Fast and Optimal Multicast-Server Selection Based on Receivers' Preference

Akihito Hiromori[1], Hirozumi Yamaguchi[1], Keiichi Yasumoto[2], Teruo Higashino[1], and Kenichi Taniguchi[1]

[1] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyamacho, Toyonaka, Osaka 560-8531, JAPAN
hiromori,h-yamagu,higashino,taniguchi}@ics.es.osaka-u.ac.jp
[2] Faculty of Economics, Shiga University
1-1-1 Bamba, Hikone, Shiga 522-8522, JAPAN
yasumoto@biwako.shiga-u.ac.jp

Abstract. In this paper, we propose static and dynamic server selection techniques for multicast receivers who receive multiple streams from replicated servers. In the proposed static server selection technique, if (a) the location of servers and receivers and shortest paths between them on a network and (b) each receiver's preference value for each content are given, the optimal server for each content that each receiver receives is decided so that the total sum of the preference values of the receivers is maximized. We use the integer linear programming (ILP) technique to make a decision. When we apply the static server selection technique for each new join/leave request to a multicast group issued by a receiver, it may cause server switchings at existing receivers and may take much time. In such a case, it is desirable to reduce both the number of server switchings and calculation time. Therefore, in the proposed dynamic server selection technique, the optimal server for each content that each receiver receives is also decided so that the total sum of the preference values is maximized, reducing the number of server switchings, by limiting both the number of receivers who may switch servers and the number of their alternative servers. Such restrictions also contribute fast calculation in ILP problems. Through simulations, we have confirmed that our dynamic server selection technique achieves less than 10 % in calculation time, more than 90 % in the total sum of preference values, and less than 5 % in the number of switchings on large-scale hierarchical networks (100 nodes), compared with the static server selection.

1 Introduction

Multicast is a useful way for saving bandwidth consumption by simultaneous transmission of a data stream such as WWW pushing of contents and live video streaming to multiple receivers [1, 2]. However, due to the limited bandwidth that can be used for multicast traffics, when multiple streams of live video are transfered, we need efficient bandwidth control of network resources used by each stream. For this purpose, we have proposed bandwidth control techniques to maximize the quality requirements of receivers for unicast and multicast streams [3, 4].

Regardless of unicast or multicast communication, bandwidth shortage caused by multiple streams is due mainly to path length between servers and receivers, since competition occurs among different streams at some common bottle-neck links. To overcome this problem, it may be useful to place some replicated servers at remote nodes [5] where video sources are transmitted to the replicated servers through high-speed links (backbone), and each receiver selects one of these servers depending on network traffics, server loads and so on. This technique improves network utilization without changing underlying routing protocols.

In recent years, such multi-server techniques have been researched [5–7]. [6, 7] have proposed unicast server selection techniques based on metric information such as packet delay, hop count and server load. [5] has proposed a multicast server selection technique where the optimal server assignment for each receiver to minimize the total link cost is formulated as a mathematical problem on a graph. [5] also gives a heuristic for the dynamic server selection problem where the server switching cost caused by join/leave requests to multicast groups is considered. However, in multi-media applications using multicast communication such as video-conferences at multiple locations, each receiver requires to receive more than one stream and his/her preference for each stream may differ from others. In general, when a receiver receives a stream which other receivers would not like to receive and their path from the server is quite long, the bandwidth used by the stream may reduce the benefit of the whole receivers. Therefore, on networks where available bandwidth is limited, it is desirable to consider each user's preference for each stream and to maximize the benefit of the whole receivers.

Such optimization can statically be calculated if the set of receivers and their preferences to all streams are known in advance. However, in general, join/leave requests are repeatedly issued by receivers. In such a case, re-optimization should be done dynamically. If we do such optimization for every request, the following problems arise.

1. Calculation time: the problem to select servers is a combinatorial optimization problem. Therefore, large amount of calculation time may be required in large-scale networks (we show calculation time against the number of nodes in Section 5).
2. Switching frequency: optimization may force existing receivers to switch the current servers of its receiving streams to others even if they do not want overhead caused by multicast join/leave requests (join/leave latency and so on).

