Optimal Partitioning of Multicast Receivers *

Y. Richard Yang, Min Sik Kim, Simon S. Lam

Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
E-mail: {yangyang,minskim,lam}@cs.utexas.edu

TR-2000-10
May 10, 2000

Abstract

Multicast sessions may have a large number of receivers with heterogeneous reception capacities. To accommodate this heterogeneity, various multi-rate schemes, based upon the use of layering or replication, have been proposed. We consider in this paper the optimal partitioning of receivers into groups for multi-rate schemes. For a general class of utility functions, we formulate the partitioning problem as an optimization problem to maximize the sum of receiver utilities. We present an efficient dynamic programming algorithm to solve the partitioning problem, and prove that the solution it finds is optimal. We also show that the majority of the benefit of a multi-rate scheme can be gained by using a small number of groups (or layers), say 4 to 5. To illustrate our solution approach, we apply it to the case where receiver capacities are determined by multi-rate max-min fair rates. A complete protocol for receiver rates computation, rates collection, optimal receiver partitioning, and receiver adaptation is presented. We then compare our approach with other multi-rate approaches as well as a single-rate approach. Experimental results show that our approach provides substantial performance improvements.

1 Introduction

A multicast session may have numerous receivers with heterogeneous data reception capacities. The reception capacity of a receiver may be limited by its own bottleneck or by the sender-to-receiver network path. To accommodate such heterogeneity, both single-rate and multi-rate approaches have been proposed.

In a *single-rate* approach [6, 24, 8, 2], the sender transmits at a fixed rate to all receivers. The rate is chosen either to conform to the slowest receiver or to maximize an inter-receiver fairness function [9]. Single-rate mechanisms, however, are inherently limited in accommodating significant heterogeneity and may not achieve certain desirable fairness properties [18].

In a *multi-rate* approach, the sender transmits at several rates to different sets of receivers using either a replicated scheme [5] or a layering scheme [19, 15, 13, 20, 12, 22].

In the replicated scheme [5], receivers are partitioned into groups. The sender generates and sends separate data streams to different groups. The sending rate to a group is adjusted according to the capacities of receivers in the group, and is restricted to be in some fixed range. Note that replicated schemes are not bandwidth efficient due to redundant data transmission whenever data streams to different groups share the same link [11].

In a layering scheme, the sender provides data in several layers organized in a hierarchy. Receivers subscribe to the layers cumulatively, i.e., if a receiver subscribes to layer k, it also subscribes to layers $1, \ldots, k-1$. In some layering schemes, the sending rate of each layer is assumed to be fixed. As a result, the sending rates may not match receiver capacities very well.

Recently, Jiang et al. studied an approach to partition receivers into groups and determine group transmission rates dynamically [10]. They formulated the problem of partitioning receivers as an optimization problem and proposed the use of three heuristics as guidelines for partitioning decisions, rather than finding an optimal one.

In this paper, we consider the partitioning of multicast receivers in a general framework, and present an efficient algorithm to find an optimal solution to the problem. For any multi-rate scheme, based upon the use of either layering or replication, the following two problems need to be addressed:

1. *How many groups?* A session with a large number of groups can accommodate wide receiver heterogeneity. However, a large number of groups would incur high overheads in sender encoding, multicast address allocation, network state, and receiver decoding. Therefore, the number of groups should be determined to

*Research sponsored in part by National Science Foundation grant No. ANI-9977267 and grant no. ANI-9506048. Experiments were performed on equipment procured with NSF grant no. CDA-9624082.

achieve as much benefit as possible without incurring excessive overheads.

2. *What are the sending rates?* For some applications, the group rates are determined regardless of receiver capacities; for example, in a layered transmission, the rates may be determined by encoding considerations. However, for applications with diverse and changing receivers, it is desirable to determine group or layer rates dynamically according to receiver capacities.

In this paper, we investigate and present solutions to these problems. We first consider a fixed number of groups and pose the problem of determining the sending rates for the groups as an optimal receiver partitioning problem. Specifically, let K denote the number of groups. Receivers are to be partitioned into groups $G_1, G_2, \ldots, G_K$ with the objective of maximizing the sum of receiver utilities. (The notion of utility is motivated and defined in Section 2. As an example, the utility of a receiver may be the data rate it receives.) We show that there is an efficient dynamic programming algorithm to find an optimal partition. For a given partition, an optimal group transmission rate g_k^* for the receivers in group G_k can be determined.

Our main results presented in Section 2 do not depend upon whether layering or replication is used in the multi-rate scheme. However, our implementation and experimental evaluation have been carried out for a layered approach (because layering is more bandwidth efficient than replication). In what follows, we use the term *group rate* for a particular group to denote the aggregate sending rate to a receiver in the group, and the term *layer rate* to denote the rate of a particular layer in a layered transmission. Suppose the receivers are partitioned into K groups, with group rates $g_1^* \leq g_2^* \leq \cdots \leq g_K^*$. Then there would be K transmission layers, with layer rates $g_1^*, g_2^* - g_1^*, \ldots, g_K^* - g_{K-1}^*$. A receiver in group k subscribes to layers from 1 to k and receives an aggregate data rate of g_k^*.

We also evaluate the impact of the number of groups on the utility of a session. We show that for a session of a reasonable size, the majority of the benefit of a multi-rate scheme can be gained even with a small number of groups, say 4 to 5.

To demonstrate the usage of our method for optimal receiver partitioning, we apply it to the case where receiver capacities are determined by multi-rate max-min fair rates. A complete protocol for rates computation, rates collection through aggregation, optimal receiver partition, and receiver adaptation is presented. We also compare our approach with other multi-rate approaches as well as a single-rate scheme. Experimental results show that our approach provides substantial performance improvements over the other approaches.

The balance of this paper is organized as follows. In Section 2 we formulate the optimal receiver partition problem. We present an efficient algorithm for its solution and prove that the solution is optimal. In Section 3, we apply it to the case where receiver "isolated" rates are their multi-rate max-min fair rates. In Section 4, we show experimental evaluation results and compare with other schemes. We conclude in Section 5.

2 The Optimal Partitioning Problem

2.1 Problem formulation

We first consider a predetermined number of groups. The impact of the number of groups is evaluated in the last subsection. Before defining the problem, we first introduce some terms. Let N denote the number of receivers and K the number of groups.

Receiver partition Given receivers $\{1, 2, \ldots, N\}$, a set $P = \{G_1, G_2, \ldots, G_K\}$ is said to be a *receiver partition* if P is a partition of $\{1, 2, \ldots, N\}$.

