Intra-Session Fairness in Multicast Communications [*]

Zongming Fei   Mengkun Yang
Department of Computer Science, University of Kentucky
Lexington, KY 40506-0046, U.S.A.
Email: {fei,myang0}@cs.uky.edu

Abstract

Multicast communication achieves scalability by sending data to multiple receivers at the same time. Receivers in a multicast session usually share the fate with each other, even though their processing speed and the capacity of the path they use can be quite different. A conventional multicast session usually consists of a single multicast group and the problem is how to set the group rate so that it is fair to both fast and slow receivers, to some extent. In a replicated multicast service, receivers are divided into groups based on their capacities and a multicast session can consist of multiple multicast groups. The question is how to divide receivers into groups exactly and set appropriate group rates so that it is fair to all the receivers. Most of current work focuses on optimizing the *social welfare* represented as a sum of some performance measures of receivers [Kar et al., 2002, Stoenescu et al., 2003]. In this paper, we define a new concept called *intra-session fairness* and give an optimal solution that can achieve fairness among receivers in the same session. The goal is to maximize the minimum fairness value of the receivers. The novelty of the framework is that it is independent of the specific definition of the fairness function on individual receivers. We illustrate a layering method to implement the max-min intra-session fair allocation and demonstrate the significant difference in fairness achieved by the maximal social welfare algorithm and the max-min intra-session fairness algorithm.

Keywords: fairness, multicast, replicated service

1 Introduction

Multicast communication achieves scalability by sending data to multiple receivers at the same time. One of the important issues to deal with is the heterogeneity among receivers. Their processing speed can be different. More importantly, the paths from the sender to them also have different rates. However, the receivers in the same

[*]Submitted on Aug. 24, 2004. Revised on April 29, 2005

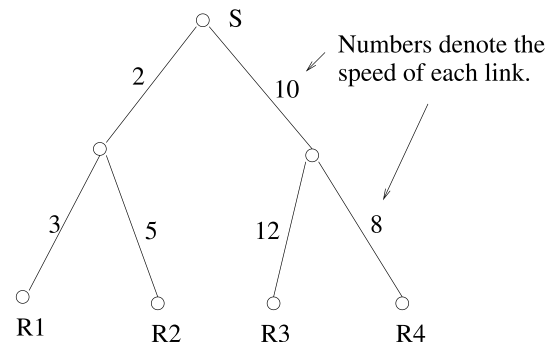

Figure 1: A multicast session with a single group

multicast session usually share the fate with each other. A conventional multicast session usually consists of a single multicast group, and the receivers in the group have to get data at the same rate. In some applications, the sender simply transmits at the lowest rate of all the receivers. It is not so fair, to some extent, to those receivers capable of receiving at a much higher speed. In other applications, receivers can tolerate a certain degree of packet loss and we can set the sending rate higher than the lowest rate. For example, for a multicast video-on-demand server, setting its sending rate higher than the lowest rate may improve the fairness for receivers with higher capacities. In Figure 1, a single group multicast session consists of one server S and four receivers $R1, R2, R3$ and $R4$. We can set the group rate to 2 (the lowest rate), or set the rate somewhat higher (e.g., 4). Here comes the first question. *In a session with a single multicast group, how should we set the group rate so that it is fair to some extent to both fast and slow receivers?*

More recently, using multiple multicast groups has been proposed to deal with the heterogeneity problem among receivers in the multicast communication [Bhattacharyya et al., 1998, Kar et al., 2001, Kar et al., 2002, Stoenescu et al., 2003, Kwon and Byers, 2004]. These schemes tend to put those receivers with similar capacities to the same group, and it is fairer to those receivers with a higher capacity. In Figure 1, we can let server S form two multicast groups and each group transmits at a different rate. For example, one group with rate 2 transmits data to receivers $R1$ and $R2$, and the other group with rate 8 transmits data to receivers $R3$ and $R4$. Note that all receivers $R1, R2, R3$ and $R4$ are still in the same multicast session even though they are allocated to different groups. This is because they are still receiving the same content, and it is perfectly fine to move a receiver ($R1, \cdots, R4$) from one group to another though there may be performance consequences.

A more general form of replicated multicast service is the replication of multicast servers. For example, in Figure 2, we have the multicast server S replicated at another place $S2$, which sends out the same content as S, but possibly at a different rate (thus quality). Each of them can form one or more multicast groups and each receiver will be allocated to one of the groups from either server.

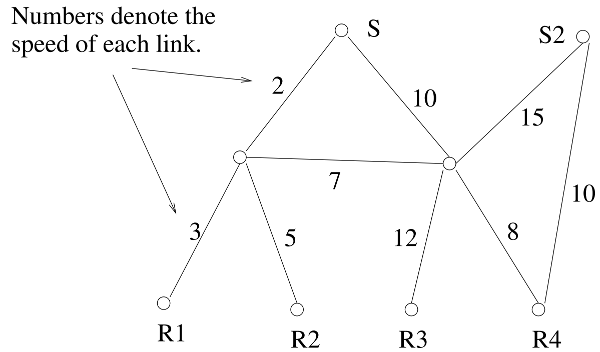

Figure 2: A multicast session with multiple groups

We need criteria to guide the allocation of receivers to one of the multicast groups in the session. The most widely used measure is the sum of utilities of receivers [Kar et al., 2001]. Receivers are divided into groups and the group rates are set in such a way that the sum is maximized. This may achieve better *social welfare* but, at the same time, has to sacrifice performance of some receivers and thus unfair. Here comes our second question. *In a session with multiple multicast groups, how should we divide receivers into groups and set the rates of those groups so that it is fair to some extent to all receivers?*

In this paper, we give a formal definition of *intra-session fairness* among receivers in the same multicast session. The concept is based on a fairness function defined for individual receivers and the goal is to maximize the minimum value of fairness of receivers in the same session.

Starting with a single group multicast session, we give optimal solutions to the rate assignment problem (the first question) for a variety of fairness function definitions. Note the solution is specific to a fairness function and there always exists an optimal solution to any fairness function. Then for a session with multiple groups, we give an algorithm that can generate an optimal solution to the problem of grouping and rate assignment (the second question). The novelty of the solution method is that it is a general framework and independent of the specific definition of the fairness function. It can be used to solve the grouping and rate assignment problem in a session with multiple multicast groups as long as we have a solution to the rate assignment problem for the fairness function in the single group session. We illustrate a method of implementing the max-min intra-session fair allocation by layering and demonstrate the significant difference in fairness achieved by the maximal social welfare algorithm and the max-min intra-session fair allocation.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives a formal definition of intra-session fairness. In Section 4, we give optimal solutions to the max-min intra-session fairness problem in sessions with a single group and with multiple groups. We also give performance evaluation of the algorithms, in comparison with the maximal social welfare algorithm. Section 5 discusses the implementation issues and illustrates a layering method for video streaming applications. We conclude the paper in Section 6.