Therefore, it is desirable to apply the optimization technique to a part of network where such dynamic changes happen, reducing the number of server switchings at receivers as well as keeping the sum of the preference values higher than a reasonable threshold.

In this paper, we propose static and dynamic server selection techniques for multicast streams transfered from multiple replicated servers. In the proposed static server selection technique, if several replicated servers and each receiver's preference value for each content are given, the optimal server for each content that each receiver receives is decided so that the total sum of the satisfied preference values is maximized. In the proposed dynamic server selection technique, when a new join/leave request to a multicast group is issued by a receiver, the number of server switchings at existing receivers should be reduced, and the total sum of the satisfied preference values should be increased. Therefore, in our dynamic server selection technique, we limit the number of receivers who may switch servers and their alternative servers so that the number of server switchings is drastically reduced and the sum of preference values is increased.

In our static optimization technique, in networks with certain link capacities, from given (1) the location of servers and receivers, (2) shortest paths between them and (3) each receiver's preference value to each stream, we construct a logical conjunction of linear inequalities which represent the bandwidth constraint on each link used by streams. The objective function is set to maximize the total sum of preference values of all receivers to streams. Thus, we use the integer linear programming (ILP) technique to make an optimal server selection. Here, we assume that streams with lower preference values may not be received in case of bandwidth shortage.

In our dynamic optimization technique, for each join/leave request issued by a receiver, we also construct linear inequalities, in order to obtain a solution where the sum of the preference values is maximized, reducing the total number of server switchings. Here, for fast calculation in the target ILP problem, we add some constraints to restrict receivers to ones who may suffer server switchings and also restrict their alternative servers to the two servers whose multicast trees are the closest of all the servers from the requested receiver.

We have simulated our static and dynamic server selection techniques and measured (a) calculation time, (b) the sum of preference values, and (c) the number of server switchings in a random topology and a hierarchical Internet topology called Tiers topology [8] consisting of LAN, MAN and WAN, where SPF (Shortest Path First) routing protocol is supposed. As a result, we have confirmed that our dynamic server selection technique achieves less than 10% in calculation time, more than 90 % in the sum of preference values, and less than 5 % in the number of server switchings, compared with the static server selection.

2 Preliminaries

A network is modeled as an undirected graph with capacity $CAP(e)$ for each link e. Replicated multicast servers (or just servers hereafter) $S = \{s_1, ..., s_m\}$ and receivers $R = \{r_1, ..., r_n\}$ exist on the network. Each server forwards contents $C = \{c_1, ..., c_p\}$ sent from source nodes to the receivers. Therefore, from the receivers, each server can be regarded as a multicast server which has these contents. Each receiver can receive each content from one of these servers, and specifies a value called a *preference value* to each content. A preference value means how eagerly the receiver wants to receive the content. An example of the network model is shown in Fig. 1.

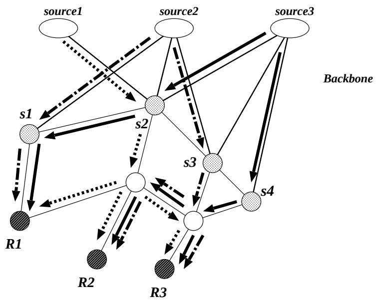

Fig. 1. Network with Replicated Multicast Servers

In Fig. 1, *source*1, *source*2 and *source*3 are the source nodes of contents c_1, c_2 and c_3, respectively. These contents are delivered to some of the replicated servers s_1, s_2, s_3 and s_4 through the connections with large capacities. s_1, s_2, s_3 and s_4 are the candidates of multicast servers, and each receiver selects one of them to receive each content. In the figure, R_1 receives c_2 from s_1, and both of c_1 and c_3 from s_2.

In this paper, for given (1) location of servers and receivers, (2) the shortest path between each pair of server s_i and receiver r_j and (3) the preference value of each receiver to each content, we formulate a problem to decide a server for each pair of a receiver and a content so that the total sum of satisfied preference values is maximized. We call this problem *static server selection problem*. Note that each receiver may not be able to receive all the contents that she/he required, due to bandwidth constraints.