Isolated rate This is the reception rate of a receiver in a multicast session if there is no constraint from other receivers in the same session. The reception rate may be limited by a bottleneck in the receiver itself, or by the sender-to-receiver network path. Let r_i denote the isolated rate of receiver i. For ease of presentation, the receivers are numbered such that their isolated rates are sorted in nondecreasing order, i.e., $r_1 \leq r_2 \leq \cdots \leq r_N$.

Loss tolerance L This specifies the largest loss rate a receiver can tolerate. Consider a receiver with isolated rate r. If the sending rate to this receiver is $x > r$, then it is possible for the receiver to have a loss rate of $\frac{x-r}{x}$. A loss tolerance of L requires $\frac{x-r}{x} \leq L$. Rearranging, we have $x \leq \frac{r}{1-L}$. Therefore, the receiver should not be included in a group with sending rate higher than $\frac{r}{1-L}$.

Receiver utility function $u(r, g)$ The utility of a receiver is determined by its isolated rate r and the sending rate g of its group. If g is equal to r, the receiver's capacity is fully utilized. Otherwise the capacity of the receiver is either underutilized or overutilized.

Group utility $U(G_k, g)$ For a group G_k with sending rate g, its utility $U(G_k, g)$ is defined to be $\sum_{i \in G_k} u(r_i, g)$.

Optimal group transmission rate $g^*(G_k)$ For a group G_k, its optimal transmission rate $g^*(G_k)$ is defined to be the rate that maximizes the group utility subject to the constraint that the loss tolerance of every group member is satisfied, i.e.,

$$U(G_k, g^*(G_k)) = \max_{0<g\leq\frac{\min_{i\in G_k} r_i}{1-L}} U(G_k, g)$$

When multiple rates exist that satisfy the above requirement, $g^*(G_k)$ is chosen to be the smallest.

Maximum group utility $U^*(G_k)$ This is the value of $U(G_k, g^*(G_k))$.

Session utility $V(P)$ **of a partition** $P = \{G_1, G_2, \ldots, G_K\}$ This is the sum of maximum group utilities, $\sum_{k=1}^{K} U^*(G_k)$.

Definition 1 (Optimal Receiver Partition) *Given a set of receivers with isolated rates $r_1, r_2, \ldots, r_N$, K groups, receiver utility function $u(r, g)$, and loss tolerance L, a receiver partition $P^* = \{G_1^*, G_2^*, \ldots, G_K^*\}$ is an* optimal receiver partition *if $V(P^*) \geq V(P)$ for any receiver partition P.*

It is obvious that the optimal solution will depend upon the receiver utility function $u(r, g)$. Before we proceed to present an algorithm to find an optimal receiver partition, we first discuss what is desired of the function $u(r, g)$, defined for $r \geq 0$ and $g \geq 0$. Intuitively, the closer are r and g to each other, the larger should $u(r, g)$ be. And the farther apart are r and g, the smaller should $u(r, g)$ be. In particular, $u(r, g)$ should achieve its maximum value when $r = g$. This intuition is formalized below.

Receiver Utility Property For a fixed r, $u(r, g)$ is nondecreasing in the interval $[0, g]$ and nonincreasing in the interval $[g, \infty)$. For a fixed g, $u(r, g)$ is nondecreasing in the interval $[0, r]$ and nonincreasing in the interval $[r, \infty)$.

2.2 Optimal partition and ordered optimal partition

A straightforward way to find the optimal partition is to enumerate all possible partitions and choose the one that maximizes the session utility V. This is impractical because of the required computational complexity.

For a better solution, we first define an ordered receiver partition and ordered optimal receiver partition:

Definition 2 (Ordered Receiver Partition) *A partition $P = \{G_1, G_2, \ldots, G_K\}$ is an* ordered *receiver partition if r_i, for all $i \in G_k$, is less than or equal to r_j, for all $j \in G_l$, for any two groups G_k and G_l, where $k < l$.*

Definition 3 (Ordered Optimal Receiver Partition) *A partition $P^* = \{G_1^*, G_2^*, \ldots, G_K^*\}$ is an* ordered optimal *receiver partition if $V(P^*) \geq V(P)$ for any ordered receiver partition P.*

In the next subsection, we present an efficient algorithm to find an ordered optimal partition. Furthermore, by the following theorem, Receiver Utility Property guarantees that there exists an optimal partition which is also ordered. Thus, the ordered optimal receiver partition found by our algorithm will indeed be optimal over all possible partitions.

Theorem 1 *For any receiver utility function $u(r, g)$ satisfying Receiver Utility Property, there exists a receiver partition that is both ordered and optimal.*

A proof is shown in Appendix A. By Theorem 1, an ordered optimal partition is also an optimal partition.

2.3 Dynamic programming algorithm to find an ordered optimal partition

In this section, we describe a dynamic programming algorithm to find an ordered optimal partition.

Let $V^*(i, m)$ denote the maximum session utility V of an ordered partition of receivers $1, \ldots, i$ into m groups. The key observation is that

$$V^*(i, m) = \max_{1\leq j<i} [V^*(j, m-1) + U^*(\{j+1, \ldots, i\})]$$

This equation is in the classic form solvable using a dynamic programming algorithm. Algorithm details are shown in Figure 1. Note that this algorithm is applicable to any optimal receiver partition problem when the receiver utility function $u(r, g)$ satisfies Receiver Utility Property.

```
OPTIMALPARTITION(K; r_1, r_2, ..., r_N)
▷ Assume that r is sorted in nondecreasing order.
for i ← 1 to N
    V*(i,1) ← U*({1,...,i})
    p(i,1) ← −1
for m ← 2 to K
    for i ← 1 to N
        V*(i,m) ← V*(i,m−1)
        p(i,m) ← i
        for j ← 1 to i−1
            v ← V*(j,m−1) + U*({j+1,...,i})
            if V*(i,m) < v then
                V*(i,m) ← v
                p(i,m) ← j
j ← N
for m ← K downto 1
    i ← j
    j ← p(i,m)
    G_m ← {j+1,...,i}
```

Figure 1. Optimal Partition Algorithm

The complexity of the algorithm is $O(KN^2)$ if $U^*(\{j, j+1, \ldots, i\})$ is pre-computed for all j and $i, j \leq i$. However, to compute $U^*(\{j, j+1, \ldots, i\})$, we need to know the definition of $u(r, g)$. Moreover, the complexity also depends on rate constraints from loss tolerance L and from the set of possible sending rates. In the next subsection, we first consider the case when the receiver utility function is piecewise convex. We consider partitioning with constraints in Section 2.5.

