Allocating Clients to Constrained Multicast Servers: An Optimal Solution*

Zongming Fei Mengkun Yang
Laboratory for Advanced Networking
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046
{fei,myang0}@cs.uky.edu

Mostafa Ammar Ellen Zegura
Networking and Telecommunications Group
College of Computing
Georgia Tech
Atlanta, GA 30332-0280
{ammar,ewz}@cc.gatech.edu

Abstract—**Multicast communication enables a server to send content to multiple clients at the same time through a multicast tree. To deal with the heterogeneity of client capacities, multiple multicast groups can be used to allocate clients with similar capacity to the same group, so that the performance perceived by clients can be improved. In this paper, we investigate the problem of allocating clients to constrained multicast servers, which, similar to clients, have different capacities. We explore some interesting issues raised by the constraints and propose an optimal solution to the allocation problem. We evaluate the solution and show substantial performance gain of our algorithm over those considering the server constraints separately.**

I. INTRODUCTION

Multicast communication is a service paradigm in which a server delivers information to multiple clients simultaneously [1], [2], [3]. Through the aggregation of request processing at the server and the aggregation of bandwidth requirements in the network, a multicast service can reduce the load on the server and network and, therefore, improve the performance perceived by the clients.

One well-known problem in multicast communication is the heterogeneity of multicast clients. A widely studied difference among clients is that of capacities, which are typically represented by the rate a client can get information from a multicast group. The difference can be the result of either the client itself or the path capacities from the server to clients. The solution is to use multiple multicast groups so that clients with similar capacities will join the same group and overall performance perceived by clients can be improved [4], [5], [6]. The basic assumption of these schemes is that a single server will form multiple groups and send information to different groups at different rates. Clients will be directed to an appropriate group. However, in this paradigm the total rate of all the groups will be limited by the capacity of the server.

In this paper, we want to consider an alternative paradigm (i.e., replication) in which different multicast groups have different servers. Though we somehow relax the single server requirement discussed above, some new problems arise. Prominent among them is the problem of differences among server capacities. This can be caused by either the server itself, or the link connecting the server. For example, one server can be located in an institution connected by a $T1$ line and another server can be located inside a major service provider backbone. Another example is that two servers may use machines with different capacities. These factors put an upper bound on how fast each server can send information out. It is called *server rate constraint* in this paper. This constraint sometimes changes the fundamental nature of the allocation problem and increase the complexity of solutions.

The focus of this paper is to study the problem of allocating clients to constrained multicast servers. The goal is to optimize the utility functions defined on receiving rates of clients. We will give a framework that can generate optimal solutions for a large category of utility functions. The performance study shows substantial performance gain of our algorithm over those considering the server constraints separately.

The rest of the paper is structured as follows. Section 2 formulates the problem. Section 3 illustrates a framework that can be used to generate optimal solutions. Performance results are presented in Section 4 and related work is discussed in Section 5. Section 6 concludes the paper.

II. PROBLEM DESCRIPTION

We consider that both servers and clients have rate constraints. Assume we have n clients and the rate (also called *isolated rate* [1]) of client i $(1 \leq i \leq n)$ is r_i. There are m servers and the maximal rate of server j $(1 \leq j \leq m)$ is s_j. Note it is the upper bound of the final sending rate of server j. In the following discussion, we simply call this upper bound *server rate* and its final sending rate *group rate*. We can assume that $r_i \leq r_{i+1}$ $(1 \leq i \leq n-1)$ and $s_j \leq s_{j+1}$ $(1 \leq j \leq m-1)$.

An allocation of clients to servers can be represented by a function $\phi : [1..n] \rightarrow [1..m]$ and $\phi(i) = j$ indicates that client i is allocated to server j. A utility function u is defined to represent the satisfaction level of a client. If a client has an isolated rate r and is allocated to a multicast group operating at rate x, the utility of this client will be $u(r, x)$. The value of most utility functions will increase monotonically as x increases. It it easy to prove that the optimal allocation has the property that $i_1 < i_2$ implies $\phi(i_1) \leq \phi(i_2)$, if the utility function satisfies the monotonicity property stated above. In

* The work of M. H. Ammar and E. W. Zegura is supported in part by grant ANI 9973115 from the National Science Foundation.