Then we consider the case that receivers dynamically start or stop receiving contents, that is, receivers join/leave multicast groups. For such a case, we can optimize the total sum of preference values by solving the static server selection problem when every join/leave request is issued. However, we cannot avoid suffering (a) the exponential growth of computation time and (b) the overhead of server switching at almost all receivers. Regarding (a), we have experienced simulation on networks with 100 nodes, 50 receivers, 10 servers and 5 contents, and it took 200 seconds in average and more than 400 seconds in the worst case to get an optimal solution on an average machine (Pentium III, 500MHz). Such durations are allowed if, for example, we design the total layout of multicast trees before a new continuous service is started, however, not feasible for each small change of receiver status. Regarding (b), if we consider that the overhead increases proportional to the number of server switchings, it is much better to reduce it. For these reasons, in this paper, we propose another optimization technique called *dynamic server selection* for each join or leave request of a receiver. In Sections 3 and 4, we describe the static and dynamic server selection problems, respectively.

Let us define the following terms.

– $path(s_i, r_j)$: the set of links on the shortest path from s_i to r_j
– $uplink(e_l, s_i, r_j)$: the up-link (next to e_l) on $path(s_i, r_j)$

- $endlink(s_i, r_j)$: the bottom-link (attached to r_j) on $path(s_i, r_j)$
- $pref(r_j, c_k)$: the preference value of r_j given to c_k
- $bw(c_k)$: the transmission rate of c_k
- (s_i, c_k): the multicast group where c_k is delivered from s_i

3 Static Server Selection Problem

In order to formulate the static server selection problem, we define the following two types of boolean variables. Each variable in one type represents the fact that a receiver can receive a content from a server. Each variable in another type represents the fact that the multicast tree of a content from a server uses a link.

- $rcv[s_i, r_j, c_k]$: its value is one only if receiver r_j receives content c_k from s_i, otherwise zero.
- $deliver[e_l, s_i, c_k]$: its value is one only if content c_k from server s_i is delivered through link e_l, otherwise zero.

Using these variables, the static server selection problem can be formulated as the following integer linear programming (ILP) problem.

$$\max \sum_i \sum_j \sum_k pref(r_j, c_k) \cdot rcv[s_i, r_j, c_k] \tag{1}$$

subject to:

$$\sum_i rcv[s_i, r_j, c_k] \leq 1, \quad \forall j,\ k \tag{2}$$

$$deliver[e_l, s_i, c_k] \leq deliver[uplink(e_l, s_i, r_j), s_i, c_k], \quad \forall i,\ j,\ k,\ l \tag{3}$$

$$rcv[s_i, r_j, c_k] \leq deliver[endlink(s_i, r_j), s_i, c_k], \quad \forall i,\ j,\ k \tag{4}$$

$$\sum_i \sum_k bw(c_k) \cdot deliver[e_l, s_i, c_k] \leq CAP(e_l), \quad \forall l \tag{5}$$

Objective function (1) represents the total sum of all receivers' preference values. Constraint (2) states that one receiver selects at most one server for each content. Constraints (3) and (4) concern the form of multicast trees and indicate that if r_j receives c_k from s_i, the multicast tree of c_k from s_i must contain the shortest path from s_i to r_j. Constraint (5) is a bandwidth constraint on each link.

4 Dynamic Server Selection Problem

Due to the dynamic behavior of receivers in multicast communication, members in a group are not unique throughout a session. Therefore, fast re-optimization for each join/leave request of a receiver is desirable.

Let $rcv'[s_i, r_j, c_k]$ denote the fact that receiver r_j is currently joining the group (s_i,c_k) (its value is one if r_j is joining the group, zero otherwise). The join/leave behavior of a receiver is described as follows.

- **join**: a receiver r_q where $\sum_i rcv'[s_i, r_q, c_k] = 0$ wants to join one of the groups (s_1, c_k), (s_2, c_k), ... and (s_m, c_k).
- **leave**: a receiver r_q where $rcv'[s_i, r_q, c_k] = 1$ wants to leave the group (s_i, c_k).