2.4 Computing group utility for piecewise convex functions

We first study the following utility function,

$$u_{\mathrm{IRF}}(r, g) = \frac{\min(r, g)}{\max(r, g)} \qquad (1)$$

which was first defined in [9]. The authors introduced it in the context of *Inter-Receiver-Fairness* (IRF); therefore, we refer to this utility function as $u_{\mathrm{IRF}}(r, g)$. Figure 2 shows the shape of the utility function. It is clear that $u_{\mathrm{IRF}}(r, g)$

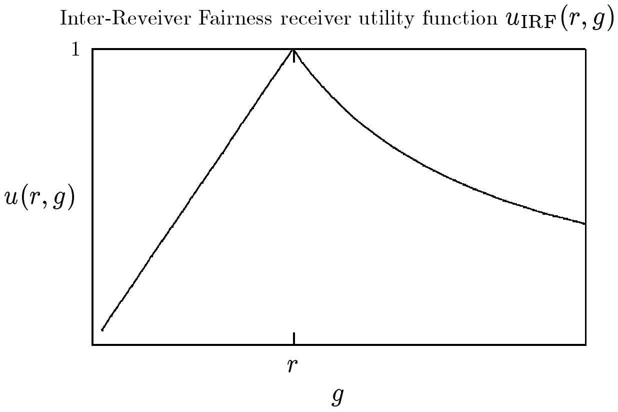

Figure 2. Receiver utility function $u_{\mathrm{IRF}}(r, g)$

satisfies Receiver Utility Property. Hence, we can apply our dynamic programming algorithm to solve it.

Further, observe that $u_{\mathrm{IRF}}(r, g)$ is a convex function of g in intervals $[0, r]$ and $[r, \infty)$. For this type of utility functions Theorem 2 says that we can compute $U^*(G)$ by considering only the points where g is equal to one of the isolated rates in group G.

Theorem 2 *If G is given and the receiver utility function $u(r, g)$ is a convex function of g in interval $[0, r]$ and in interval $[r, \infty)$, $U(G, g)$ is maximized when g is equal to one of the rates in $\{r_i : i \in G\}$.*

A proof is shown in Appendix B.

Therefore, we know that for a given group G, $U(G, g)$ has a maximum value when g is equal to the isolated rates of one of the receivers in group G. Thus, a straightforward way to determine the optimal group sending rate $g^*(\{j, \ldots, i\})$ is to compute the value of $U(\{j, \ldots, i\}, r_k)$ for all k, $j \leq k \leq i$, and choose the largest one. Then the time complexity to calculate $U^*(\{j, \ldots, i\})$ for each pair of j and i is $O(N^2)$, and the total complexity for all j, i pairs becomes $O(N^4)$ if carried out in a straightforward manner.

However, we have found a more efficient algorithm by removing redundant computations. The details of the algorithm are shown in Figure 3. For this algorithm pre-computation takes $O(N^3)$ and OPTIMALPARTITION() takes $O(KN^2)$. Therefore, the total complexity is $O(N^3 + KN^2)$. Since K is much smaller than N, the total complexity is $O(N^3)$.

```
COMPUTEGROUPUTILITY(r_1, r_2, ..., r_N)
▷ Assume that r is sorted in nondecreasing order.
for i ← 1 to N
    M(i,i) ← 0
    for j ← i + 1 to N
        M(i,j) ← M(i,j − 1) + u(r_j, r_i)
    M(i,i) ← u(r_i, r_i)
    for j ← i − 1 downto 1
        M(i,j) ← M(i,j + 1) + u(r_j, r_i)
for i ← 1 to N
    m ← 0
    for j ← i to N
        G ← {i, ..., j}
        m ← m + u(r_j, r_i/(1 − L))
        g*(G) ← r_i/(1 − L)
        U*(G) ← m
        for k ← i to j
            if r_k > r_i/(1 − L) then break
            u ← M(k,i) + M(k,j)
            if U*(G) < u then
                U*(G) ← u
                g*(G) ← r_k
```

Figure 3. Algorithm to compute group utilities

2.5 Computing group utility for discrete sending rates and zero loss tolerance

We first consider the constraint of discrete sending rates. The algorithm in Figure 3 assumes that the sender can control its sending rate with fine granularity. While some encoding schemes such as PVH [14] and H.261 [3] can control sending rates by adjusting refresh rate, quantizer, and movement detection threshold, some others such as

WWHVQ [23] cannot send at an arbitrary rate. For example, the sending rates can only be 16kbps, 32kbps, 64kbps, 128kbps, and so on, or they must be multiples of a fixed value like 16kbps. When using $u_{\mathrm{IRF}}(r, g)$, our algorithm chooses one of the isolated rates as a group rate, which may not be one of the possible sending rates. Next, we show that our algorithm can handle this constraint with just minor modifications.

Consider any group $G = \{j, \ldots, i\}$, where $j < i$. We know from the proof in Appendix B that $U(G, g)$ is a convex function of g in any interval $[r_k, r_{k+1}]$, where $j \leq k < i$. Suppose there are N_k possible sending rates in the interval $[r_k, r_{k+1}]$, $R_{k,1} \leq R_{k,2} \ldots \leq R_{k,N_k}$. Then we know $U(G, g)$ is convex in $[R_{k,1}, R_{k,N_k}]$. Therefore, $U(G, g)$ will have its maximum value when $g = R_{k,1}$ or $g = R_{k,N_k}$. Thus, in order to find the optimal sending rate $g^*(G)$, we should compute $U(G, R_{k,1})$ and $U(G, R_{k,N_k})$ instead of $U(G, r_k)$ in the algorithm. Although the required amount of computation doubles with this modification, the time complexity remains the same.

We next consider the loss tolerance constraint. We note that under certain circumstances it is undesirable for the group rate to be higher than the isolated rates of some receivers, not only because these receivers experience packet loss, but also because it hurts network fairness by taking bandwidth from other sessions. If this is considered to be important, we can specify L to be 0 so that the optimal group rate is always the lowest isolated rate in the group. As a result, we do not need the innermost loop in COMPUTEGROUPUTILITY(), and the time complexity to compute group utilities is reduced to $O(N^2)$. The overall complexity of optimal receiver partitioning is reduced to $O(KN^2)$.

Next we consider the case including both zero loss tolerance and sending rate constraints. Figure 4 shows an example of a partitioning of receivers with sending rate constraints. Given zero loss tolerance, the group rate should be the largest possible sending rate that is smaller than the lowest isolated rate in the group.