[1] As defined in [7], it can be the rate that would be achieved if this were the only client in the multicast group, or the minimum of the max-min fair shares in the links from a server to the client.

the rest of this paper, we will only consider those allocations ϕ with the property.

The goal is to maximize the sum of utilities of all the clients. We use x_j to represent the group rate of server j and obviously $0 \leq x_j \leq s_j$. The goal of our optimization becomes: maximize

$$\sum_{i=1}^{n} u(r_i, x_{\phi(i)}) \tag{1}$$

subject to $0 \leq x_j \leq s_j$ $(1 \leq j \leq m)$. The allocation problem is to determine the function ϕ, which defines how clients are allocated to servers, and the group rates x_j $(1 \leq j \leq m)$, which will be the receiving rate of those clients i $(1 \leq i \leq n)$ with $\phi(i) = j$.

Because of the property that $i_1 < i_2$ implies $\phi(i_1) \leq \phi(i_2)$, we observe that the allocation problem can be transformed into selecting m indices from n possible locations. If we define $k_1 = 1$ and $k_{m+1} = n + 1$, the problem is to select m indices k_j $(1 \leq j \leq m + 1$ *and* $1 \leq k_j \leq k_{j+1} \leq n + 1)$, such that all clients i with $k_j \leq i < k_{j+1}$ are grouped together and allocated to server j.

Server rate constraints capture the difference in capacities of servers, parallel to the different rates of clients. The problem with server rate constraints is a generalization of the problem without them, because we can instantiate the above problem to it by letting $s_j = \infty$ $(1 \leq j \leq m)$. Yet the constraints on server rates increase the complexity of the problem. During the optimization process, we have to take into consideration of server rates, in addition to client rates. One minor implication is that the problem of dividing clients into m groups becomes that of at most m groups. Consider the case in which we do not consider server rate constraints. It is beneficial for the purpose of improving client performance if we divide clients to as many groups as we can [2]. However, with the server rate constraints, it is possible that some server rate is so low that we do not want to allocate clients to it. Even if the rate of a server lies between client rates, there are cases in which we do not need to allocate clients to the server.

We now give two specific allocation problems, which will be used to evaluate the performance of the solution presented in the next section.

- *Max Rate Problem.* In this problem, we assume that clients cannot tolerate any loss and we have to set the group rate of a server to the minimum of server rate and the rates of all the clients allocated to it. We define $u(r, x) = x$ where $x \leq r$. The goal becomes maximizing the total rate of all the clients. The optimization of $\sum_{i=1}^{n} u(r_i, x_{\phi(i)}) = \sum_{i=1}^{n} x_{\phi(i)}$ can also be reduced to maximizing $\sum_{j=1}^{m} min(r_{k_j}, s_j)(k_{j+1} - k_j)$.
- *Max Satisfaction Problem.* In this problem, we assume that clients can tolerate loss and we may set the group rate of a server higher than the isolated rates of some clients to increase the sum of utilities of clients in the group. We use the utility

[2]The extreme case is unicast with one client in each group.

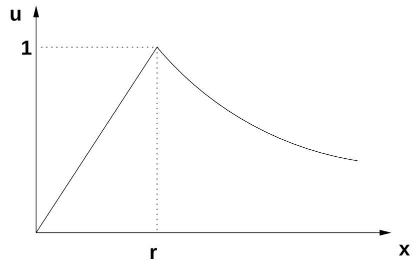

Fig. 1. An example utility function

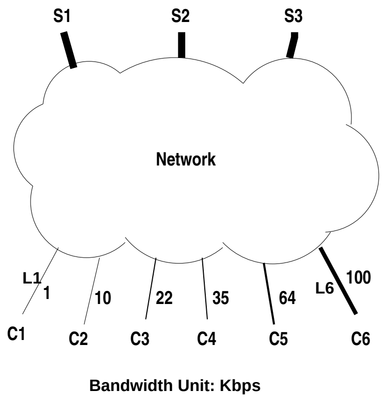

Fig. 2. An example of allocation problem

function $u(r, x) = \frac{min(r,x)}{max(r,x)}$, which is a linear function of x in the range $[0..r]$ and an inverse function in the range $[r..\infty]$, as shown in Fig. 1. This utility function indicates the satisfaction level of each individual client. We want to maximize the overall satisfaction level in this problem.