We limit the number of receivers who are forced to switch their servers to others, in order to prevent a receiver's join/leave behavior from affecting all the receivers spread in wide-area networks. We also limit the number of possible alternative servers (servers to be switched) when a receiver switches its server of a content so that the receiver does not select servers far from him/her. On assuming such restrictions, we formulate the dynamic server selection problem.

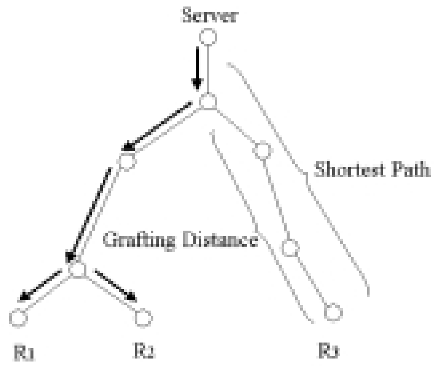

Fig. 2. Grafting Distance

[join] We define a *grafting distance* as the number of links on the shortest path from a server to a receiver through which a content c_k from the server has not been delivered yet (*i.e.*, the number of links where the content would be started to deliver when the receiver joins the group). An example is shown in Fig. 2. We adopt the following policy.

- receiver r_q selects one of the two servers (say s_{i_1} and s_{i_2}) with the shortest two grafting distances.
- For each $path(s_{i'}, r_{j'})$ which shares some of links with $path(s_{i_1}, r_q)$ or $path(s_{i_2}, r_q)$, if receiver $r_{j'}$ is receiving content c_k from server $s_{i'}$, $r_{j'}$ is one of the receivers who may switch servers. As an alternative server to receive c_k, $r_{j'}$ may select one of the two servers with the shortest two grafting distances.

Intuitively, each receiver whose receiving streams may compete with r_q's new stream may have to switch servers. Furthermore, we limit the number of their possible alternative servers to only two for each pair of a receiver and a content.

The dynamic server selection problem by each receiver's join request is an ILP problem with the same objective function (1), the same constraints (2)–(5) as in Section 3 and the following constraint to fix the status of receivers who should not switch servers:

$$rcv[s_i, r_j, c_k] = 1, \quad \forall (i, j, k) \neq (i', j', k') \tag{6}$$

where (i', j', k') is a tuple satisfying the following constraint. Note that all the paths are given and $rcv'[s_i, r_j, c_k]$ in the constraint have already been decided. Therefore such tuples (i', j', k') are uniquely determined.

$$(path(s_{i_1}, r_q) \cap path(s_{i'}, r_{j'}) \neq \emptyset \vee path(s_{i_2}, r_q) \cap path(s_{i'}, r_{j'}) \neq \emptyset) \\ \wedge rcv'[s_{i'}, r_{j'}, c_{k'}] = 1$$

[leave] We adopt the following switching policy when r_q leaves group (s_i, c_k).

– For each $path(s_{i'}, r_{j'})$ which shares some of links with $path(s_i, r_q)$, $r_{j'}$ can select $s_{i'}$ as the server to receive content c_k which $r_{j'}$ has not received.

The dynamic server selection problem by each receiver's leave request is an ILP problem with the same objective function (1), the same constraints (2)–(5) and the following constraint to fix the status of receivers who should continue to receive streams.

$$rcv[s_i, r_j, c_k] = 1, \quad \text{if } rcv'[s_i, r_j, c_k] = 1 \tag{7}$$

In Section 5, we have measured the performance of dynamic server selection for a receiver's join request compared with the static server selection, in terms of the computation time, the total sum of satisfied preference values and the total number of server switchings.