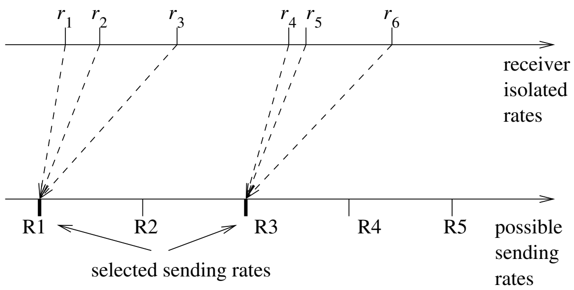

Figure 4. A receiver partition with sending rate constraints

Clearly we can still use the utility function $u_{\mathrm{IRF}}(r, g)$ when there are both types of constraints. However, to demonstrate the versatility of our framework, we next introduce a different utility function. One possible interpretation of the utility of a multicast session is the total received bandwidth. To achieve this session utility, the receiver utility function can be defined as

$$u(r, g) \triangleq u_{\mathrm{rate}}(r, g) = \min(r, g) \qquad (2)$$

Figure 5 shows the shape of this utility function. It is easy to see that with this utility function and zero loss tolerance the session utility is the sum of the bandwidth received at all receivers.

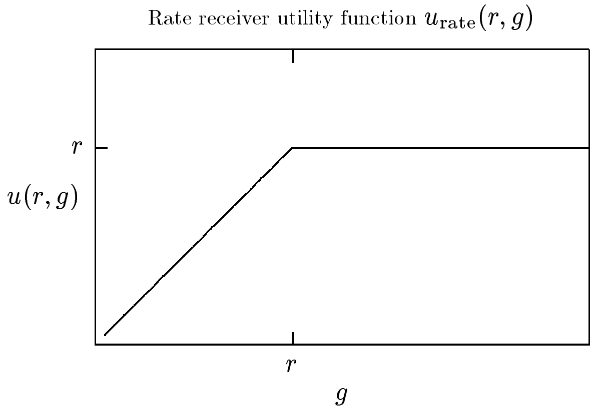

Figure 5. Receiver utility function $u_{\mathrm{rate}}(r, g)$

For the rate utility function, the partitioning algorithm can be simplified by first changing each receiver isolated rate to the largest possible sending rate just below it. Observe that this change will not affect the utility of any receiver. Then we apply our algorithm to find the optimal receiver partition. Observe also that we can compute the group utilities in OPTIMALPARTITION() without precomputation as follows:

$$U^*(\{j+1, \ldots, i\}) = r_{j+1} \cdot (i - j)$$

Although it does not require COMPUTEGROUPUTILITY(), the time complexity of optimal receiver partitioning is still $O(KN^2)$.

2.6 How many groups?

In the previous subsections, the number of groups is assumed to be given and fixed. We next study quantitatively the impact of the number of groups on session utility.

To quantify this impact, for a given configuration of receivers, we define

$$Q_K = \frac{V^*(K; r_1, \ldots, r_N)}{V^*(N; r_1, \ldots, r_N)}$$

where $V^*(K; r_1, \ldots, r_N)$ is the maximum session utility when we have K groups. With Receiver Utility Property, it is easy to see that the number of groups needed to achieve the maximum utility is N, the number of receivers.

Q_K depends on the distribution of receiver isolated rates. In our experiments, we used the following distributions:

- A uniform distribution over the interval $[1, 10)$.
- A normal distribution with mean 5 and standard deviation 2.
- A bi-modal distribution consisting of two normal distributions. The means are 2 and 8 respectively, and the variance is 1 in both distributions. One third of the receivers belong to the first normal distribution.

Figure 6 demonstrates how much improvement we can achieve through increasing the number of groups for each distribution of 300 receivers. We used $u_{\text{IRF}}(r, g)$ as the receiver utility function in (a), and $u_{\text{rate}}(r, g)$ in (b). Q_K increases in each case as we add more groups. However, most of the benefit is obtained with about 4 groups. We believe that applications are unlikely to provide more than 5 groups because of encoding complexity, e.g., the number of filters required. Also, the higher is the number of groups, the higher the network overhead will be to keep routing states. Therefore, we think 4 or 5 is the most reasonable number of groups for most applications. With 4 groups, our algorithm achieves about 80% of the maximum utility.

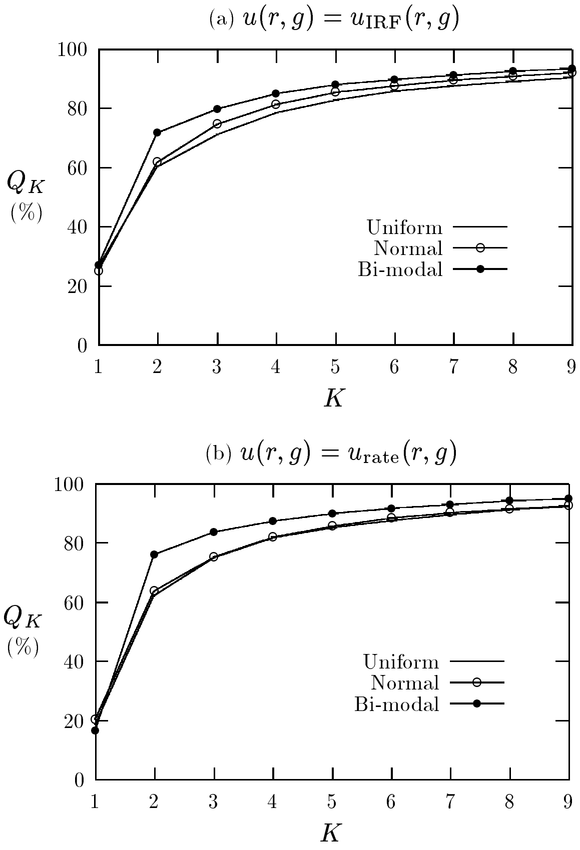

Figure 6. Effects of the number of groups

3 Protocol for Max-min Fair Rates

The optimal partitioning algorithm presented in the previous section is applicable to any network environment where receivers can compute their isolated rates; it does not depend on how to compute those rates.

Receiver isolated rates are generally determined by receiver device constraints or network fairness requirements. As an example, we can define the isolated rate of a receiver to be its TCP-fair rate. In this case, the receiver measures the loss rate and round-trip time between itself and the sender, and calculates its TCP-fair rate using a TCP throughput formula [16, 7]. In a high multiplexing environment, the rate calculated this way would be accurate because the loss rate is independent of the sending rate. However, in a low multiplexing environment, the loss rate would depend on the sending rate, and an iterative algorithm is required to find TCP-fair rates.

In the experiments to be presented, we consider the isolated rate of a receiver to be its multi-rate max-min fair rate [18].

Our protocol consists of four steps: isolated rates computation, rates collection, optimal receiver partitioning, and receiver adaptation. In the following subsections we specify each step of the protocol. Figures 7, 8, and 9 summarize the sender, router, and receiver algorithms.