III. A Framework for Generating Optimal Allocations

The brute force solution to the problem involves trying every possible way to select m indices from n possible positions. We develop a framework that can solve the allocation problems in polynomial time. We first transform the original problem to the shortest path problem and then use Dijkstra's algorithm to solve it. We present instantiations of the framework to get algorithms for solving the max rate problem and max satisfaction problem.

Informally, we represent possible grouping of clients and association with a server using a directed graph, with the vertices arranged in rows and columns as follows. For the problem of allocating n clients to m servers, we have $n \times m + 1$ vertices. Fig. 2 illustrates the case in which we have 6 clients and 3 servers. There is one column of vertices for each server; there is one row of vertices for each client. Vertex $v_{i,j}$ in row i and column j represents the potential that the clients selecting server j will start from client i. An edge from $v_{i,j}$ to $v_{k,j+1}$ $(i \leq k)$ represents that client i to client $k - 1$ will be grouped

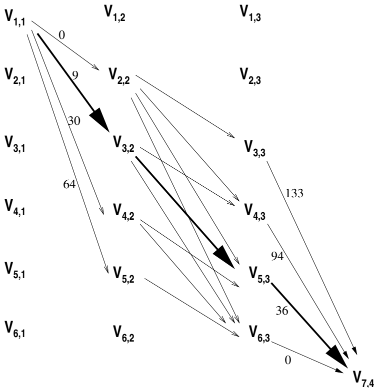

Fig. 3. An example solution

together and associated with server j.

The weight on this edge describes the performance associated with the choice. We use the weight on this edge to represent the "deviation" from ideal value for clients i to $k-1$ because of their association with server j. For example, in the max rate problem the ideal total rate for n clients is $\sum_{i=1}^{n} r_i$. Instead of maximizing the actual total rate of clients directly, our algorithm solves it by minimizing the deviation from the above ideal rate. For the edge from $v_{i,j}$ to $v_{k,j+1}$, all except client i will therefore receive at a rate lower than their isolated rate and deviation contributed by this edge will be $\sum_{\ell=i}^{k-1}(r_\ell - min(r_i, s_j))$.

A solution is represented as a path that includes one vertex per column in the graph. With the edges and weights as described, the shortest path corresponds to an optimal solution.

Formally, we define a deficit digraph $G_d = (V_d, E_d)$, where $V_d = \{v_{i,j}\} \cup \{v_{n+1,m+1}\}$ $(i = 1, \cdots, n, j = 1, \cdots, m)$ and E_d is given as follows:

1) For $1 \le i \le n$, connect $v_{1,1}$ to $v_{i,2}$;
2) For $1 \le i < k \le n,\ \ 2 \le j \le m-1$, connect $v_{i,j}$ to $v_{k,j+1}$;
3) For $1 \le i \le n$, connect $v_{i,m}$ to $v_{n+1,m+1}$;
4) For $2 \le j \le m-1$, connect $v_{1,j}$ to $v_{1,j+1}$.

The weight on the edge from $v_{i,j}$ to $v_{k,j+1}$ is denoted by $D_C(i,j,k)$ and represents the deviation from the ideal value if clients i to $k-1$ are allocated to server j. Define $D_C(i,j,k) = 0$ if $i \ge k$. Using Dijkstra's algorithm, we can compute the shortest path from $v_{1,1}$ to $v_{n+1,m+1}$. The path must be in the form $(v_{1,1}, v_{k_2,2}, \cdots, v_{k_j,j}, \cdots, v_{k_m,m}, v_{k_{m+1},m+1})$. These k_i's ($2 \le i \le m$) are where we want to partition the client set.