5 Simulation

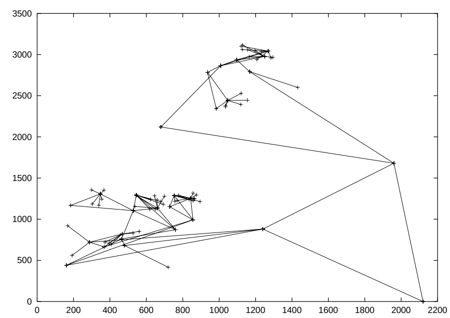

Fig. 3. Network Topology (a) Tiers Model

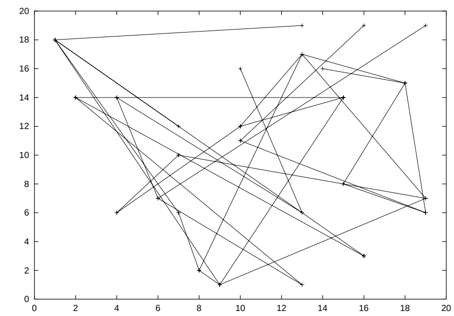

Fig. 4. Network Topology (b) Random Model

We have used two types of networks based on (a) Tiers Model [8] (Fig. 3) and (b) Random model (Fig. 4). Tiers is a hierarchical model organized by three domains, LAN, MAN, and WAN. For Tiers model, we randomly decided the number of nodes contained in LAN, MAN and WAN. We also decided the link capacities of LAN, MAN and WAN so that they are in the ratio of 1 : 10 : 100. Then we simulated 5 times on the networks varying the number of nodes $|N|$. For Random model, we randomly decided each link capacity based on Gaussian distribution, and simulated 20 times on the networks varying $|N|$. Also, we had $|S| = 0.1|N|$ servers, $|R| = 0.5|N|$ receivers and $|C| = 5$ contents in the networks and selected receivers' preference values from 25, 16, 9, 4 and 1, randomly. We simulated the dynamic server selection for a receiver $r_{|R|}$ who tried to join a group in the situation that the server selection had been already optimized for the receivers $r_1, ..., r_{|R|-1}$ by the static server selection (this simulation is denoted by (d)), and the static server selection for the receivers $r_1, ..., r_{|R|}$ (denoted by (s)). Then we have measured the calculation time, the total sums of satisfied preference values and the numbers of server switchings of (d) and (s). The results are shown in Section 5.1. Also in order to examine the validity that the number of alternative servers is limited to 2, we have also measured these values in the dynamic server selection with the different number of alternative servers $|AS|$, (d-1) $|AS| = 0.25|S|$, (d-2) $|AS| = 0.5|S|$ and (d-3) $|AS| = 0.75|S|$. The results are shown in Section 5.2.

5.1 Comparison of Static and Dynamic Server Selection

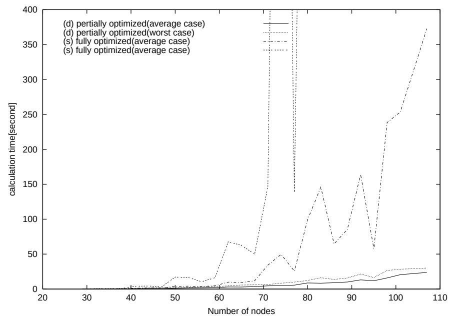

Fig. 5. Number of Nodes vs. Calculation Time: (a) Tiers Model

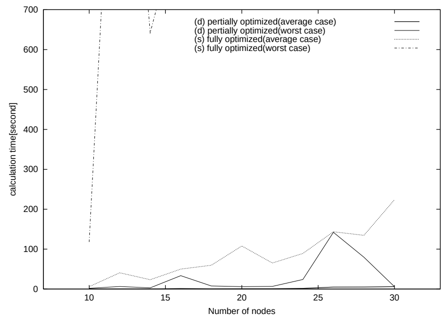

Fig. 6. Number of Nodes vs. Calculation Time: (b) Random Model

Calculation Time We show the calculation time of the static server selection and the dynamic server selection. We varied $|N|$ (the number of nodes) from 28 to 107 by every 3 nodes on Tiers model (Fig. 5) and from 10 to 30 by every 2 nodes on Random model (Fig. 6). In these graphs, the plots of the average time of (s) the static and (d) dynamic server selections are connected by dashed and solid lines, respectively. For each number of nodes, the range between the worst time and the best time is also shown by a vertical line.