3.1 Isolated rates computation

We use the algorithm proposed in [17] to calculate max-min fair rates. The sender initiates the protocol every T seconds by multicasting a message $\langle$*compute-rates*;$r\rangle$, where r is its desired sending rate. When sending r for the first time, the sender sets r to its maximum sending rate. Once it has received feedbacks from all receivers, it adjusts r (to be used in the next round). If the highest reported rate is equal to the r just sent, it indicates that some receivers are likely to have higher isolated rates than the reported rate, and the sender increases r.

The multicast message $\langle$*compute-rates*;$r\rangle$ is propagated

▷ M is the number of intervals.
▷ I_i is the ith interval [left, right].
▷ n_i is the number of receivers in the ith interval.

Upon every T seconds:
    Determine the desired rate r.
    Multicast $\langle$*compute-rate*;$r\rangle$ to all children.

Upon receiving $\langle$*rate intervals*;$(I_1, n_1), \dots, (I_M, n_M)\rangle$:
    Mark the child as received.
    Store the message.
    if all children are marked **then**
        Retrieve all messages and sort rates into r_i, $1 \le i \le N$ assuming uniform distribution in each interval.
        Call COMPUTEGROUPUTILITY$(r_1, \dots, r_N)$ and OPTIMALPARTITION$(K; r_1, \dots, r_N)$.
        Unmark all children.
    Multicast $\langle$*group-rates*;$g_1, \dots, g_K\rangle$.

Figure 7. Sender Algorithm

throughout the multicast tree. Each intermediate router computes the max-min fair share α for each downstream link [21], and, if α is less than r, replaces r with α before it forwards the message through that link.

When a receiver gets $\langle$*compute-rates*;$r\rangle$, it knows that r is its max-min fair rate. It assigns $[r, r]$ to each of the intervals I_i, $1 \le i \le M$, and sets the number of receivers n_1 for I_1 to be 1. The number of receivers in every other interval $n_i, 1 < i \le M$ is set to 0.

3.2 Rates collection

Receivers report their isolated rates to their parent routers in the multicast tree. However, if there is a large number of receivers, these messages should be aggregated to avoid feedback implosion. When a router receives interval reports from all its children, it distributes all the reported intervals into M new intervals, assuming uniform distribution.

3.3 Optimal partitioning

After receiving rates from all child nodes, the sender determines the optimal partition and the group rates using the algorithm proposed in Section 2. The determined rates are sent to all receivers by multicast so that they can make decisions to join the appropriate layers.

Upon receiving $\langle$*compute-rate*;$r\rangle$:
    Calculate the fair share α.
    Multicast $\langle$*compute-rate*;$\min(r, \alpha)\rangle$ to all children.

Upon receiving $\langle$*rate intervals*;$(I_1, n_1), \dots, (I_M, n_M)\rangle$:
    Mark the child as received.
    Store the message.
    if all children are marked **then**
        Retrieve all messages and distribute rates into new intervals assuming uniform distribution in each interval.
        Send $\langle$*rate intervals*;$(I'_1, n'_1), \dots, (I'_M, n'_M)\rangle$ to the parent.
        Unmark all children.

Figure 8. Router Algorithm

Upon receiving $\langle$*compute-rate*;$r\rangle$:
    $r_{\text{isolated}} \leftarrow r$
    $I_i \leftarrow [r, r]$ for all i, $1 \le i \le M$.
    Send $\langle$*isolated-rates*;$(I_1, 1), (I_2, 0), \dots, (I_M, 0)\rangle$ to the parent.

Upon receiving $\langle$*group-rates*;$g_1, \dots, g_K\rangle$:
    for $k \leftarrow 1$ **to** K
        if $g_k \le r_{\text{isolated}}/(1 - L)$ **then**
            Subscribe to layer k.
        else
            Unsubscribe to layer k.

Figure 9. Receiver Algorithm

3.4 Receiver adaptation

A receiver compares its isolated rate with the group rates announced by the sender, and joins a layer such that its loss tolerance will not be exceeded.

4 Experimental Evaluation

In this section we first examine errors introduced by aggregation at routers, and then compare our optimal partitioning algorithm (using the protocol in the previous section) with other approaches.

For all experiments in this section, we show results for two different receiver utility functions, $u_{\text{IRF}}(r, g)$ and $u_{\text{rate}}(r, g)$, defined in equations (1) and (2), respectively. The loss tolerance L is set to 0 because a nonzero L makes some receivers use more bandwidth than their max-min fair rates and violates max-min fairness.

4.1 Aggregation accuracy

Our partitioning algorithm finds the optimal rates for groups (layers) given receiver isolated rates. However, since routers in our protocol aggregate reported rates for scalability, it may introduce inaccuracy. In this section, we examine aggregation accuracy for different distributions of isolated rates.

Because a router assumes a uniform distribution of rates in an interval, the aggregation is accurate for flat distributions. For example, when isolated rates distribute uniformly, the inaccuracy was less than 3% even with only one interval. But uniform distribution may not always be the case. In most networks, the actual distribution is more likely to have several peaks because receivers are grouped by their connection types, e.g., dial-up modems, ISDN, DSL, etc.

We next evaluate the effect of distributions on aggregation accuracy. We use the same normal distribution and bi-modal distribution as in Section 2.6. The aggregation accuracy is measured by the difference between maximum session utility achieved with aggregation, $V_{\rm ag}$, and that without aggregation, $V_{\rm nag}$. Figure 10 shows the errors, $\frac{V_{\rm nag}-V_{\rm ag}}{V_{\rm nag}}$, induced by aggregation. In this experiment, the number of receivers is 200, and the number of groups is 4.

Although the results in Figure 10 have some randomness, it is clear that the error decreases as we increase the number of intervals. From the figure we can see that 4 or 5 is a reasonable number of intervals to use, because the error is less than 3% with 4 intervals and further increase would not have enough benefit to justify the additional overhead imposed on routers.

4.2 Comparison with other approaches

In this section, we first highlight the differences between our approach and two other approaches: one layering and one replicated. Then we compare the achieved session utilities of our approach and other approaches.

Most layering schemes use fixed layer rates. Shacham [19] presented a method to assign bandwidth to layers to maximize the average signal quality of receivers. Although he did not explicitly define the concept of utility function, his average signal quality can be considered an instance of our utility function and thus his paper was possibly the first to address the problem of partitioning receivers. He did not, however, adequately address optimality because he did not try to show the existence of an optimal partition that is ordered. In comparison, we solved the optimal receiver partitioning problem for a general class of utility functions. Our Theorem 1 shows the existence of an optimal partition that is ordered, which gives rise to efficient algorithms to find an optimal partition based upon dynamic programming.