The remaining problem is how to calculate $D_C(i,j,k)$. The complexity depends on the definition of the utility function $u(r,x)$. We show solutions to the two problems defined in Section II.

- Max rate problem. The group rate x_j of server j is the minimum of the server rate and the lowest client rate, i.e., $x_j = min(r_i, s_j)$. The deficit caused by client ℓ $(i \le \ell \le k-1)$ is $r_\ell - x_j = r_\ell - min(r_i, s_j)$. Therefore, the total deviation is $D_C(i,j,k) = \sum_{\ell=i}^{k-1}(r_\ell - min(r_i, s_j))$. If we do not consider server rate constraints (or assume all s_j's are fairly large), the shortest path in the problem in Fig. 3 is $(v_{1,1}, v_{3,2}, v_{5,3}, v_{7,4})$ (shown as bold lines) and the optimal solution to the problem is to set servers s_1, s_2, s_3 at rates r_1, r_3, r_5, or 1, 22, 64 and allocate clients c_1, c_2 to s_1, clients c_3, c_4 to s_2 and clients c_5, c_6 to s_3. If we do consider server rate constrains and assume that $s_1 = 1, s_2 = 15, s_3 = 40$. The weights in Fig. 3 have to be changed and the shortest path will be $(v_{1,1}, v_{2,2}, v_{4,3}, v_{7,4})$. The optimal solution becomes allocating client c_1 to server s_1, clients c_2, c_3 to s_2, and clients c_4, c_5, c_6 to s_3.
- Max satisfaction problem. The calculation of $D_C(i,j,k)$ is more complicated, especially considering that the group rate can be larger than the lowest rate of clients. For the group of clients from i to $k-1$, the ideal satisfaction value is $k-i$, i.e., each client has satisfaction level equal to 1. Because they are allocated to the server j, the penalty is the deviation from the ideal value. If we can solve the optimization problem of maximizing the satisfaction level $\sum_i^{k-1} u(r_i, x_j)$ of this group, we can get the deviation $D_C(i,j,k) = k - i - max \sum_i^{k-1} u(r_i, x_j)$.

According to [7], for the utility functions satisfying the condition that $u(r,x)$ has one point of discontinuity at $x = r$ and the second derivative greater than or equal to 0 in other x values, $\sum_i^{k-1} u(r_i, x)$ will be maximized when x is one of r_ℓ $(i \le \ell \le k-1)$ or s_j. For this category of utility functions, the algorithm only need to test these r_ℓ's and s_j. Assuming that the goal is maximized at $x = a$, we get $D_C(i,j,k) = k - i - \sum_i^{k-1} u(r_i, a)$.

Our framework can be used to a wide range of problems, as long as we can find a way to calculate the minimal deviation $D_C(i,j,k)$. For example, if clients have different utility functions, we can use their respective utility function in the calculation of the distance $D_C(i,j,k)$ and get an optimal solution to this new problem. We can even use the framework to minimize the total load on all the servers by defining an appropriate $D_C(i,j,k)$ function [3].

The correctness of the framework is straightforward. Any allocation can be mapped to a path in the directed graph, with the length of the path being the minimal possible deviation from the ideal value by this allocation. The fact that the path found by our algorithm is the shortest implies that the solution has a minimal deviation from the ideal value. Therefore, it is an optimal solution.

