated as follows.

Problem P:

Minimize $\sum_{\substack{(i,j) \in E2 \\ i<j}} c_{ij} y_{ij} + \sum_{i \in L} u_i z_i$

subject to:

$$\sum_{j \neq i} x_{ij}^{\ m} - \sum_{k \neq i} x_{ki}^{\ m} = 1 \qquad for\ i = s\ and\ all\ m \in R \qquad (1)$$

$$\sum_{j \neq i} x_{ij}^{\ m} - \sum_{k \neq i} x_{ki}^{\ m} = 0 \qquad for\ i \in V / \{m, s\}\ and\ all\ m \in R \qquad (2)$$

$$\sum_{j \neq i} x_{ij}^{\ m} - \sum_{k \neq i} x_{ki}^{\ m} = -1 \qquad for\ i = m\ and\ all\ m \in R \qquad (3)$$

$$\sum_{m} x_{ij}^{\ m} + \sum_{m} x_{ji}^{\ m} \leq M y_{ij} \qquad for\ i < j \qquad (4)$$

$$\sum_{j \neq i} y_{ij} + \sum_{k \neq i} y_{ki} \leq w_i z_i \qquad for\ all\ i \qquad (5)$$

$$\sum_{(i,j) \in E} d_{ij} x_{ij}^{\ m} \leq D \qquad for\ all\ m \in R \qquad (6)$$

$x_{ij}^{\ m}$, y_{ij}, z_i are binary variables

Considering the NP-hardness of the overlay multicast tree problem without the delay constraint [11], the proposed binary integer programming problem is NP-hard. In the above formulation constraints (1), (2) and (3) are well known constraints for the shortest path problem. Thus, relaxation of constraints (4), (5) and (6) may lead the problem to a less complicated version. Lagrangean relaxation is a general solution strategy for solving such a relaxed problem. The procedure permits us to decompose a problem into several easy subproblems by exploiting its special structure. This solution approach is used for solving many models with embedded network structure, such as the one considered in [10, 15]. In fact, the Lagrangean relaxation leads the overlay multicast tree reconstruction problem into three decomposed subproblems. The first subproblem is reduced into a simple shortest path problem where many sophisticated algorithms are available. The other two subproblems are reduced into unconstrained minimization problems that are easy to handle. Therefore, in the next section we thus develop a Lagrangean relaxation algorithm as a promising solution procedure for the overlay tree rearrangement problem.

4. Lagrangean Relaxation Based Heuristic for Tree Rearrangement

In this section, we consider the Lagrangean relaxation [9] to solve the problem formulated in Section 3. The relaxed problem is decomposed into three subproblems which can easily be solved by employing the known algorithms for the underlying network structures [10].

Consider the tree rearrangement problem P formulated in the previous section. The model is an integer programming problem which is, in general, difficult to solve. Rather than solving the difficult optimization problem directly, we combine the cumbersome constraints with the original objective function, where some multiplier values act as penalty factors. Then the problem as a whole is transformed to a more tractable form. The motivation for adopting this approach is based on the fact that the original problem P has an attractive substructure, the shortest path problem, which we would like to exploit algorithmically.

Let us discuss the Lagrangean relaxation procedure in more detail. By relaxing constraints (4), (5), and (6), the relaxed problem P_L is obtained as follows.

Problem P_L

Minimize $Z_L(\lambda)$ =

$$\sum_{m\in R}\Big(\sum_{(i,j)\in E}(\lambda_1(i,j)+\lambda_3(m)d_{ij})x_{ij}^{\,m}-\lambda_3(m)D\Big)+\sum_{\substack{(i,j)\in E2\\ i<j}}(c_{ij}-\lambda_1(i,j)M+\lambda_2(i)+\lambda_2(j))y_{ij}$$

$$+\sum_{i\in L}(u_i-w_i\lambda_2(i))z_i$$

Subject to

$$\sum_{j\neq i}x_{ij}^{\,m}-\sum_{k\neq i}x_{ki}^{\,m}=1 \qquad for\ i=s\ and\ all\ m\in R$$