2 Related Work

Fairness is one of the key issues in bandwidth allocation. Max-min fairness [Jaffe, 1981] aimed at maximizing the minimum bandwidth allocated to unicast flows involved. It was extended to the multicast case in [Tzeng and Siu, 1997], where all receivers in the same multicast group are allocated the same group rate, which is determined by all links involved from the sender to all receivers in the group. The goal is to maximize the minimum group rate. The utility max-min fairness maximizes the minimum of the utility of receivers instead of the rate [Cao and Zegura, 1999]. The scenarios discussed by them is different from ours in that each group constitutes a multicast session and it is not possible to move a receiver from one group to another to improve the fairness. All of them focused on allocating bandwidth among multicast groups, and therefore, fairness among receivers is not a concern.

Multi-rate max-min fair allocation extended the multicast max-min fairness further to the case where receivers in the same multicast session can receive at different rates [Rubenstein et al., 2002, Kwon and Byers, 2004, Stoenescu et al., 2003]. However, the allocation algorithm cannot be used to answer the two questions raised in Section 1 because its target is the *inter-session* fairness, i.e., the fairness between receivers from different sessions rather than the fairness among receivers in the same session. In a multirate multicast session, each receiver can receive at a different rate. Therefore, in a multicast session with n receivers, we need to be able to let them receive at n different rates. To implement multi-rate max-min fairness *end-to-end*, we have to use up to n groups if we have n receivers. If the number of groups we can use is smaller than n, we still have the problems of deciding the allocation of receivers into groups and choosing a rate for each group, in a way that is fair to receivers. This is the *intra-session fairness* problem we want to solve in this paper.

Another related work is the use of multiple multicast groups to deal with the problem of heterogeneous receiver capacities [Ammar and Wu, 1992, Kasera et al., 2000, Bhattacharyya et al., 1998, Kar et al., 2001]. In the scheme of destination set splitting [Ammar and Wu, 1992] the sender would send to different groups at different rates controlled by the feedback from the receivers in each group. Multiple multicast channels were also used in a reliable multicast scheme to reduce end-host processing costs and network bandwidth usage [Kasera et al., 2000]. Bhattacharyya et al. designed a scheme to minimize the average transfer time by using multiple multicast groups in bulk data transfer [Bhattacharyya et al., 1998]. Kar et al. addressed the problem of achieving rates that maximize the total receiver utility for multirate multicast sessions [Kar et al., 2001, Kar et al., 2002]. Most of them aimed at optimizing some performance measure that is the summation (or average) of receivers' performance, at the cost of sacrificing performance of some receivers.

Most closely related work is the inter-receiver fairness which intends to solve the same intra-session fairness problem [Jiang et al., 1998]. A fairness function is defined for individual receivers and the inter-receiver fairness is defined as the weighted sum of the fairness' of all receivers. Our observation [Fei and Yang, 2002] is that the sum is optimized, but the fairness of each individual receiver is not taken care of. It is still possible that a receiver has a very small fairness value in the solution. Therefore, it is not so fair to the receiver. In contrast, we will define the concept of max-min fairness that will maximize the minimum fairness value of the receivers, rather than optimize the summation of them.

3 Model

3.1 Performance Measure of an Individual Receiver

Performance of receivers can be measured in different ways. For bulk data transfer, we may be interested in the rate a receiver can get data from the sender. For multimedia applications, we may define more complicated utility functions, such as the proportion of useful data.

Recall that due to the difference in the processing speed and the rate of the path, the capacity of a receiver to receive data from the sender can vary from one to another. We define the measurement function with two parameters. One is the capacity (c) of the receiver and the other is the rate (r) of the multicast group it receives data from. Here are some examples.

1) $f_1(c,r) = min(c,r) = \begin{cases} r & \text{if } r \leq c \\ c & \text{if } r > c \end{cases}$

2) $f_2(c,r) = \frac{min(c,r)}{max(c,r)} = \begin{cases} r/c & \text{if } r \leq c \\ c/r & \text{if } r > c \end{cases}$

3) $f_3(c,r) = \begin{cases} r^2/c^2 & \text{if } r \leq c \\ c^2/r^2 & \text{if } r > c \end{cases}$

4) $f_4(c,r) = \begin{cases} 1-(1-r/c)^2 & \text{if } r \leq c \\ 1-(1-r/c)^2 & \text{if } c < r \leq 1.5c \\ 1.125c/r & \text{if } r > 1.5c \end{cases}$

The performance can be as simple as the actual receiving rate as in f_1, which targets at getting an equal share for each receiver. An alternative is the ratio of the group rate over the capacity such as f_2, which tries to achieve the goal that each receiver gets the rate proportional to its capacity. Performance measurement for individual receivers can be even more complicated (e.g., using functions f_3 and f_4). For most multimedia applications, even if a receiver has to drop some packets when the sending rate is higher than its capacity, there is still some value to the receiver. That is why we define $f(c, r) > 0$ when $r > c$ in these functions. However, the value to the receiver when it has to drop packets is not as good as in the case where the sending rate is equal to its capacity. As a matter of fact, as the sending rate (r) increases in the range $(c, +\infty)$, the fairness value decreases in f_2, f_3 and f_4. This implies that the higher the sending rate, the less value for the receiver in this range.

In some other applications, such as file or software distributions, data are no use at all for a receiver if the server is sending at a rate (r) greater than its capacity (c) and the receiver may lose part of a file. In this case, we may prefer the following fairness functions that have value 0 if $r > c$.

1') $f'_1(c, r) = \begin{cases} r & \text{if } r \leq c \\ 0 & \text{if } r > c \end{cases}$

2') $f'_2(c, r) = \begin{cases} r/c & \text{if } r \leq c \\ 0 & \text{if } r > c \end{cases}$

3') $f'_3(c, r) = \begin{cases} r^2/c^2 & \text{if } r \leq c \\ 0 & \text{if } r > c \end{cases}$

4') $f'_4(c, r) = \begin{cases} 1 - (1 - r/c)^2 & \text{if } r \leq c \\ 0 & \text{if } r > c \end{cases}$

In some multimedia applications, if the sending rate is lower than a certain lower bound, receivers will not be able to get the minimal quality media stream and all the data will be useless. Therefore, we can define a group of fairness functions with an additional argument, i.e., the lower bound ℓ. Obviously, $\ell \leq c$. We can define $h_j(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ f_j(c, r) & \text{if } r \geq \ell \end{cases}$. If we expand $f_j(c, r)$, we get the following fairness functions.