One thing worth mentioning is that the framework naturally embeds the optimal solutions that have less than m groups. For example, the path $(v_{1,1}, v_{1,2}, v_{4,3}, v_{7,4})$ corresponds to two groups, with one group of 1, 2, 3 and an-

[3] However, we cannot use the framework for load balancing among servers, because generally the framework can only be used to solve problem of maximizing (or minimizing) some cumulative measure of all clients.

other group of 4, 5, 6. We avoid the edges from $v_{i,j}$ to $v_{i,j+1}$ for $2 \leq i \leq n, \quad 2 \leq j \leq m-1$, because the paths containing these edges have equivalent paths consisting of the edge from $v_{1,1}$ to $v_{1,2}$ and we do need to worry about empty groups which have a higher server rate. For example, in Fig. 3, the path $(v_{1,1}, v_{3,2}, v_{3,3}, v_{7,4})$ is equivalent to the path $(v_{1,1}, v_{1,2}, v_{3,3}, v_{7,4})$. If we allow the path $(v_{1,1}, v_{3,2}, v_{3,3}, v_{7,4})$, we need to worry about that server 2 is associated with no clients, and when we calculate $D_C(1, j, 3)$ (where $j = 1$) we need to use s_2 (i.e., s_{j+1}) rather than s_1. Our formulation eliminates these problems.

IV. PERFORMANCE RESULTS

We evaluate the performance of the framework for generating optimal solutions to the allocation problem. Specifically we study the performance of the derived algorithms for solving the two problems defined in Section II.

In our evaluation, we use 400 clients and their rates range from 0 to 1. The distribution of these rates is uniform, exponential, or normal. The mean value of the rates is always set to 0.5. For normal distribution, the variance is set to 1. We only use the values in the range in the cases of the exponential and the normal distributions. We experiment with 1 to 10 servers and their rates are randomly selected from (0,1), following the uniform distribution. For each data point in the results, we test 1000 times and get an average value.

In the max rate problem, the performance is the total rate. For a specific allocation, we determine the group rate of a server to be the minimum of server rate and client rates. We calculate the total rate of all the clients. This rate is normalized by using the ratio of this rate over the sum of isolated rates of clients.

In the max satisfaction problem, the performance measure is the total satisfaction level. For a specific allocation, the group rate of a server is either the server rate, or one of the client rates that are less than the server rate, whichever leads to a higher total satisfaction wins. We calculate the utility for each client and use the average value over all clients as the total satisfaction level.

In each of these two problems, we compare our optimization algorithm with two other algorithms.

- Random selection. Each client randomly selects a server.
- Natural selection. Each client selects a server with the highest rate from those having a lower (or equal) rate than this client.

If there are n clients and m servers, the complexity of random selection is $O(n)$ and the complexity of natural selection is $O(m+n)$. The complexity of the two maximum algorithms are $O(mn^3)$.

Fig. 4 and 5 show the results when the rates of clients are uniformly distributed in (0,1). In Fig. 4, we compare the performance of the three algorithms used to solve the *max rate problem*. The average rate of clients increases as the number

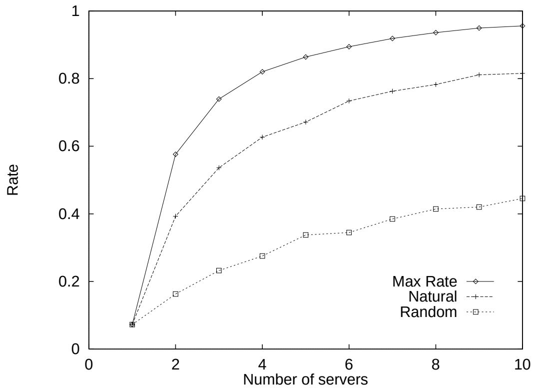

Fig. 4. The rate (uniform distribution of client rates

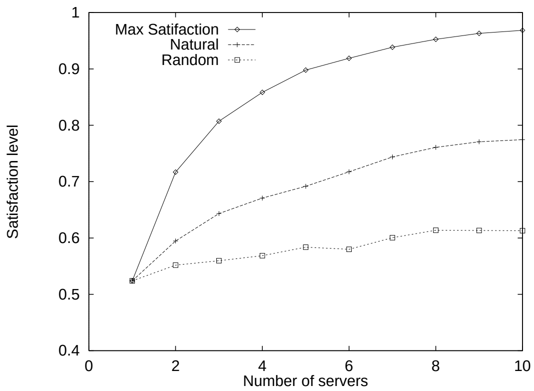

Fig. 5. The satisfaction level (uniform distribution of client rates)

of servers increases from 1 to 10. As expected, the random algorithm performs worst and our max rate algorithm performs the best. We notice our algorithm gives significant gains over the *natural selection* algorithm. It demonstrates a sharp increase when the number of servers increases from 1 to 6. Beyond that, the curve becomes flatter. We can conclude that the performance benefits will be obvious as we add replicas, but only up to a small number of them.