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

$$\sum_{j\neq i}x_{ij}^{\,m}-\sum_{k\neq i}x_{ki}^{\,m}=-1 \qquad for\ i=m\ and\ all\ m\in R$$

$x_{ij}^{\,m}$, y_{ij}, z_i are binary variables

The Lagrangean multipliers λ_1, λ_2 and λ_3 correspond to the constraint sets (4), (5) and (6) respectively. Since $\lambda_1(i,j)$ is defined only for $i<j$, we let $\lambda_1(i,j) = \lambda_1(j,i)$ for computational convenience. Here, note that problem P_L can be decomposed into following three independent subproblems:

Subproblem P_{L1}

Minimize $\sum_{m\in R}\Big(\sum_{(i,j)\in E}(\lambda_1(i,j)+\lambda_3(m)d_{ij})x_{ij}^{\,m} - \lambda_3(m)D\Big)$

Subject to

$$\sum_{j\neq i} x_{ij}^{\,m} - \sum_{k\neq i} x_{ki}^{\,m} = 1 \qquad for\ i = s\ and\ all\ m \in R$$

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

$$\sum_{j\neq i} x_{ij}^{\,m} - \sum_{k\neq i} x_{ki}^{\,m} = -1 \qquad for\ i = m\ and\ all\ m \in R$$

$x_{ij}^{\,m}$ are binary variables

Subproblem P_{L2}

Minimize $\sum_{\substack{(i,j)\in E2\\ i<j}} (c_{ij} - \lambda_1(i,j)M + \lambda_2(i) + \lambda_2(j))y_{ij}$

Subproblem P_{L3}

Minimize $\sum_{i\in L}(u_i - w_i\lambda_2(i))z_i$

For a specific rejoin node m, subproblem P_{L1} is a shortest path problem. Subproblem P_{L2} and P_{L3} are unconstrained minimization problems. Now we present a systematic solution procedure for the decomposed subproblems.

(i) Solution to subproblem P_{L1}

The well-known Dijkstra's algorithm [9, 14] is used for fixed m in subproblem P_{L1}. The algorithm finds the shortest path from the source node to the rejoin multicast node m in a network. Dijkstra's algorithm solves the shortest path problem in $O(N^2)$ time. Since totally m shortest path problems have to be solved at every iteration, the computational complexity of subproblem P_{L1} is $O(mN^2)$.

(ii) Solution to subproblem P_{L2}

Subproblem P_{L2} is an unconstrained minimization problem. If the coefficient of y_{ij} is negative in the objective function, then $y_{ij} = 1$ minimizes the objective function value of subproblem P_{L2}. Otherwise, $y_{ij} = 0$. Thus, the decision criterion of y_{ij} is as follows.

Let $l_{ij} = (c_{ij} - \lambda_1(i,j)M + \lambda_2(i) + \lambda_2(j))$

$y_{ij} = 0$ if $l_{ij} \geq 0$

$y_{ij} = 1$ otherwise

(iii) Solution to subproblem P_{L3}

Subproblem P_{L3} is also an unconstrained minimization problem. Thus, the decision criterion of z_i is as follows.

Let $v_i = u_i - w_i\lambda_2(i)$

$Z_i = 0$ if $v_i \geq 0$

$z_i = 1$ otherwise

A solution to the Lagrangean relaxation problem P_L is obtained by solving the above subproblems. However, the solution to P_L is usually not a feasible solution to the original problem P. Thus, a feasible solution is obtained at each iteration as follows. The solution acquired in Lagrangean relaxation problem P_L is transformed into tree form through Step 1 and Step 2. Step 3 makes the above solution satisfy the delay and degree constraints by

reconstructing the feasible path between the source and rejoin node m. The solution in Step 3 is further improved in Step 4. Let $p(m)$ be an existing path between the multicast source and rejoin node m, employed for the current multicast tree, and let $p'(m)$ be a new path between the multicast source and rejoin node m. In Step 4, if $p'(m)$ has lower cost than $p(m)$, $p(m)$ is removed from the multicast tree and $p'(m)$ is added to the multicast tree. The heuristic for generating feasible solutions is as follows.

Step 1. $y_{ij} = 0$ for all i, j.

$z_i = 0$ for all i.