1) $h_1(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ r & \text{if } \ell \leq r \leq c \\ c & \text{if } r > c \end{cases}$

2) $h_2(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ r/c & \text{if } \ell \leq r \leq c \\ c/r & \text{if } r > c \end{cases}$

3) $h_3(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ r^2/c^2 & \text{if } \ell \leq r \leq c \\ c^2/r^2 & \text{if } r > c \end{cases}$

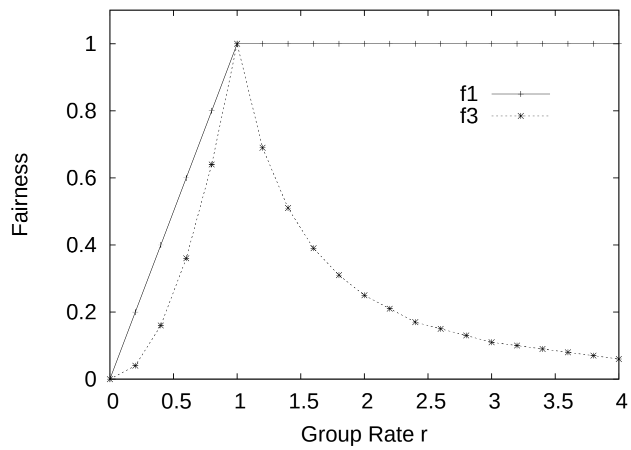

Figure 3: Fairness functions f1 and f3

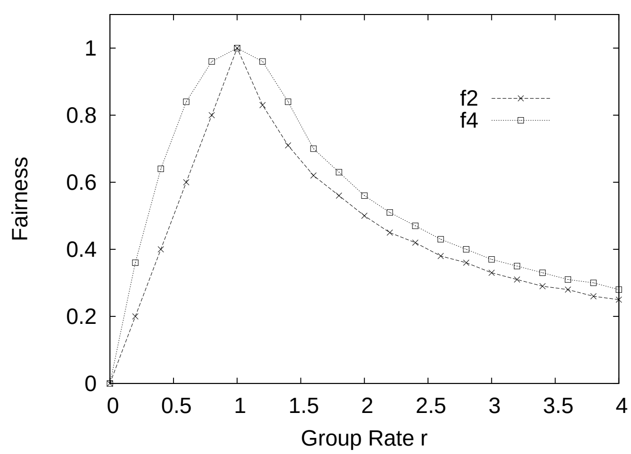

Figure 4: Fairness functions f2 and f4

4) $h_4(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ 1 - (1 - r/c)^2 & \text{if } \ell \le r \le c \\ 1 - (1 - r/c)^2 & \text{if } c < r \le 1.5c \\ 1.125c/r & \text{if } r > 1.5c \end{cases}$

Similarly, we can define $h'_j(c, \ell, r) = \begin{cases} 0 & \text{if } r < \ell \\ f'_j(c, r) & \text{if } r \ge \ell \end{cases}$.

We plot functions f_1 and f_3 in Figure 3, and functions f_2 and f_4 in Figure 4, where $c = 1$. All these functions (including f'_j's, h_j's, and h'_j's, where $1 \le j \le 4$) have the following property: it monotonically increases over $(0, c)$ and decreases over $(c, +\infty)$. The intuition behind this is that the closer the group rate to the capacity, the better its performance. A receiver achieves the maximal utility or fairness if the group rate is equal to its capacity.

Based on the measures defined for individual receivers, we will define *social welfare* and *intra-session fairness*. We want to emphasize that our definitions and algorithms do not depend on any specific form of functions defined for individual receivers, and one can use whatever performance measure he/she thinks appropriate for a specific application.

3.2 Social Welfare

The social welfare emphasizes the total or average performance of all receivers. We define the social welfare of a session with n receivers as the weighted sum of performance measures over all receivers, i.e.,

$$W = \sum_{i=1}^{n} w_i * v_i$$

where w_i is the weight assigned to receiver i and v_i is the performance value for receiver i using one of the performance functions (f_j, f'_j, h_j, h'_j) appropriate for the application. For example, if we use f_2 as the performance function for individual receivers, we have $v_i = f_2(c_i, r_{g(i)})$, where c_i is the capacity of receiver i, $g(i)$ is the group to which receiver i belongs and $r_{g(i)}$ is its group rate. The social welfare will be $W = \sum_{i=1}^{n} w_i * f_2(c_i, r_{g(i)})$.

The weights (w_i's) and performance function will be determined by a specific application. The simplest case for weights is all $w_i = 1$. Note even if f is some fairness measure for each receiver (for example f_2), we still consider W is a social welfare measure, not an intra-session fairness measure.

3.3 Intra-Session Fairness

Intra-session fairness achieves the fairness among receivers in the same multicast session. Unlike the social welfare measure, the main focus is on individual values of some performance function f for receivers. Therefore, we define the intra-session fairness to be a vector of performance measures of receivers [1]. Given n receivers, the intra-session fairness of an allocation is defined as a vector, i.e.,

$$F = \langle v_1, v_2, \cdots, v_n \rangle$$

where v_i is the performance value for receiver i using one of the performance functions (f_j, f'_j, h_j, h'_j) appropriate for the application.

[1] Actually it can also be defined as a multi-set of values of the fairness function for each receiver and then we can define an ordering over multi-sets. However, we choose to use the vector notation for the ease of presentation.

To compare the relative fairness between allocations, we define an ordering over the fairness vectors. It is hard to compare two vectors if their elements (v_i's) are not ordered. Our idea is that for each vector, we define a canonical form in which elements in the vector are ordered. We then define an ordering over the canonical forms and finally extend that ordering to any fairness vectors.

Def 1: The canonical form of vector F is denoted as $\mathcal{C}(F)$. If $F = \langle v_1, v_2, \cdots, v_n \rangle$, then $\mathcal{C}(F) = \langle u_1, u_2, \cdots, u_n \rangle$, where $u_1, u_2, \cdots, u_n$ is a permutation of $v_1, v_2, \cdots, v_n$, and $u_i \leq u_{i+1}$ for all $1 \leq i \leq n-1$.

Def 2: We define an ordering $\prec_v$ over canonical forms. $\langle u_1, u_2, \cdots, u_n \rangle \prec_v \langle t_1, t_2, \cdots, t_n \rangle$ iff there exists an i $(1 \leq i \leq n)$ such that $u_i < t_i$ and for all $1 \leq j < i$, $u_j = t_j$.

Def 3: Two vectors $F_1 \prec_v F_2$ iff $\mathcal{C}(F_1) \prec_v \mathcal{C}(F_2)$.

To make it complete we define an equivalent relation over fairness vectors.

Def 4: Two vectors $F_1 =_v F_2$ if they have the same canonical form, i.e., $\mathcal{C}(F_1) = \mathcal{C}(F_2)$.

Def 5: Two vectors $F_1 \preceq_v F_2$ if $F_1 \prec_v F_2$ or $F_1 =_v F_2$.

The goal of the *max-min intra-session fair* allocation is to maximize the minimum of fairness values in the fairness vector. With the ordering defined, the optimal solution is the one that maximizes F over the ordering $\prec_v$, or to try to get $\max_{\prec_v} \langle v_1, v_2, \cdots, v_n \rangle$.

3.4 Summary

The social welfare and the intra-session fairness are two different measures in multicast services. Improving one of them does not necessarily make the other worse. We do observe that in some cases we can improve both measures at the same time. We do not argue that the intra-session fairness is a better measure than the social welfare in allocating receivers to one of the replicated multicast groups and determining the rates of these multicast groups. Nor do we object to the ideas of using or adapting the algorithm maximizing one measure to get a (possibly approximate) solution to the other problem. However, we want to point out that in general we cannot achieve the max-min intra-session fairness by maximizing the social welfare and *vice versa*.

4 A Framework for Max-min Intra-session Fair Allocations

4.1 Single Multicast Group

In a single group multicast session, all receivers get the same group rate no matter what their capacities are. The problem is to determine the rate of the group so that the fairness F is maximized with regard to $\prec_v$. This depends on the definition of individual fairness function (f_j, f'_j, h_j, h'_j). Given n receivers, we assume the capacity of receiver i is c_i and the lower bound for receiver i is ℓ_i ($\ell_i \leq c_i$). Without loss of generality, we assume $c_i \leq c_{i+1}$ for all $1 \leq i \leq n-1$. Note we do not have assumptions about the relationships between ℓ_i's, and it is quite possible that $\ell_i > \ell_{i+1}$ for some i. Table 1 gives the optimal group rate to maximize the intra-session fairness F for the functions defined in Section 3. Appendix A gives the details on how to derive these results.