Fig. 5 shows the results when the algorithms are used to solve *max satisfaction problem*. The max satisfaction algorithm has the largest satisfaction level, compared to the natural selection and the random selection algorithms.

Fig. 6 to 9 show the results when the distribution of client rates is exponential or normal. The basic patterns are similar to the case of uniform distribution. Both the rate and satisfaction level are a little bit smaller, possibly because these two distributions are more skewed, while the distribution of server rates are more or less uniformly distributed over the range.

We compare the optimal solution to an algorithm (called *separate algorithm*) which considers server rate constraints separately. To be more specific, it is a modification of the max rate algorithm. First it divides clients into groups using the max rate algorithm, assuming that there are no server rate constraints Then it associates each group with a server.

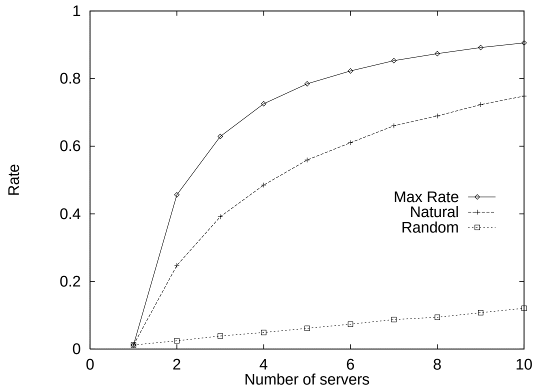

Fig. 6. The rate (exponential distribution of client rates)

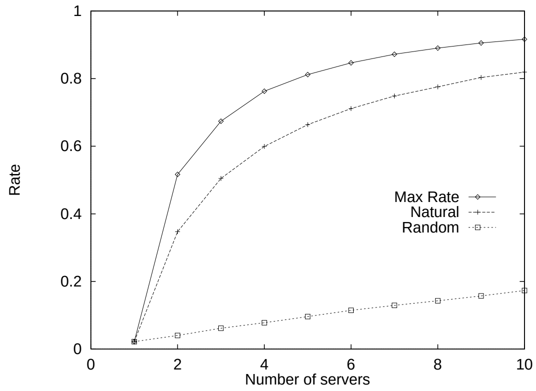

Fig. 8. The rate (normal distribution of client rates)

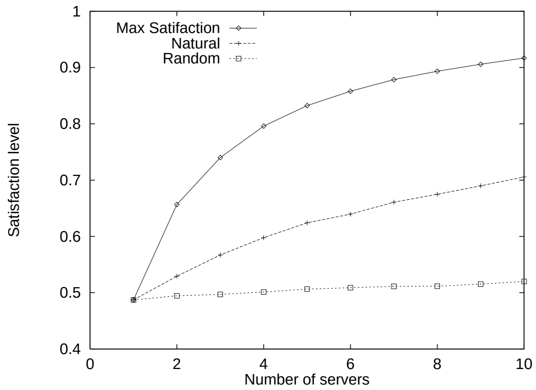

Fig. 7. The satisfaction level (exponential distribution of client rates)

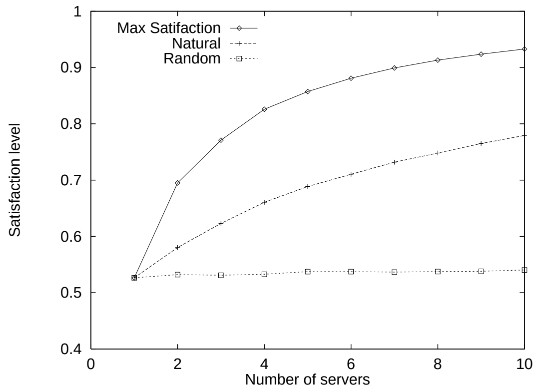

Fig. 9. The satisfaction level (normal distribution of client rates)