Step 2. If $\Sigma_m x_{ij} \geq 1$, $y_{ij} = 1$.

If $(y_{ij} = 1)$ and $(i \in P)$, $z_i = 1$.

If $(y_{ij} = 1)$ and $(j \in P)$, $z_j = 1$.

Step 3. Check feasibility for each node m.

If the solution is feasible for all node m, then Goto Step 4.

Otherwise, update the network for feasible node m

and compute Dijkstra's algorithm to recompute x_{ij}^m for infeasible node m.

Goto Step 1.

Step 4. If termination criterion is satisfied,

While all rejoin nodes are not searched,

Select a rejoin node m that is not searched.

Find a $p'(m)$.

If $p'(m)$ has lower cost than $p(m)$,

$p(m)$ is removed from the current solution.

$p'(m)$ is added to the current solution.

Stop.

Otherwise, Stop.

According to the Lagrangean bounding principle [9] we can calculate the maximum allowable error range for each of the feasible solutions obtained. For any vector λ of the Lagrangean multipliers, the objective function value $Z_L(\lambda)$ of the Lagrangean dual function is a lower bound to the optimal objective function value Z_P* of the original optimization problem P. Thus, we have

$Z_L(\lambda) \leq Z_P$*.

The bounding principle immediately implies one advantage of the Lagrangean relaxation approach. The method gives us a useful termination criterion involving maximum allowable error range. At each iteration, Lagrangean relaxation problem P_L is solved by using updated multiplier value λ. A feasible solution to the problem P is then generated via the proposed heuristic from the solution of the relaxed problem. Let Z_P be the objective function value of problem P by the feasible solution. Then the maximum allowable error range ε is defined as

$$\varepsilon = \frac{Z_P - Z_L(\lambda)}{Z_L(\lambda)} \times 100\%$$

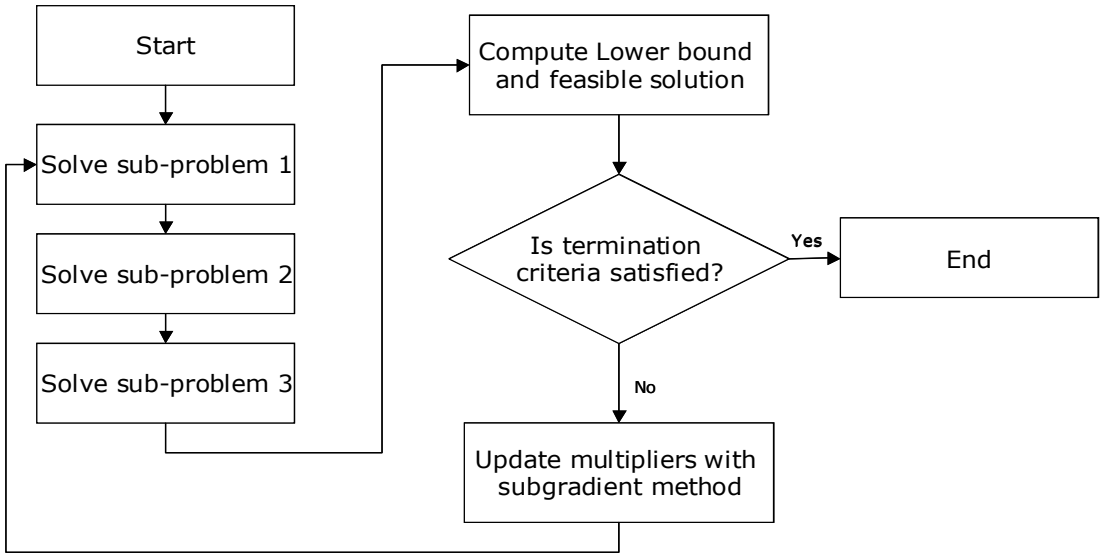


Figure 4. Procedure of the proposed heuristic

Lagrangean relaxation problem P_L is repeatedly solved by using updated multiplier values until the maximum allowable error range ε satisfies a termination threshold or the number of iterations exceeds the threshold limit. We use the subgradient method [10, 13] for updating Lagrangean multipliers. Figure 4 shows the overall procedure of the proposed algorithm.