Fairness function	Optimal group rate
$f_1(c,r) = \begin{cases} r & \text{if } r \leq c \\ c & \text{if } r > c \end{cases}$	$r = c_n$
$f_2(c,r) = \begin{cases} \frac{r}{c} & \text{if } r \leq c \\ \frac{c}{r} & \text{if } r > c \end{cases}$	$r = \sqrt{c_1 * c_n}$
$f_3(c,r) = \begin{cases} \frac{r^2}{c^2} & \text{if } r \leq c \\ \frac{c^2}{r^2} & \text{if } r > c \end{cases}$	$r = \sqrt{c_1 * c_n}$
$f_4(c,r) = \begin{cases} 1-(1-\frac{r}{c})^2 & \text{if } r \leq c \\ 1-(1-\frac{r}{c})^2 & \text{if } c < r \leq 1.5c \\ 1.125\frac{c}{r} & \text{if } r > 1.5c \end{cases}$	$r = \begin{cases} \frac{2*c_1*c_n}{c_1+c_n} & \text{if } c_n \leq 3c_1 \\ x, \text{where } x^3 - 2c_n x^2 + 1.125 c_1 c_n^2 = 0 & \text{if } c_n > 3c_1 \\ \quad \text{and } c_1 \leq x \leq c_n. & \end{cases}$
$f'_j(c,r)$	$r = c_1$
$h_j(c,\ell,r)$	$r = max(max(\ell_i), r_{f_j})$, where r_{f_j} is the solution to f_j.
$h'_j(c,\ell,r)$	$r = \begin{cases} c_1 & \text{if } max(\ell_i) \leq c_1 \\ \text{depends on } f'_j \text{ and relative} & \text{if } max(\ell_i) > c_1 \\ \quad \text{values of } c_i\text{'s and } \ell_i\text{'s} & \end{cases}$

Table 1: The optimal group rate in the case of single multicast group

In most cases, we have a simple solution. However, the computation can be complicated depending on the definition of the fairness function. For example, the solution for function f_4 needs to solve an equation $P(x) = 0$ if $c_n > 3c_1$, where $P(x) = x^3 - 2c_n x^2 + 1.125 c_1 c_n^2$. For f'_j ($j = 1, \cdots, 4$), receivers cannot tolerate loss ($f'_j(c,r) = 0$ if $r > c$), we have to set the group rate to the slowest receiver to avoid having an element of 0 in the vector. For both f_j and f'_j, the group rate depends only on rates of the slowest receiver and the fastest receiver. This is because we impose the requirement that the fairness function increases monotonically over $(0, c)$ and decreases monotonically over $(c, +\infty)$. For h_j and h'_j, lower bound values ℓ_is play a role in the solution. For h'_j, we have a simple solution $r = c_1$ if $max(\ell_i) \leq c_1$. In the cases when $max(\ell_i) > c_1$, the solution depends on the relative values of c_i's and ℓ_i's. We do not have an algebraic solution. Instead, we can design a simple algorithm to find the optimal group rate.

4.2 Multiple Multicast Groups

To deal with the heterogeneity problem of receivers, multiple multicast groups are used to send the content to receivers at different rates. Receivers with similar capacity join the same group so that slower receivers will not affect the performance of faster receivers. If the maximal number of groups we can have is m, the problem becomes grouping receivers with similar rates together and assigning them to one of the m groups with the goal of maximizing the intra-session fairness F.

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 problem in polynomial time. The idea is to transform the original problem to the longest path problem and then use a modified Dijkstra's algorithm [Dijkstra, 1959] to solve it. In the following discussion, we assume $c_i \leq c_{i+1}$ for all $1 \leq i \leq n-1$.

Informally, we represent possible groupings of receivers and their association with a multicast group as a directed graph, with the vertices arranged in rows and columns as follows. For the problem of allocating n receivers to m groups, we have $n \times m + 1$ vertices. Figure 5 illustrates the case in which we have 6 receivers and 3 groups. There is one column of vertices for each group; there is one row of vertices for each receiver. Vertex $v_{i,j}$ in row i and column j represents that the receivers assigned to group j will start from receiver i. An edge from $v_{i,j}$ to $v_{k,j+1}$ ($i \leq k$) represents that receiver i to receiver $k-1$ will be grouped together and assigned as group j. The weight on this edge is the intra-session fairness vector of this multicast group. Obviously we only have edges between adjacent columns, and those edges always start from a node on the left side and point to a node on the right side on the same row or below. An edge pointing to a node on the same row means that no nodes will be allocated in this group. For clarity of presentation, we do not show these edges in Figure 5. Also removed from the figure are those edges that have to be combined with such edges to form a path from $v_{1,1}$ to $v_{n+1,m+1}$. The length of a path is the concatenation of elements from the fairness vectors on the edges in the path. The comparison between the lengths of two paths is based on the ordering $\prec_v$ defined in Section 3.3. Note we only need to compare two paths with the same end points and therefore, they have the same number of elements in the vectors.

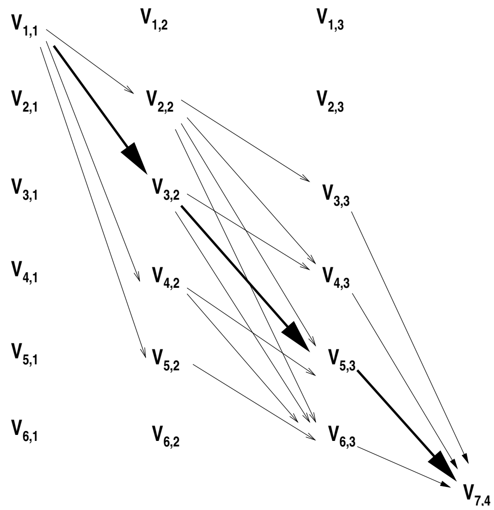

Figure 5: An example with six receivers and three groups

A solution is represented as a path that includes one vertex per column in the graph. With the edges and weights as described, the longest path corresponds to an optimal solution. The details for generating the optimal solution are given in Algorithm 1. It makes use of the operation $concate$ to combine two vectors together. We use $L(u, v)$ to record the fairness vector, and $G(u, v)$ to record how the receivers are splitted into groups. It starts with the initialization of the edges in the graph by statements 1 to 11. We use $F_{v_{i,j},v_{k,j+1}}$ to represent the fairness vector calculated based on the definition in Section 3.3 for the group with receivers $i, i+1, \cdots, k-1$, if $i < k$. We also define $F_{v_{i,j},v_{i,j+1}} = \langle\rangle$, i.e., a vector with no element. Statements from 12 to 23 in the algorithm find the longest path from $v_{1,1}$ to $v_{n+1,m+1}$. The result is in $G(v_{1,1}, v_{n+1,m+1})$, which records how the receivers should be splitted. For example, if $n = 10, m = 3$ and $G(v_{1,1}, v_{11,4}) = \langle 3, 8, 11\rangle$, we will know that we should divide the receivers into three groups with 1,2 as a group, 3, 4, 5, 6, 7 as a group, and 8, 9, 10 as a group.