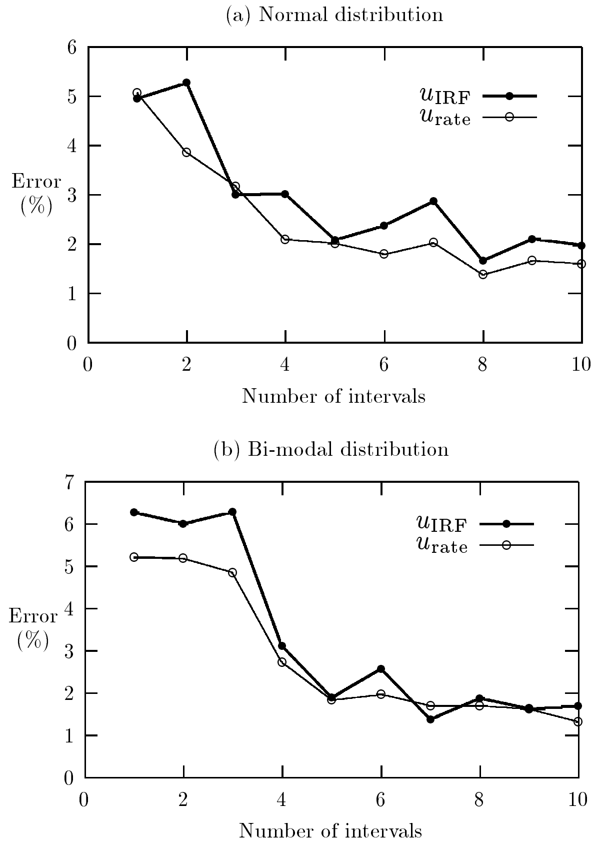

Figure 10. Aggregation Accuracy

Among replicated schemes, the representative approach is DSG [1, 4]. Jiang et al. extended DSG to partition receivers and determine group transmission rates dynamically [10]. They formulated the problem of partitioning receivers as an optimization problem by using receiver utility function $u_{\rm IRF}(r, g)$. However, they used three heuristics as guidelines for partitioning decisions, rather than finding an optimal one. It is easy to come up with examples that do not work well using their heuristics. From Section 2 observe that we solved their problem optimally as a special case.

4.2.1 Comparison with a single-rate scheme and a layering scheme with equal partition

In this section, we compare our approach with a single-rate scheme and a layering scheme with equal partition. The sending rate of the single-rate scheme is the isolated rate of the slowest receiver, $r_{\rm min}$. For the layering scheme with equal partition the first (base) layer has rate $r_{\rm min}$; the rates of the other layers are $\frac{r_{\rm max}-r_{\rm min}}{K}$, where $r_{\rm max}$ is the largest isolated rate. This scheme is better than sending at a fixed rate because it adapts to receiver capacities.

In the experiments presented below, the receiver isolated rates have the same distributions as in the aggregation accuracy experiments. The number of receivers is 200, and the number of layers is 4. We ran simulations for receiver utility functions $u_{\mathrm{IRF}}(r, g)$ and $u_{\mathrm{rate}}(r, g)$.

Figure 11 shows the achieved session utilities of the three layered approaches: equal partition, optimal partition with aggregation, and optimal partition without aggregation. The optimal parition without aggregation scheme gives the highest achievable session utility.

Not shown in the figures are the session utilities achieved with the single-rate scheme. The session utilities of the single-rate scheme are 50.2, 200, 54.6, and 200 for (a), (b), (c), and (d), respectively, which are substantially lower than session utilities of the layered approaches. Both optimal partition and equal partition achieve a session utility more than three times of the utility achieved by the single-rate scheme. This is expected because the session utility of the single-rate scheme is determined by the slowest receiver, therefore its utility is usually much lower than that of layering unless the isolated rates are very close to each other.

Notice that equal partition would be an optimal partition if the isolated rates are distributed uniformly. Since this is not the case in general, there is always a gap between the highest achievable utility and the utility obtained with equal partition. We can clearly see the gaps in the figures of Figure 11. This gap becomes bigger as the distribution of isolated rates deviates more from the uniform distribution. The optimal partition (with aggregation) achieves utility in this gap and outperforms the equal partition for both receiver isolated rate distributions. Furthermore, even with a small number of intervals for aggregation, our approach achieves a utility close to the maximum.

4.2.2 Comparison with replicated scheme

To compare our approach with the replicated scheme DSG, we use the same network topology as the one in [10], which is shown in Figure 12. A is the sender of a multicast session, and A_1, A_2, and A_3 are receivers. B and C are sets of two and nine unicast sessions, respectively.

Table 1 shows the sending rates of all sessions with different multicast mechanisms. It also shows V_{IRF} and V_{rate} of the multicast session A, which are the session utilities for two different receiver utility functions $u_{\mathrm{IRF}}(r, g)$ and $u_{\mathrm{rate}}(r, g)$.

The first row is for the single-rate case where the multicast rate is determined by the lowest rate among all receivers' rates. The table has two rows for the 2 layer optimal partition because there are two ways to partition three receivers into two groups: the first one maximizes V_{IRF} and the second maximizes V_{rate}.

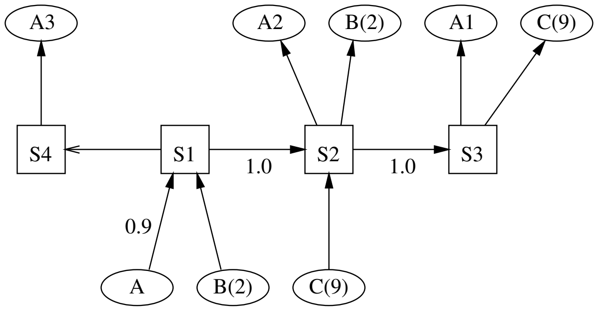

Figure 12. Network Topology

Although both the replicated and layering schemes significantly improve utilities compared with the single-rate scheme, layering always achieves a higher utility because, with a replicated scheme, each receiver utilizes only a portion of the bandwidth in links shared by several groups. Consider the case where receivers A_1, A_2, and A_3 are partitioned into three groups. In our topology, the link between A and S_1 is shared by all three groups with shares of 0.10, 0.23, and 0.57. In a replicated scheme, since a receiver can join only one group, its achieved rate is limited by these rates. However, in a layering scheme, a receiver can subscribe to more layers incrementally as long as the cumulative rate does not exceed its isolated rate.

Figure 12 is a very simple topology. For a more complex topology where there are many receivers sharing links, the benefit of our approach is even more pronounced.