Fig. 10 shows the results of the max rate algorithm and the separate algorithm. They start with the same value when there is only one server (not shown in the plot, for the sake of clarity of the rest part of the plot). The difference between them becomes larger when the number of servers increases. Similar pattern exhibits in the Fig. 11. This tells us that when there are server rate constraints, we have to integrate them into the solution process.

V. Related Work

Server replication and selection has been extensively studied in the unicast environment [8], [9], [10], [11], [12], [13]. The emphases can be on architectural framework for providing selection services [14], [15], [13], or selection strategies and methods for improving performance [16], [10], or specific issues in a particular kind of (e.g., differentiated service) networks [12].

The use of multiple multicast groups or channels has been proposed to deal with the problem of heterogeneous client capacities. Ammar et al [4] proposed the idea of destination set splitting. The clients are divided into groups based on their capabilities and the sender would send to different groups at different rates controlled by the feedback from the clients in each group. In [5] Bhattacharyya et al designed a scheme to minimize the transfer time by using multiple multicast groups in bulk data transfer. Each client may join multiple channels and is characterized by a "rate" that represents how fast it can receive data. A simple rate control scheme was presented to determine the rate of each multicast channel. Multiple multicast channels were also used in a reliable multicast scheme [6] to reduce the end-host processing cost and network bandwidth usage. The basic assumption of these schemes is that the rate of a group is mainly limited by the capacities of clients. In contrast, our work generalizes the problem to the case in which servers also have different capacities and studies its impact on the decision of directing clients to an appropriate server.

VI. Concluding Remarks

Heterogeneity is one of most important problems in the multicast communication. In this paper, we investigate the issue of difference in server capacities and how they affect the allocation of clients. The contributions of the paper can be summarized in three aspects. First we define the optimization problem of allocating clients with different capacities to servers, which also have rate constraints. We analyze the implications of this added complexity. Second we give a general framework for solving the optimization problem effi-

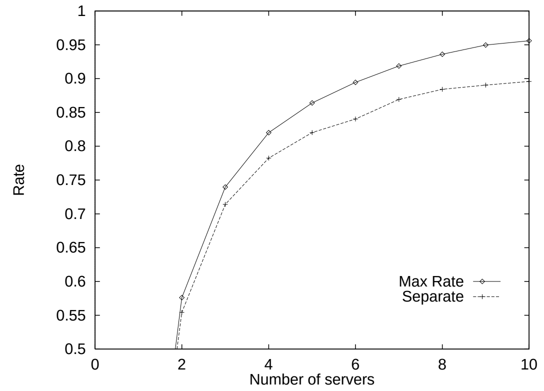

Fig. 10. Effect of server constraints – the rate

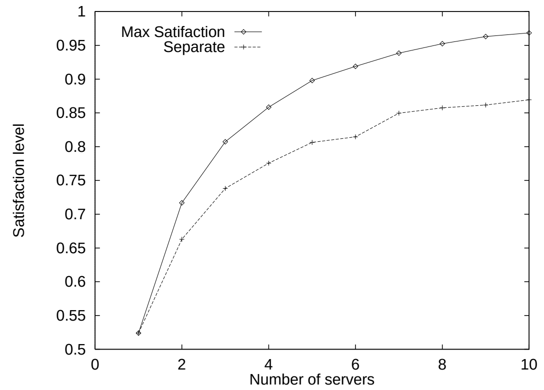

Fig. 11. Effect of server constraints – the satisfaction level

ciently. The conceptual framework has been demonstrated to be easily instantiated to solve two specific problems. Third the performance evaluation illustrates the potential benefits of our proposed algorithms over random and natural selection algorithms. More importantly, we demonstrate the advantages of our algorithms over those without considering server rate constraints in the solution process.