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 intra-session fairness of the allocation. The fact that the path found by our algorithm is the longest implies that the solution achieves the maximal intra-session fairness. Therefore, it is an optimal solution.

Algorithm 1 Generating optimal solutions for the multiple group case

for $i = 1$ to n **do**
  $L(v_{1,1}, v_{i,2}) = F_{v_{1,1},v_{i,2}}$
  $G(v_{1,1}, v_{i,2}) = \langle i \rangle$
for $j = 2$ to $m - 1$ **do**
  for $i = 1$ to n **do**
    for $k = i$ to n **do**
      $L(v_{i,j}, v_{k,j+1}) = F_{v_{i,j},v_{k,j+1}}$
      $G(v_{i,j}, v_{k,j+1}) = \langle k \rangle$
for $i = 1$ to n **do**
  $L(v_{i,m}, v_{n+1,m+1}) = F_{v_{i,m},v_{n+1,m+1}}$
  $G(v_{i,m}, v_{n+1,m+1}) = \langle m + 1 \rangle$
for $j = 3$ to m **do**
  for $k = 1$ to n **do**
    Initialize $L(v_{1,1}, v_{k,j})$ to a vector of length $j - 1$ with all elements equal to zero
    for $i = 1$ to k **do**
      if $L(v_{1,1}, v_{k,j}) \prec_v \mathit{concate}(L(v_{1,1}, v_{i,j-1}), L(v_{i,j-1}, v_{k,j}))$ **then**
        $L(v_{1,1}, v_{k,j}) = \mathit{concate}(L(v_{1,1}, v_{i,j-1}), L(v_{i,j-1}, v_{k,j}))$
        $G(v_{1,1}, v_{k,j}) = \mathit{concate}(G(v_{1,1}, v_{i,j-1}), G(v_{i,j-1}, v_{k,j}))$
Initialize $L(v_{1,1}, v_{n+1,m+1})$ to a vector of length n with all elements equal to zero
for $i = 1$ to n **do**
  if $L(v_{1,1}, v_{n+1,m+1}) \prec_v \mathit{concate}(L(v_{1,1}, v_{i,m}), L(v_{i,m}, v_{n+1,m+1}))$ **then**
    $L(v_{1,1}, v_{n+1,m+1}) = \mathit{concate}(L(v_{1,1}, v_{i,m}), L(v_{i,m}, v_{n+1,m+1}))$
    $G(v_{1,1}, v_{n+1,m+1}) = \mathit{concate}(G(v_{1,1}, v_{i,m}), G(v_{i,m}, v_{n+1,m+1}))$

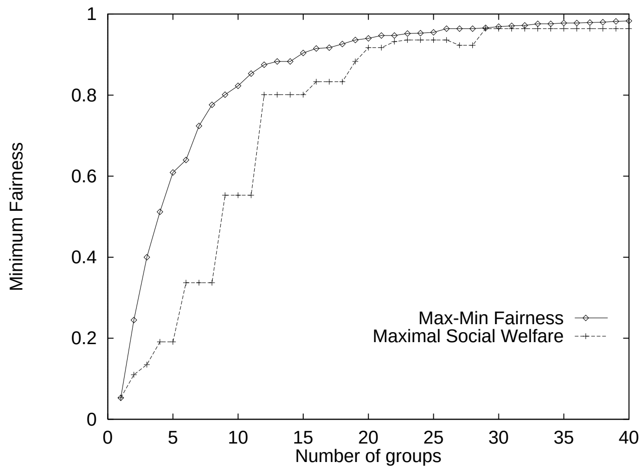

Figure 6: Minimum fairness (uniform)

4.3 Performance Results

We evaluate the performance of the max-min intra-session fair algorithm and the maximal social welfare algorithm. We have 100 receivers with capacities ranging from 0 to 1. Three methods are used to generate them. First, they are uniformly distributed from 0 to 1. Second, we generate values that follow the exponential distribution with mean equal to 0.5. We only use thoese values between 0 and 1. The actual mean of the capacities of the 100 receivers is 0.30. Third, we use the normal distribution with a mean of 0.5 and variance equal to 0.25. We also only use the values between 0 and 1. The function f_2 is used as the individual fairness function in the calculation of the intra-session fairness and the social welfare (equal weights). Figure 6 shows the minimum fairness for the max-min fair allocation algorithm and the maximal social welfare algorithm when the capacity of receivers follows the uniform distribution. As the number of groups in a session increases from 1 to 40, the minimum fairness of receivers increases from 0.05 to more than 0.96. However there are significant differences between the max-min intra-session fairness algorithm and the maximal social welfare algorithm. For example, when the number of groups is 8, the max-min fairness algorithm has a minimum fairness of 0.78 while the maximal welfare algorithm has a minimum fairness of 0.34.

Figure 7 shows the minimum fairness when the capacity follows the exponential distribution. The minimum fairness is a bit closer between two algorithms, but we can still see the maximal welfare algorithm has a much lower minimum fairness than the max-min fairness algorithm, especially when the number of groups is between 4 and 14. We have similar results in the case of normal distribution.

To take a closer look at the fairness value for other receivers, we plot all elements of the fairness factor in Figures 8 and 9 when the number of groups is 12. We sort the fairness values of receivers in increasing order.

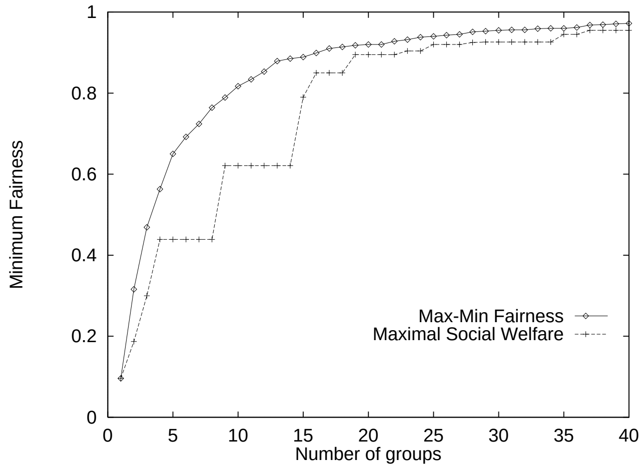

Figure 7: Minimum fairness (exponential)

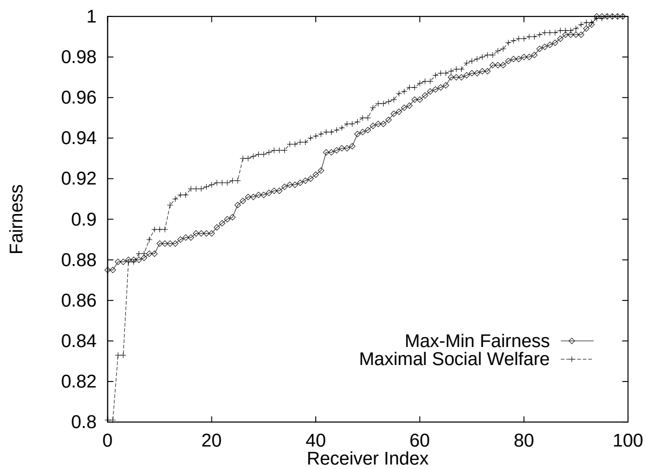

Figure 8: Fairness of receivers (uniform)

The x-axis is the receiver index and the y-axis is the fairness of each receiver. Figure 8 is the result when the distribution of receiver capacity is uniform. The max-min fairness algorithm has larger fairness values than the maximal welfare algorithm when the receiver index is smaller than 5, and therefore it is fairer for those at the lower end of fairness values based on the fairness function used. It is at the expense of smaller fairness value of those receivers having a higher index. Figure 9 shows the result when the distribution is exponential. The max-min fairness has higher fairness values than the maximal welfare when the receiver index is smaller than 8. Though the max-min fairness has a smaller total average of fairness values than the optimal welfare, it is fairer based on the max-min criterion.