5 Conclusion

To accommodate multicast sessions with diverse receiver capacities, various multi-rate schemes, based upon the use of layering or replication, have been proposed. We considered the optimal partitioning of receivers into groups for multi-rate schemes. For a general class of utility functions, we formulated the partitioning problem as an optimization problem to maximize the sum of receiver utilities. We proved that there exists an an optimal partition that is ordered (Theorem 1). We then presented an efficient dynamic programming algorithm to find an optimal ordered receiver partition. We also showed that the majority of the benefit of a multi-rate scheme can be gained by using a small number of groups (or layers), say 4 to 5.

To illustrate our solution approach, we applied it to the case where receiver capacities are determined by multi-rate max-min fair rates. A complete protocol for receiver rates computation, rates collection, optimal receiver partitioning, and receiver adaptation was designed. For scalability, receiver rates information is aggregated at routers. We showed that error introduced by aggregation is smaller

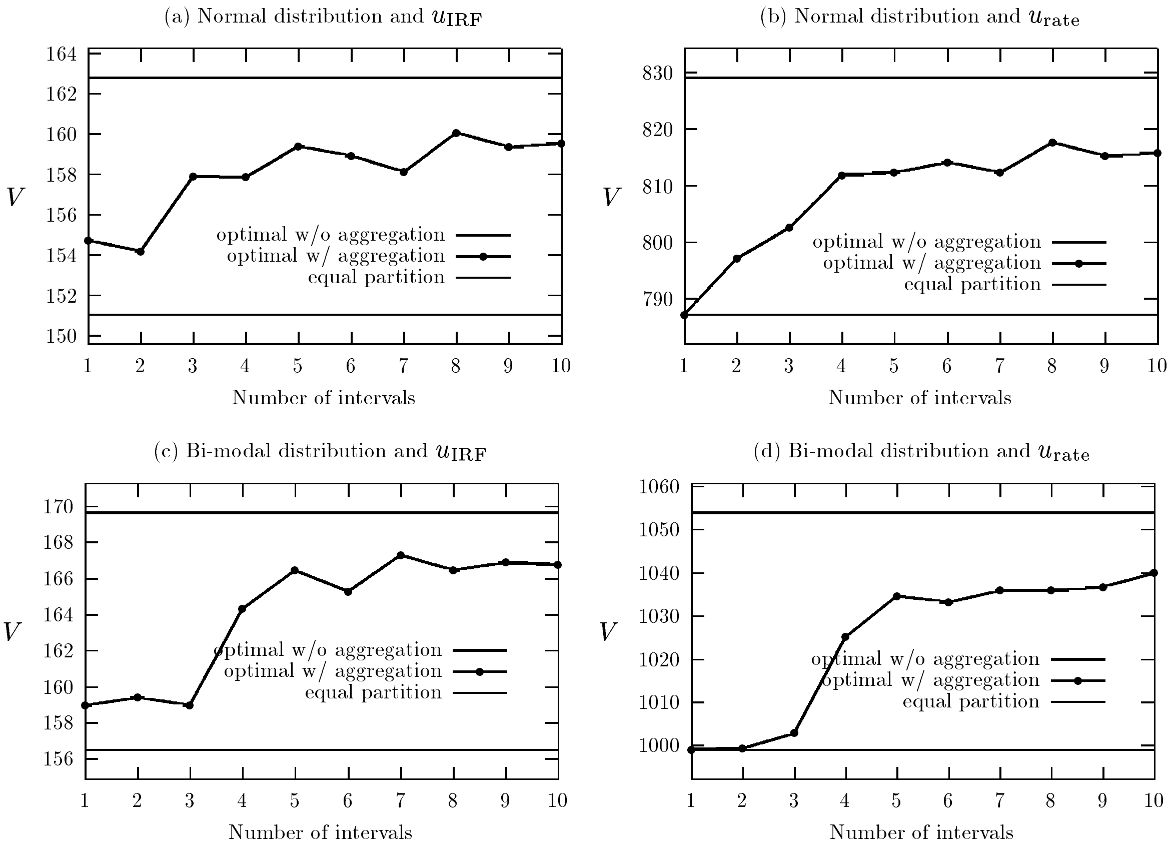

Figure 11. Comparison of layering schemes

Mechanism	Sending rates					$V_{\text{IRF}}(A)$	$V_{\text{rate}}(A)$
	A_1	A_2	A_3	B	C		
single-rate	0.10	0.10	0.10	0.45	0.10	1.41	0.30
DSG: $\{A_1\}, \{A_2, A_3\}$	0.10	0.23	0.23	0.33	0.10	1.95	0.56
2 layers	0.10	0.33	0.33	0.33	0.10	2.37	0.77
	0.10	0.10	0.90	0.33	0.10	2.30	1.10
DSG: $\{A_1\}, \{A_2\}, \{A_3\}$	0.10	0.23	0.57	0.33	0.10	2.33	0.90
3 layers	0.10	0.33	0.90	0.33	0.10	3.00	1.33

Table 1. Comparison with replicated scheme

than 3% when more than 4 aggregation intervals are used. We then compared our approach with other multi-rate approaches as well as a single-rate approach. Experimental results showed that our approach provides substantial performance improvements.

Acknowledgments

The authors would like to thank Mr. Xincheng Zhang for interesting discussions.

A Proof of Theorem 1

First, we restate Receiver Utility Property using formal notation as follows:

1. $\forall g, \forall i, \forall j, 1 \leq i, j \leq N$,

$$\begin{array}{lcl} r_i < r_j \leq g & \rightarrow & u(r_i, g) \leq u(r_j, g) \\ g \leq r_i < r_j & \rightarrow & u(r_i, g) \geq u(r_j, g) \end{array}$$

2. $\forall r, \forall i, \forall j, 1 \leq i, j \leq K$,

$$\begin{array}{lcl} g_i < g_j \leq r & \rightarrow & u(r, g_i) \leq u(r, g_j) \\ r \leq g_i < g_j & \rightarrow & u(r, g_i) \geq u(r, g_j) \end{array}$$

Consider an optimal partition P and choose any two groups G_a and G_b in P, whose group rates are g_a and g_b, where $g_a \leq g_b$. Let $r_m = \max_{k \in G_a} r_k$, and $r_n = \min_{k \in G_b} r_k$. If $r_m \leq r_n$ for any two groups, G_a and G_b, then P is already ordered. Otherwise there are three possibilities. For each of the following cases, we make a new partition P' by removing a misordering and show that $\Delta V = V(P') - V(P) \geq 0$.