We find the exponential increase of the calculation time in the static server selection around 90 nodes or higher on Tiers model and around 28 nodes or higher on Random model. On the other hand, we find the linear increase of the calculation time in the dynamic server selection. From these results, we can say that the proposed dynamic server selection can solve the problem within a reasonable time. Especially on Tiers model, the calculation time is just 10 seconds on 70 nodes in the worst case. On Random model, it took more calculation time in the static server selection. This is due to the distribution of distances between servers and receivers. On Tiers model, since the distances to servers largely differ from each other, the number of the candidate servers may be reduced in the calculation process. On the other hand, the distances are very close to each other on Random model.

The Total Sum of Preference Values We have measured the total sum of preference values of the dynamic and static server selections. The ratios of the former to the latter on Tiers model and Random model are shown in Fig. 7 and Fig. 8, respectively. On Tiers model, even in the worst case, the ratio is 89% and the average ratio is 95%. They are good enough to consider the tradeoff between the calculation time and optimality of satisfied preference. On Random model, there are a few worst cases that the ratios are in the range of 50%∼60%. This is because many receivers' alternative servers are converged to a few servers. However, the average ratio is kept more than 95%, therefore our dynamic server selection can keep high optimality compared to the static server selection on both models.

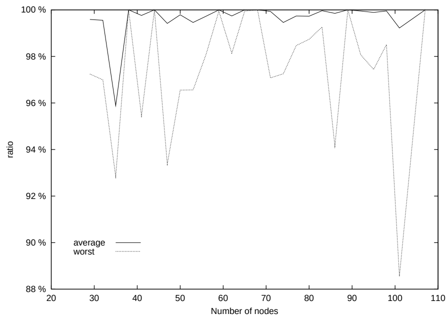

Fig. 7. Number of Nodes vs. Total Sum of Preference Values: (a) Tiers Model

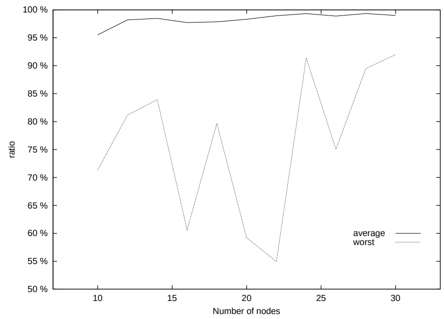

Fig. 8. Number of Nodes vs. Total Sum of Preference Values: (b) Random Model

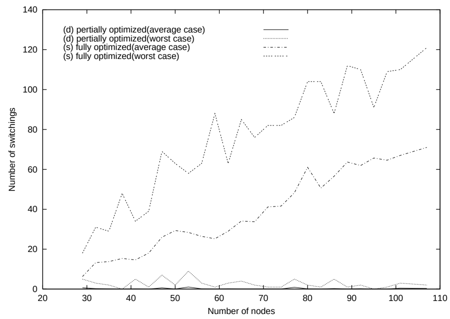

Fig. 9. Number of Nodes vs. Number of Server Switchings: (a) Tiers Model

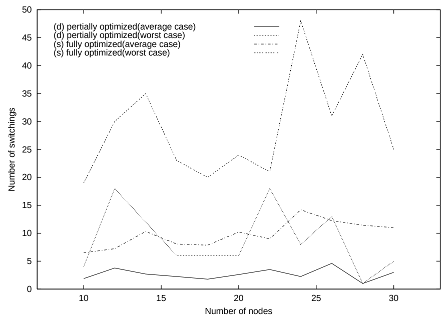

Fig. 10. Number of Nodes vs. Number of Server Switchings: (b) Random Model

The Number of Server Switchings We define a new variable $switch[s_i, r_j, c_k]$ for each set of variables s_i, r_j and c_k where $rcv'[s_i, r_j, c_k] = 1$ as follows.

$$switch[s_i, r_j, c_k] = 1 - rcv[s_i, r_j, c_k] \tag{8}$$

$switch[s_i, r_j, c_k]$ represents the fact that r_j stops receiving c_k from s_i. Thus we can represent the number of server switchings as follows.

$$\sum_i \sum_j \sum_k switch[s_i, r_j, c_k] \tag{9}$$

We have measured the number of server switchings. We show the results on Tiers model (Fig. 9) and on Random model (Fig. 10).