5. Computational Results

In this section, we discuss the computational experiment of the Lagrangean relaxation based heuristic for the tree rearrangement problem. Multicast networks with 50 and 100 nodes are generated. In each network, links are randomly generated such that the total number of links becomes $|E| = 5|V|$. The average link cost is assumed to be 10. Each link has average delay value of three. The end-to-end delay bound has the value of 10 and 20 in the network with 50 nodes and 15 and 25 in the network of 100 nodes. The average cost of MSN is assumed to be 20 and degree constraint has the value of 3 and 8 in the network with 50 nodes. These values are 5 and 15 in the network of 100 nodes. All solution procedures are run on a Pentium III-660MHz PC.

Table 1 and 2 respectively show the result of the proposed algorithm in 50 and 100 nodes multicast networks. The CPLEX [12] is employed to have optimal solutions. The proposed Lagrangean heuristic shows good performance in every problem category of delay bound and degree constraints. The solution gap from the optimal solution is slightly higher in problems with lower delay bound and degree constraint. As shown in the table, the average solution gap from the optimal solution is 5% and 5.6% respectively in the 50 and 100 node problems. The CPU seconds are also compared as shown in the tables. The computational effort required by the proposed algorithm is dramatically reduced when compared to the exact solution procedure.

The computational result is further investigated by examining the required number of MSNs,

due to the different values of delay bounds and degree constraints. Figure 5 shows that more MSNs are employed as the problem becomes more difficult. As the delay bound and degree constraint becomes tighter, the employed number of MSNs increases for the tree rearrangement. The increase seems to be more sensitive to the delay bound when the degree constraint is tight.

Further experiments are performed to examine the cost sensitivity of the overlay tree rearrangement. Figure 6 shows the number of employed MSNs when different unit MSN cost is applieds. It is clear that more MSNs are employed as the number of rejoining multicast nodes increases. The increase of the required MSNs is also sensitive to the unit cost. Higher unit cost constraints the number of MSNs to cover the rejoin hosts.

6. Conclusion

To overcome node failures in overlay multicast networks, multicast tree rearrangement is considered by employing the multicast service nodes (MSNs). The service nodes generate new paths from a source to multicast members such that the end-to-end delay bound is satisfied. Since each multicast member is responsible for forwarding the packets in the overlay network, the degree constraint of a multicast node is also considered.

The tree rearrangement problem is formulated as a binary integer linear program that minimizes the cost of MSNs and newly adopted links. Lagrangean relaxation based heuristic is developed to solve the problem. Some intractable constraints are relaxed and added into the objective as a penalty. Then the problem is decomposed into three subproblems that are easy to handle.

Networks with 50 and 100 nodes are generated for computational experiments. In each network, four categories of problems are tested with different delay bounds and degree constraints. The average solution gap from the optimal solution is 5% and 5.6% respectively in

the 50 and 100 node problems. The computational time required by the proposed algorithm is dramatically reduced compared to the exact solution procedure. The required number of MSNs due to the different values of delay bound and degree constraint is also investigated. As the problem becomes tighter, the employed number of MSNs increases to recover the node failure. The increase is more sensitive to the delay bound when the degree constraint is tighter. The cost of MSN also affects the employment of the nodes.

Problem Number	(Delay bound, Degree constraint)	Heuristic Solution (HS)	CPU seconds	Optimal value (OPT)	Optimal CPU seconds	GAP*
1	(10,3)	132	3.5	125	55.9	0.056
2	(10,3)	124	4.2	119	52.5	0.042
3	(10,3)	139	4.5	131	65.7	0.061
4	(10,3)	123	4.0	117	58.6	0.051
5	(10,3)	113	3.4	108	58.1	0.046
6	(10,8)	121	3.4	114	58.6	0.061
7	(10,8)	120	4.2	115	50.9	0.043
8	(10,8)	109	4.4	103	66.4	0.058
9	(10,8)	122	4.0	114	60.4	0.070
10	(10,8)	108	3.5	102	59.6	0.059
11	(20,3)	119	3.6	113	56.3	0.053
12	(20,3)	117	3.9	114	49.8	0.026
13	(20,3)	105	4.5	101	64.1	0.040
14	(20,3)	115	4.1	108	58.9	0.065
15	(20,3)	105	3.6	101	57.6	0.040
16	(20,8)	115	3.5	108	56.4	0.065
17	(20,8)	106	3.7	103	50.8	0.029
18	(20,8)	103	4.6	99	60.8	0.040
19	(20,8)	112	4.1	107	60.6	0.047
20	(20,8)	102	3.6	97	58.0	0.052