i. $r_m \geq g_b$

$$\begin{array}{rcl} G'_a & = & G_a - \{m\} \\ G'_b & = & G_b \cup \{m\} \\ \Delta V & \geq & u(r_m, g_b) - u(r_m, g_a) \geq 0 \end{array}$$

ii. $r_n \leq g_a$

$$\begin{array}{rcl} G'_a & = & G_a \cup \{n\} \\ G'_b & = & G_b - \{n\} \\ \Delta V & \geq & u(r_n, g_a) - u(r_n, g_b) \geq 0 \end{array}$$

iii. $g_a < r_n < r_m < g_b$

$$\begin{array}{rcl} G'_a & = & (G_a \cup \{n\}) - \{m\} \\ G'_b & = & (G_b \cup \{m\}) - \{n\} \\ \Delta U(G_a) & \geq & -u(r_m, g_a) + u(r_n, g_a) \geq 0 \\ \Delta U(G_b) & \geq & -u(r_n, g_b) + u(r_m, g_b) \geq 0 \\ \Delta V & \geq & \Delta U(G_a) + \Delta U(G_b) \geq 0 \end{array}$$

For each case, $V(P') - V(P) = \Delta V \geq 0$, where $P' = (P - \{G_a, G_b\}) \cup \{G'_a, G'_b\}$. However, since P is an optimal partition, $V(P')$ cannot exceed $V(P)$. Thus $V(P')$ is equal to $V(P)$. By repeating this until there are no G_a and G_b such that $r_m > r_n$, where $r_m = \max_{k \in G_a} r_k$ and $r_n = \min_{k \in G_b} r_k$, we get an ordered optimal receiver partition. □

B Proof of Theorem 2

Without loss of generality, assume $G = \{1, 2, \ldots, m\}$. Following our convention, we assume $r_1 \leq r_2 \leq \cdots \leq r_m$.

Consider any receiver i, where $i \in G$. We notice that with Receiver Utility Property $u(r_i, g)$ is nondecreasing with respect to g in the interval $[0, r_1]$ and nonincreasing in the interval $[r_m, \infty)$. Since $U(G, g) = \sum_{i \in G} u(r_i, g)$, we know $U(G, g)$ is nondecreasing in the interval $[0, r_1]$ and nonincreasing in the interval $[r_m, \infty)$. Therefore, to find the maximum of $U(G, g)$, we only need to consider the interval $[r_1, r_m]$.

Next, again consider any receiver i, where $i \in G$. Consider any interval $[r_j, r_{j+1}]$ for $1 \leq j < m$. We know that in this interval $u(r_i, g)$ is a convex function of g. Since $U(G, g)$ is the sum of $u(r_i, g)$, for $i \in G$, it is also convex in each of the intervals. According to the property of convex functions, we know that $U(G, g)$ will have a maximum value at $g = r_j$ or $g = r_{j+1}$ when g is in the interval $[r_j, r_{j+1}]$. □

References

[1] M. H. Ammar and L. R. Wu. Improving the throughput of point-to-multipoint ARQ protocols through destination set splitting. In *Proceedings of IEEE INFOCOM '92*. IEEE Communications Society, 1992.

[2] S. Bhattacharyya, D. Towsley, and J. Kurose. The loss path multiplicity problem in multicast congestion control. In *Proceedings of IEEE INFOCOM '99*, New York, New York, U.S.A., Mar. 1999.

[3] J. Bolot and T. Turletti. A rate control mechanism for packet video in the Internet. In *Proceedings of IEEE INFOCOM '94*, Toronto, Ontario, Canada, June 1994.

[4] S. Y. Cheung and M. H. Ammar. Using destination set grouping to improve performance of window-controlled multipoint connections. *Computer Communications*, (19):723–736, 1996.

[5] S. Y. Cheung, M. H. Ammar, and X. Li. On the use of destination set grouping to improve fairness in multicast video distribution. In *Proceedings of IEEE INFOCOM '96*, San Francisco, California, U.S.A., Mar. 1996.

[6] D. DeLucia and K. Obraczka. Multicast feedback suppression using representatives. In *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, Apr. 1997.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[8] M. Handley and S. Floyd. Strawman specification for TCP friendly (reliable) multicast congestion control (TFMCC), working draft, Dec. 1998.

[9] T. Jiang, M. H. Ammar, and E. W. Zegura. Inter-receiver fairness: A novel performance measure for multicast ABR sessions. In *Proceedings of ACM SIGMETRICS '98*, Madison, Wisconsin, U.S.A., June 1998.

[10] T. Jiang, M. H. Ammar, and E. W. Zegura. On the use of destination set grouping to improve inter-receiver fairness for multicast ABR sessions. In *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[11] X. Li, M. H. Ammar, and S. Paul. Video multicast over the internet. *IEEE Network Magazine*, 13(2):46–60, Mar. 1999.

[12] X. Li, S. Paul, and M. H. Ammar. Multi-session rate control for layered video multicast. In *Proceedings of Multimedia Computing and Networking '99*, San Jose, California, U.S.A., Jan. 1999.

[13] X. Li, S. Paul, P. Pancha, and M. H. Ammar. Layered multicast with retransmission (LVMR): Evaluation of error recovery schemes. In *Proceedings of NOSSDAV '97*, St. Louis, Missouri, U.S.A., May 1997.

[14] S. McCanne. *Scalable Compressoin and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, Dec. 1996.

[15] S. McCanne, V. Jacobson, and M. Vetterli. Receiver driven layered multicast. In *Proceedings of ACM SIGCOMM '96*, Stanford, California, U.S.A., Aug. 1996.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM '98*, Vancouver, British Columbia, Canada, Aug. 1998.

[17] W. Ren, K. Siu, and H. Suzuki. On the performance of congestion control algorithm for multicast ABR service in ATM. In *Proceedings of IEEE ATM '96 Workshop*, San Francisco, California, U.S.A., Aug. 1996.

[18] D. Rubenstein, J. Kurose, and D. Towsley. The impact of multicast layering on network fairness. In *Proceedings of ACM SIGCOMM '99*, Cambridge, Massachusetts, U.S.A., Aug. 1999.

[19] N. Shacham. Multipoint communication by hierarchically encoded data. In *Proceedings of IEEE INFOCOM '92*, Florence, Italy, May 1992.

[20] T. Turletti, S. F. Parisis, and J. Bolot. Experiments with a layered transmission scheme over the Internet. In *Proceedings of IEEE INFOCOM '98*, San Francisco, California, U.S.A., Mar. 1998.

[21] H. Tzeng and K. Siu. On max-min fairness congestion control for multicast ABR service in ATM. *JSAC*, 15(3), Apr. 1997.

[22] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of IEEE INFOCOM '99*, New York, New York, U.S.A., Mar. 1999.

[23] M. Vishwanath and P. Chou. An efficient algorithm for hierarchical compression of video. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, U.S.A., Nov. 1994.

[24] H. A. Wang and M. Schwartz. Achieving bounded fairness for multicast traffic and TCP traffic in the Internet. In *Proceedings of ACM SIGCOMM '98*, Vancouver, British Columbia, Canada, Aug. 1998.