In the static server selection, the server switchings occurred 60 times on average on Tiers model with 100 nodes. Since we decided $|R| = 0.5|N|$, each receiver has at least one server switching in estimation. It is too much overhead in consideration of multicast join/leave latencies. On the other hand, the maximum number of server switchings is largely reduced in the dynamic server selection.

5.2 Effect of Number of Alternative Servers

In order to examine the effect of the number of alternative (selectable) servers to the calculation time, the total sum of satisfied preference values and the number of server switchings, we have measured these values on Tiers model in the dynamic server selection with the different number of alternative servers $|AS|$, (d-1) $|AS| = 0.25|S|$, (d-2) $|AS| = 0.5|S|$ and (d-3) $|AS| = 0.75|S|$.

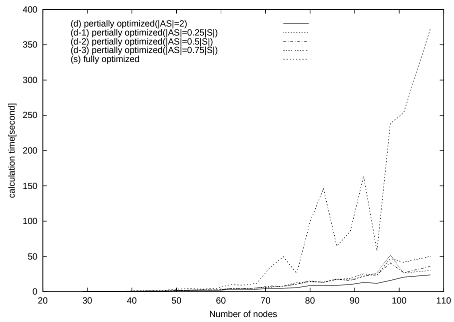

Fig. 11. Number of Nodes vs. Calculation Time on Tiers Model

Calculation Time We show the average of the calculation time in Fig. 11. Compared with (d) where $|AS| = 2$, we find the feature of divergence in (d-3), not so much as (s). Therefore we can say that our policy to limit the number of alternative server is adequate enough.

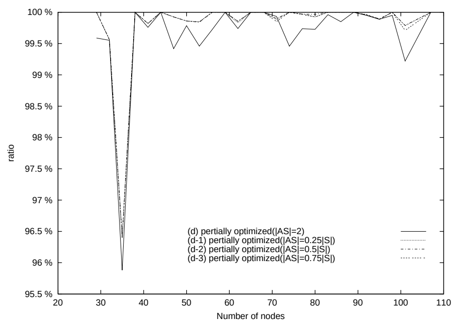

Fig. 12. Number of Nodes vs. Total Sum of Preference Values: Average Case

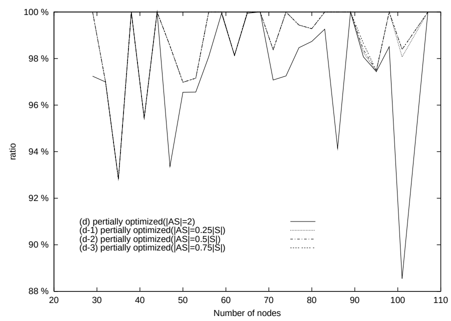

Fig. 13. Number of Nodes vs. Total Sum of Preference Values: Worst Case

The Total Sum of Preference Values We have measured the average and worst of the total sum of preference values and shown the ratios of (d), (d-1), (d-2) and (d-3) to (s) in Fig. 12 (average case) and Fig. 13 (worst case), respectively. The ratios of (d-1), (d-2) and (d-3) are greater than (d) and are kept more than 95% in the worst case. However, in the average case, (d) achieved almost the same values as (d-1), (d-2) and (d-3).

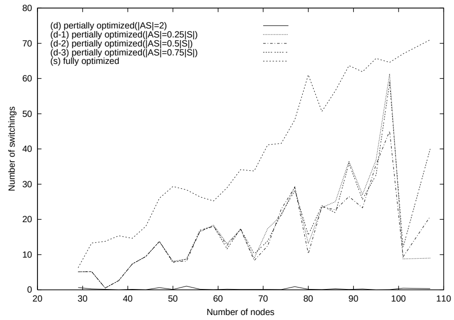

Fig. 14. Number of Nodes vs. Number of Server Switchings on Tiers Model

The Number of Server Switchings We show the number of server switchings in Fig. 14. The behavior of (d-1), (d-2) and (d-3) is similar to (s), while it is kept low in (d). From the results above, we can say that out policy to limit the number of alternative servers is adequate enough.