*GAP = (HS-OPT)/OPT

Table 1. Performance of the proposed heuristic with 50 nodes

Problem Number	(Delay bound, Degree constraint)	Heuristic Solution (HS)	CPU seconds	Optimal value (OPT)	Optimal CPU seconds	GAP*
1	(15,5)	201	22.1	193	375.8	0.041
2	(15,5)	195	21.8	182	388.0	0.071
3	(15,5)	212	24.5	197	335.7	0.076
4	(15,5)	208	23.7	199	344.0	0.045
5	(15,5)	216	25.5	202	365.8	0.069
6	(15,15)	187	22.3	174	428.6	0.075
7	(15,15)	192	22.1	181	411.6	0.061
8	(15,15)	194	25.5	181	353.4	0.072
9	(15,15)	162	22.8	152	340.1	0.066
10	(15,15)	193	26.5	184	388.4	0.049
11	(25,5)	171	21.9	165	353.9	0.036
12	(25,5)	188	22.0	179	391.9	0.050
13	(25,5)	190	24.0	177	343.5	0.073
14	(25,5)	161	23.9	151	345.7	0.066
15	(25,5)	164	25.4	157	353.8	0.045
16	(25,15)	169	22.0	160	365.0	0.056
17	(25,15)	185	22.2	177	424.9	0.045
18	(25,15)	171	24.3	165	344.7	0.036
19	(25,15)	156	21.9	149	342.4	0.047
20	(25,15)	161	26.3	155	376.3	0.039