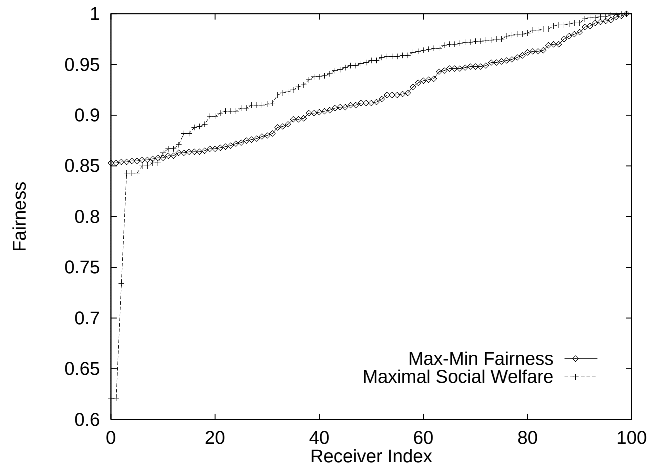

Figure 9: Fairness of receivers (exponential)

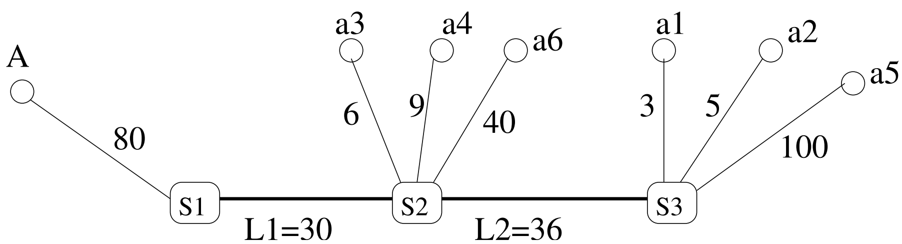

Figure 10: Sharing with receivers in the same multicast session

5 Implementation Issues

One of key issues in the max-min fair allocation is to determine the capacity for each receiver. When the links in the path from the source to a receiver are *not* shared with other unicast/multicast sessions, such as receivers in Figure 10, the capacity of a receiver is the bandwidth of the bottleneck link in the path, because this is the maximal rate that the receiver can get data from the source. When the links from the source to a receiver are shared with other sessions, we can use the multi-rate max-min fair allocation to determine the bandwidth at each link that can be used by this multicast session [Rubenstein et al., 2002]. For example, in Figure 11 we want to know the capacities of receivers in a multicast session with A as the sender and $a1$, $a2$, ..., $a6$ as receivers. The links ($L1$,$L2$) to receivers are shared with other two sessions, one having source B and receiver $b1$ and the other having source C and receiver $c1$. The bandwidth allocated to multicast session A by the multi-rate max-min fair allocation is shown in Figure 12. The capacities for $a1$ to $a6$ are 3, 5, 6, 9, 12 and 18, respectively. Note that L_1 and L_2 play a role in determining the bandwidth that can be allocated to the session based on the multi-rate max-min fair allocation, and the capacities of the receivers.

The multi-rate max-min fair allocation can be implemented end-to-end by using multiple multicast groups

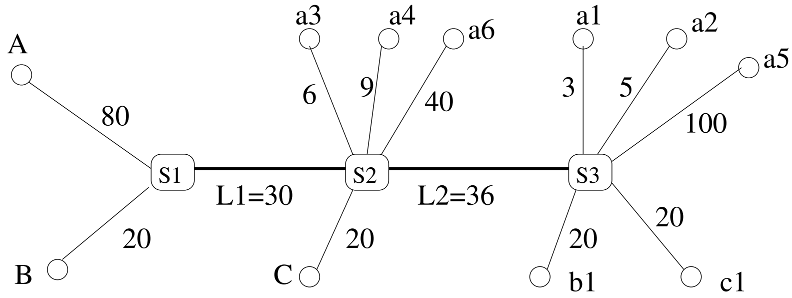

Figure 11: Links shared by receivers from other sessions

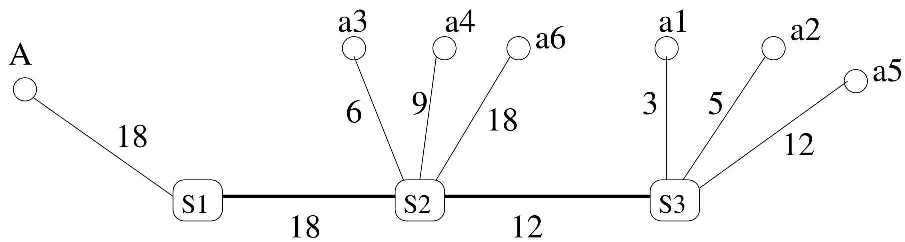

Figure 12: Bandwidth allocated for the session

for each session. If we have enough groups to use, we can achieve their capacity for all receivers. We need to use six multicast groups for session A. However, in practice we have to implement with a limited number of groups and usually this will inevitably result in an underutilization of bandwidth at various points in the network. This is a tradeoff of bandwidth utilization for simplicity of management and implementation. The problem is, with a given number of groups we can use, how should we divide receivers into groups and determine a rate for each group? We can use the algorithms in the previous section to find the max-min intra-session fair groupings and group rates. For example, if we can only use 3 groups, the optimal solution is to group $a1$ and $a2$ together, $a3$ and $a4$ together, and $a5$ and $a6$ together. Their rates will be set to 3, 6 and 12, respectively.