Our future work includes developing protocols to use the algorithms in multicast applications and services. The solution given in the paper assumes global knowledge about the clients and their utility functions and does not consider dynamic behavior of clients joining or leaving multicast session. We will investigate heuristic solutions to these problems. The solution in this paper gives us hint that we may need to establish some directory service that allows clients to know the current status of multicast groups.

ACKNOWLEDGMENT

The authors would like to thank anonymous referees for their helpful suggestions and comments on the paper.

REFERENCES

[1] Christophe Diot, Walid Dabbous, and Jon Crowcroft, "Multipoint communication: A survey of protocols, functions and mechanisms," *IEEE JSAC*, vol. 15, April 1997.
[2] Rodriguez and E. Biersack, "Continuous multicast push of web documents over the Internet," *IEEE Network Magazine*, vol. 12, no. 2, pp. 18–31, Mar-Apr 1998.
[3] Asit Dan, D. Sitaram, and P. Shahabuddin, "Scheduling policies for an on-demand video server with batching," in *Proceedings of ACM Multimedia 94*, 1994, pp. 15–23.
[4] Mostafa H. Ammar and Li-Ran Wu, "Improving the throughput of point-to-multipoint arq protocols through destination set splitting," in *Proceedings of INFOCOM'92*, May 1992, pp. 262–71.
[5] Supratik Bhattacharyya, James Kurose, Don Towsley, and Ramesh Nagarajan, "Efficient rate-controlled bulk data transfer using multiple multicast groups," in *Proceedings of IEEE Infocom'98*, March 1998, San Francisco, CA.
[6] Sneha Kasera, Gisli Hjalmtysson, Don Towsley, and James Kurose, "Scalable reliable multicast using multiple multicast channels," *IEEE/ACM Transactions on Networking*, vol. 8, June 2000.
[7] Tianji Jiang, Mostafa H. Ammar, and Ellen W. Zegura, "Inter-receiver fairness: A novel performance measure for multicast ABR sessions," in *Proceedings of ACM SIGMETRICS 98/Performance 98*, June 1998, Madison, Wisconsin.
[8] Andy Myers, Peter Dinda, and Hui Zhang, "Performance characteristics of mirror servers on the Internet," in *Proceedings of INFOCOM'99*, Mar. 1999, New York.
[9] Srinivasan Seshan, Mark Stemm, and Randy H. Katz, "SPAND: Shared passive network performance discovery," in *Proc 1st Usenix Symposium on Internet Technologies and Systems (USITS '97)*, Dec. 1997, Monterey, CA.
[10] Robert L. Carter and Mark E. Crovella, "Server selection using dynamic path characterization in wide-area networks," in *Proceedings of INFOCOM'97*, 1997, Kobe, Japan.
[11] Ellen Zegura, Mostafa Ammar, Zongming Fei, and Samrat Bhattacharjee, "Application-layer anycasting: A server selection architecture and use in a replicated web service," *IEEE/ACM Transactions on Networking*, vol. 8, August 2000.
[12] Fang Hao, Ellen W. Zegura, and Mostafa H. Ammar, "Supporting server selection in differentiated service networks," in *Proceedings of IEEE Infocom'2001*, April 2001, Anchorage, Alaska.
[13] Dina Katabi and John Wroclawski, "A framework for global IP-anycast (GIA)," in *Proceedings of ACM SIGCOMM 2000*, 2000.
[14] Craig Partridge, Trevor Mendez, and Walter Milliken, "Host anycasting service," *RFC 1546*, November 1993.
[15] Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei, "Application layer anycasting," in *Proceedings of INFOCOM'97*, 1997, Kobe, Japan.
[16] Zongming Fei, Samrat Bhattacharjee, Ellen Zegura, and Mostafa Ammar, "A novel server selection technique for improving the response time of a replicated service," in *Proceedings of INFOCOM'98*, 1998, San Francisco, CA.