*GAP = (HS-OPT)/OPT

Table 2. Performance of the proposed heuristic with 100 nodes

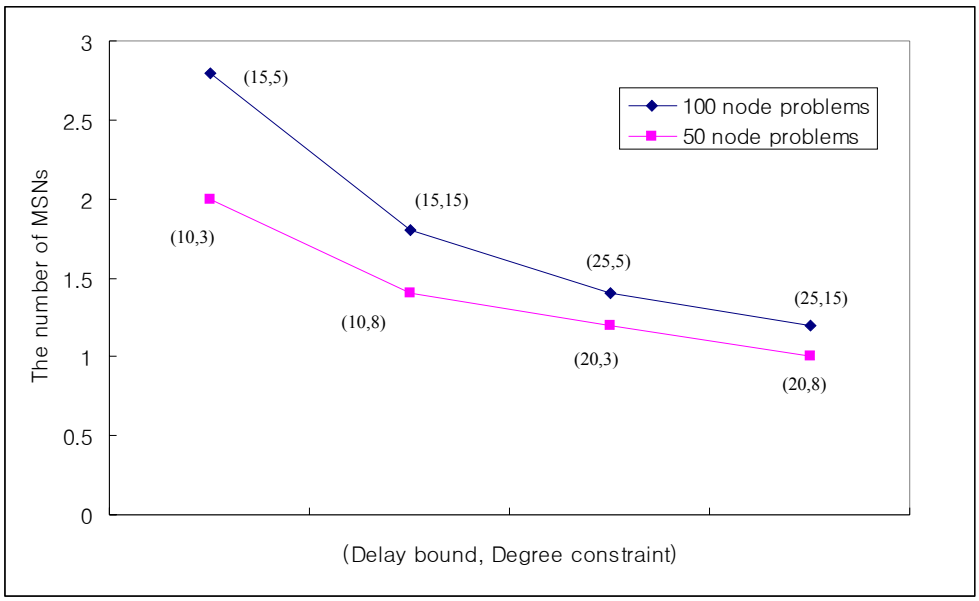


Figure 5. The number of MSNs employed for the tree rearrangement

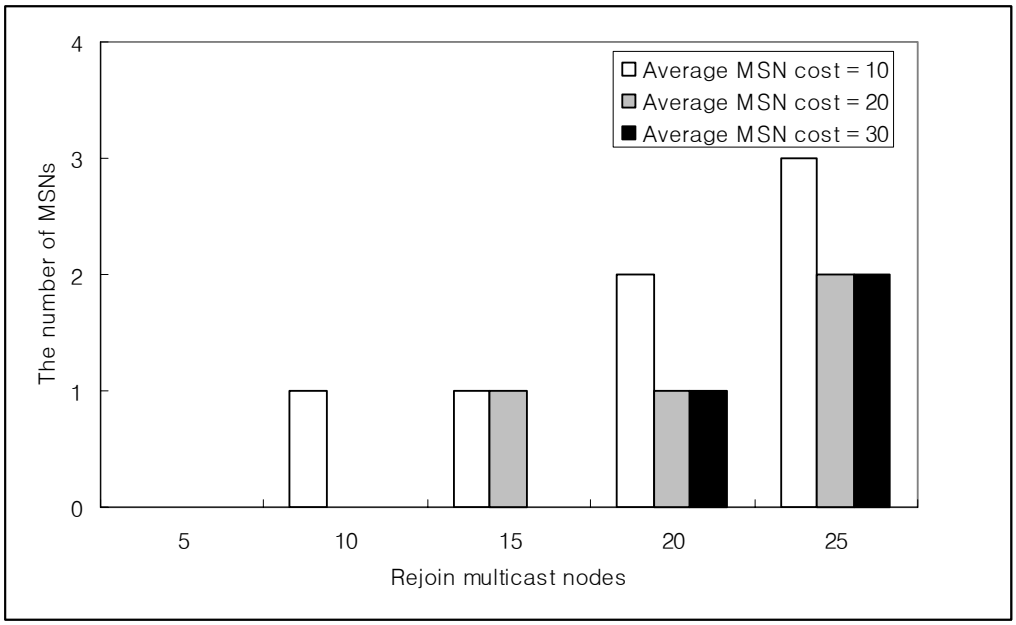


Figure 6. The number of MSNs for the tree rearrangement

References

[1] P. Francis. “Yoid: Your own Internet distribution,” UC Berkeley ACIRI Tech Report, Apr. 2000.

[2] Y. Chawathe. “Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service,” Ph. D. thesis, Department of EECS, UC Berkeley, Dec. 2000.

[3] Y. Chu, S. Rao, and H. Zhang. “A case for end system multicast,” ACM Sigmetrics, 2000.

[4] 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.

[5] D. Tellium. “ALMI: An Application Level Multicast Infrastructure,” 3rd Usenix Symposium on Internet Technologies and Systems, March, 2001.

[6] D. Helder and S. Jamin. “Banana Tree Protocol, an End-host Multicast Protocol,” Tech Report, July 2000.

[7] Y. Zhu, M. Wu, and W. Shu. “Comparison Study and Evaluation of overlay Multicast Networks,” ICME, 2003.

[8] 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.

[9] R. K. Ahuja, T. L. Magnanti and J. B. Orlin. “Network Flows: Theory, Algorithms, and Applications,” Prenticehall, 1993.

[10] H. Choi, H. Lee, K. Lee, S. Kim, and J. Lee. "Optimal location of switches and interconnections for ATM LANs," Computers & Operations Research, v. 28, pp1347-1366, Nov. 2001.

[11] S. Shi and J. Turner. “Routing in Overlay Multicast Networks,” IEEE INFOCOM, 2002.

[12] CPLEX 7.0, CPLEX Optimization Inc. 2002.

[13] M. Held, P. Wolfe and H. P. Crowder. “Validation of Subgradient Optimization,” *Mathematical Programming*, vol. 6, pp 62-88, 1974.

[14] D. Bertsekas and R. Gallager. “Data Networks,” Englewood Cliffs, NJ:Prentice Hall, 1987.

[15] L. Bahiense, F. Barahona, and O. Porto. “Solving Steiner Tree Problems in Graphs with Lagrangian Relaxation,” Journal of combinatorial optimization, v.7 no.3, pp.259-282, 2003.