Setting three multicast groups with rates 3, 6, 12 and having receivers join corresponding groups will result in the total rates over shared links exceeding the bandwidth allocated for the session. For example, the bandwidth allocated to this session over link $L1$ is 18. However, the total rate of three groups over link $L1$ is 3+6+12=21, higher than the bandwidth allocated to the session, which is 18. For multimedia applications where not all receivers need to get all the data, we propose a layering based solution. If we have m groups of receivers and the max-min intra-session fair allocation results in the group rates from r_1 to r_m, we can implement it as m layers and set the rate of layer j to $r_j - r_{j-1}$ (assume $r_0 = 0$). All the receivers in group j will listen to layer 1 up to layer j. The actual rate of each receiver is still equal to the optimal intra-session fair allocation while the total rate over a given link will be smaller than or equal to the bandwidth allocated. For the example in Figure 12,

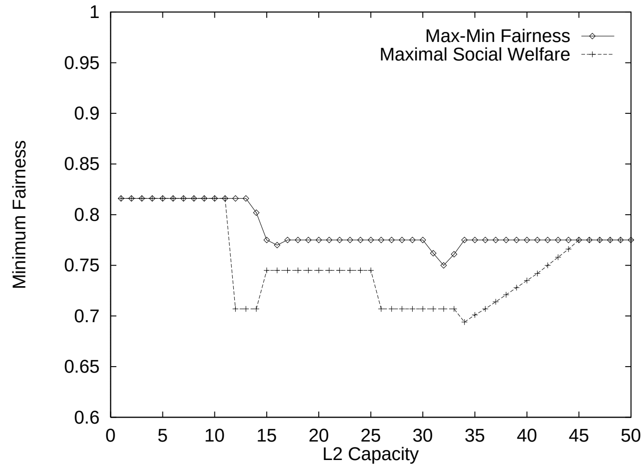

Figure 13: Minimum fairness

three layers will be set at rates 3, 3 ($= 6 - 3$), and 6 ($= 12 - 6$). Receivers $a1$ and $a2$ listen to layer 1, $a3$ and $a4$ listen to layers 1 and 2, $a5$ and $a6$ listen to layers 1, 2 and 3. They get the max-min intra-session fair rate. The rate used at each link is the sum of the rates of the layers going through that link. All three layers go through both links $L1$ and $L2$. Therefore, the rate used at $L1$ is 3 + 3 + 6 =12. This does not exceed the rate allocated to this session over link $L1$. The same is true for link $L2$, except that the total rate used is equal to the bandwidth allocated to this session over link $L2$.

5.1 Performance Results

We evaluate the performance of the max-min intra-session fair allocation and the maximal social welfare allocation in the environment where links are shared with receivers from other unicast/multicast sessions. The topology used is shown in Figure 11. The bandwidth for each link is labeled next to it, except that we vary the bandwidth of link $L2$ from 1 to 50. The goal is to use three layers to implement the multicast session with receivers $a1$ to $a6$. We show the results using the max-min fairness algorithm and the maximal welfare algorithm, with f_2 as the individual fairness function.

Figure 13 shows the minimum individual fairness value over all receivers. When the bandwidth of link $L2$ is less than 10, there is no difference because receivers $a1$, $a2$, and $a5$ are all constrained by link $L2$ and has the capacity of 3. When the bandwidth of $L2$ is between 10 and 45, we see significant differences between these two algorithms. Figure 14 shows the average fairness of receivers. We can see that the maximal welfare algorithm gets a higher average fairness value, or better social welfare. However, we find that better social welfare is achieved at the cost of being not so fair to some unlucky receivers in the session. Through these two plots, we can see the tradeoff between the max-min intra-session fairness and the maximal social welfare. While

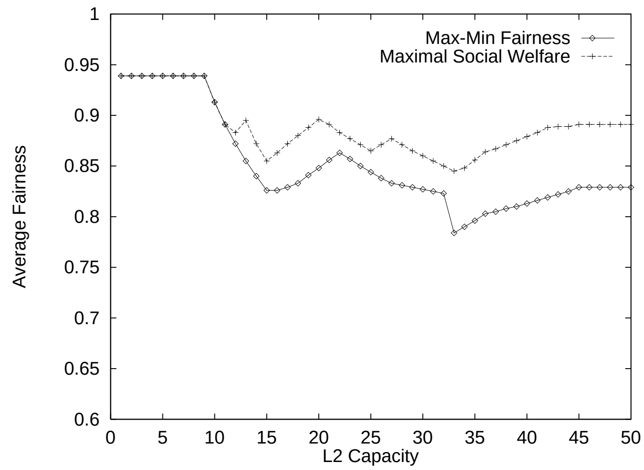

Figure 14: Average fairness

social welfare is an important measure for grouping multicast receivers and rate assignment, we cannot ignore the intra-session fairness if we want to be fair to receivers in the same session.

6 Conclusion

In this paper we gave a formal definition of intra-session fairness and illustrated its essential difference from social welfare. Intra-session fairness is a very important concept usually ignored in practice. In the case of single group sessions, we gave optimal solutions to the max-min intra-session fairness problem for various fairness function definitions to set an appropriate group rate. In the case of multiple group sessions, we presented an algorithm framework that can generate optimal solutions to the problem of grouping receivers to achieve the max-min intra-session fairness. The novelty of the solution is that it does not depend on any specific individual fairness function definition. We also discussed implementation issues and showed the difference in fairness achieved by the maximal social welfare and the max-min intra-session fair allocations.

The future work includes investigating the heterogeneous fairness function definitions. It is possible that different receivers want to use different fairness functions. The problem becomes much more complicated if the fairness function is in a complicated form, which may involve some special utility each receiver is interested in. Our perception is that we may not have a general solution and have to do an application specific analysis to find an intra-session fair solution to the problem.

Acknowledgment

This work was supported in part by the National Science Foundation under Grants CCR-0204304 and EIA-0101242, and a grant from the Kentucky Science and Engineering Foundation as per Grant Agreement #KSEF-148-502-05-139 with the Kentucky Science and Technology Corporation.

References

[Ammar and Wu, 1992] Ammar, M. H. and Wu, L.-R. (1992). Improving the throughput of point-to-multipoint ARQ protocols through destination set splitting. In *Proceedings of INFOCOM'92*, pages 262–71.

[Bhattacharyya et al., 1998] Bhattacharyya, S., Kurose, J., Towsley, D., and Nagarajan, R. (1998). Efficient rate-controlled bulk data transfer using multiple multicast groups. In *Proceedings of IEEE INFOCOM'98*, pages 1172–1179. San Francisco, CA.

[Cao and Zegura, 1999] Cao, Z. and Zegura, E. W. (1999). Utility max-min: An application-oriented bandwidth allocation scheme. In *Proceedings of IEEE Infocom'1999*, pages 793–801. New York, NY.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

[Fei and Yang, 2002] Fei, Z. and Yang, M. (2002). On max-min intra-session fairness in replicated multicast services. In *Proceedings of Fourth International Workshop on Networked Group Communication (NGC'02)*, pages 115–122. Boston, MA.

[Jaffe, 1981] Jaffe, J. M. (1981). Bottleneck flow control. *IEEE Transactions on Communications*, 29:954–962.

[Jiang et al., 1998] Jiang, T., Ammar, M. H., and Zegura, E. W. (1998). Inter-receiver fairness: A novel performance measure for multicast ABR sessions. In *Proceedings of ACM SIGMETRICS 98/Performance 98*, pages 202–211. Madison, Wisconsin.

[Kar et al., 2001] Kar, K., Sarkar, S., and Tassiulas, L. (2001). Optimization based rate control for multirate multicast sessions. In *Proceedings of IEEE Infocom'2001*, pages 133–141. Anchorage, Alaska.

[Kar et al., 2002] Kar, K., Sarkar, S., and Tassiulas, L. (2002). Low-overhead rate control for multirate multicast sessions. *IEEE Journal on Selected Areas in Communications*, 20:1541–1557.