6 Conclusion

In this paper, we have proposed static and dynamic replicated server selection techniques for multiple multicast streams. In the proposed static server selection technique, if the location of servers and receivers and the shortest path between each pair of a server and a receiver on a network and each receiver's preference value for each content are given, the optimal combinations of the servers and the contents for each receiver are decided so that the total sum of the preference values of the receivers is maximized. We use the integer linear programming (ILP) technique to make a decision. Furthermore, in our dynamic server selection technique, the combinations of the servers and contents for each receiver are decided so that the number of server switchings is reduced and the total sum of the preference values is kept high, by restricting receivers who may suffer server switching and also alternative servers to be switched. Such restrictions also contribute fast calculation in ILP problems. Through simulations, we have confirmed that our dynamic server selection technique achieves less than 10 % in calculation time, more than 90 % in the sum of preference values and less than 5 % in the number of switchings, compared with the static server selection.

As our future work, we plan to design and implement an architecture to let receivers select optimal servers in existence of replicated multicast servers, based on the proposed method. We consider that our technique can be incorporated into application-layer anycast[9]. Application-layer anycast is an implementation of anycast at an application level, and is organized from ADN (Anycast Domain Name). An ADN server provides location service and replies to client's request with the list of servers that can provide the requested service. The ADN server can select those servers based on certain metrics. Therefore if the ADN server knows certain network information needed for our server selection technique, calculating the optimal allocation of servers will be possible on the ADN server. However, we have to consider the following two problems. The first one is that ADN servers reply to only the receivers who requested services. Therefore, we need an additional mechanism to let the other receivers switch their servers. The second one is how to collect the information. We are now investigating an efficient way to realize these requirements.

Moreover, in order to show the feasibility of the receiver/server limitation policy adopted in our dynamic server selection, we will try to analyze the upper bound of the solution (that is, the optimality of the solution of dynamic server selection) compared with the static server selection on Tiers model.

References

1. C. Diot, W. Dabbous and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 277-290, 1997.
2. X. Li, M.H. Ammar and S. Paul, "Video Multicast over the Internet," *IEEE Network Magazine*, Vol. 13, No. 2, pp46-60, 1999.
3. H. Sakate, H. Yamaguchi, K. Yasumoto, T. Higashino and K. Taniguchi, "Resource Management for Quality of Service Guarantees in Multi-party Multimedia Application," *Proc. of 1998 Int. Conf. on Network Protocols (ICNP'98)*, pp. 189–196, 1998.
4. H. Yamaguchi, K. Yasumoto, T. Higashino and K. Taniguchi, "Receiver-Cooperative Bandwidth Management for Layered Multicast," *Proc. of 1999 Int. Conf. on Network Protocols (ICNP'99)*, pp. 43–50, 1999.
5. Z. Fei, M.H. Ammar and E.W. Zegura, "Optimal Allocation of Clients to Replicated Multicast Servers," *Proc. of 1999 Int. Conf. on Network Protocols (ICNP'99)*, pp. 69–76, 1999.
6. R.L. Carter, M.E. Crovella, "Server Selection Using Dynamic Path Characterization in Wide-area Networks," *Proc. of INFOCOM'98*, 1998.
7. Z. Fei, S. Bhattacharjee, E.W. Zegura and M.H. Ammar, "A Novel Server Selection Technique for Improving the Response Time of a Replicated Service," *Proc. of INFOCOM'98*, 1998.
8. K.L. Calvert, M.B. Doar and E.W. Zegura, "Modeling Internet Topology," *IEEE Communications Magazine*, Vol. 35, No. 6, pp. 160–163, 1997.
9. S. Bhattacharjee, M.H. Ammar, E.W. Zegura, V. Shah and Z. Fei, "Application-Layer Anycasting," *Proc. of INFOCOM'97*, 1997.
10. G. Riley, M. Ammar and L. Clay, "Receiver-Based Multicast Scoping: A New Cost-Conscious Join/Leave Paradigm," *Proc. of 1998 Int. Conf. on Network Protocols (ICNP'98)*, pp. 254–261, 1998.
11. E. Amir, S. McCanne and R. Katz, "Receiver-driven Bandwidth Adaptation for Light-weight Session," *ACM Multimedia*, 1997.