[Kasera et al., 2000] Kasera, S., Hjalmtysson, G., Towsley, D., and Kurose, J. (2000). Scalable reliable multicast using multiple multicast channels. *IEEE/ACM Transactions on Networking*, 8:294–310.

[Kwon and Byers, 2004] Kwon, G.-I. and Byers, J. W. (2004). Leveraging single rate schemes in multiple rate multicast congestion control design. *IEEE Journal on Selected Areas in Communications*, 22:1975–1986.

[Rubenstein et al., 2002] Rubenstein, D., Kurose, J., and Towsley, D. (2002). The impact of multicast layering on network fairness. *IEEE/ACM Transactions on Networking*, 10:169–182.

[Stoenescu et al., 2003] Stoenescu, T. M., Liu, M., and Teneketzis, D. (2003). In *Proc. of IEEE Conference on Decision and Control (CDC)*, pages 2100 – 2105. Maui, Hawaii.

[Tzeng and Siu, 1997] Tzeng, H.-Y. and Siu, K.-Y. (1997). On max-min fair congestion control for multicast ABR service in ATM. *IEEE Journal on Selected Areas in Communications*, 15:545–556.

A Derivation of the Optimal Group Rate

The n receivers have capacities $c_1, c_2, \cdots, c_n$, with $c_i \leq c_{i+1}$ for $1 \leq i \leq n-1$. We want to find the group rate x, such the intra-session fairness defined as $\langle f(c_1, x), f_(c_2, x), \cdots, f(c_n, x)\rangle$ is maximized with regard to $\prec_v$.

Let $c_0 = -\infty$ and $c_{n+1} = +\infty$. Without loss of generality, we assume that $c_i < x \leq c_{i+1}$ for some $0 \leq i \leq n$.

For function f_1, the fairness vector of the group is $\langle c_1, c_2, \cdots, c_i, x, \cdots, x\rangle$. To maximize it, we need to set i to be n. That is, when $x = c_n$, the fairness is maximized with regard to $\prec_v$.

For function f_2, the fairness vector of the group is $\langle \frac{c_1}{x}, \frac{c_2}{x}, \cdots, \frac{c_i}{x}, \frac{x}{c_{i+1}}, \cdots, \frac{x}{c_n}\rangle$. The smallest element is either $\frac{c_1}{x}$ or $\frac{x}{c_n}$, because $\frac{c_j}{x} \geq \frac{c_1}{x}$ for $2 \leq j \leq i$, and $\frac{x}{c_j} \geq \frac{x}{c_n}$ for $i+1 \leq j \leq n-1$. If x is too small, $\frac{x}{c_n}$ will be too small. If x is too big, $\frac{c_1}{x}$ will be too small. To maximize the fairness vector, we set $\frac{c_1}{x} = \frac{x}{c_n}$. That is, $x = \sqrt{c_1 * c_n}$. Similar arguments can be made for function f_3.

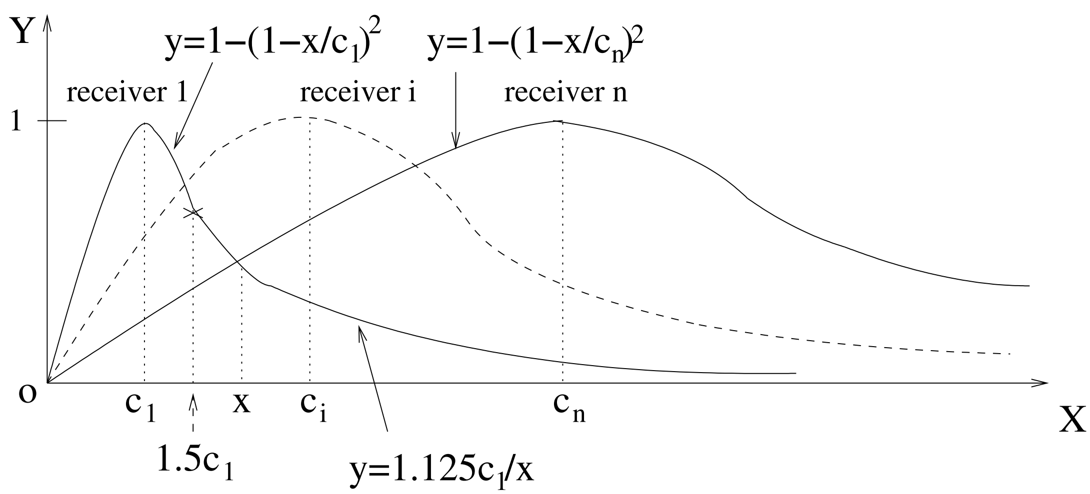

Figure 15: Solution for f4

For function f_4, we start with an observation that if $x < c_1$ or $x > c_n$, we will have a very small element in the fairness vector. Assume that the fairness function for receiver n intercepts with the fairness function of receiver 1 at x, as shown in Figure 15. The next question we want to answer is whether $x \leq 1.5c_1$ or $x > 1.5c_1$. To find out, we let $1 - (1 - x/c_1)^2 = 1 - (1 - x/c_n)^2$, and get $x = \frac{2*c_1*c_n}{c_1+c_n}$. In order for $x \leq 1.5c_1$, we have $\frac{2*c_1*c_n}{c_1+c_n} \leq 1.5c_1$, i.e., $c_n \leq 3c_1$. That is, when $c_n \leq 3c_1$, the function of receiver n intercepts with the function of receiver 1 at $x = \frac{2*c_1*c_n}{c_1+c_n}$, which make them have the same fairness value, while the fairness values of other receivers are greater than this value. If we set the group rate to some value other than x, it will make either the fairness value of receiver n or receiver 1 to be smaller than its current value, and thus make the overall fairness vector smaller than the current vector. Therefore, the group rate should be set to $\frac{2*c_1*c_n}{c_1+c_n}$ when $c_n \leq 3c_1$.

When $c_n > 3c_1$, the function for receiver n intercepts with the function for receiver 1 at some $x > 1.5c_1$. So we let $1.125\frac{c_1}{x} = 1 - (1 - \frac{x}{c_n})^2$, and get $x^3 - 2c_nx^2 + 1.125c_1c_n^2 = 0$. The optimal solution is to set the group rate to x, such that $x^3 - 2c_nx^2 + 1.125c_1c_n^2 = 0$ and $c_1 \leq x \leq c_n$.

For fairness functions $f_j'(c, r)$, if the group rate r is greater than c_1, it will cause the fairness value of receiver 1 to be 0. Therefore, we have to set the group rate r to c_1 to maximize the fairness vector with regard to $\prec_v$.

For fairness functions $h_j(c, \ell, r)$, if the solution to the corresponding f_j is bigger than $max(\ell_j)$, then the solution r_{f_j} is the solution to h_j. If r_{f_j} is smaller than $max(\ell_j)$, we have to set the group rate to $max(\ell_j)$ to ensure that no element in the fairness vector is zero. Therefore, we have to set the group rate r to $max(max(\ell_i), r_{f_j})$.

For fairness functions $h_j'(c, \ell, r)$, if $max(\ell_i) \leq c_1$, we can set the group rate to c_1 to ensure that no element in the fairness vector is zero. If $max(\ell_i) > c_1$, the optimal group rate will depend on f_j' and relative values of c_i's and ℓ_i's.