Efficient Multicast Flow Control using Multiple Multicast Groups *

Supratik Bhattacharyya
James F. Kurose
Don Towsley
Department of Computer Science
University of Massachusetts
Amherst MA 01003 USA

Ramesh Nagarajan
Bell Laboratories
Lucent Technologies
Holmdel NJ 07733 USA

CMPSCI Technical Report TR 97-15

March 10, 1997

Abstract

Controlling the rate of multicast data transfer to a large number of receivers is difficult, due to the heterogeneity among the end-systems' capabilities and their available network bandwidth. If the data transfer rate is too high, certain receivers will lose data, and retransmissions will be required. If the data transfer rate is too slow, an inordinate amount of time will be required to transfer the data. In this paper, we examine an approach towards multicast flow control in which the sender uses multiple multicast groups to transmit data at different rates to different sub-groups of receivers. Transmission rates for the multicast groups are chosen so as to minimize the average time needed to transfer data to all receivers. We present simple algorithms for determining the transmission rate associated with each multicast channel. Analysis and simulation are used to show that our techniques for rate selection and rate adjustment perform well for large and diverse receiver groups and make efficient use of network bandwidth. Moreover, we find that only a small number of multicast groups are needed to reap most of the performance benefits possible.

*This work was supported by the National Science Foundation under grant NCR-95-08274. The authors can be contacted at : [bhattach,kurose,towsley@cs.umass.edu and rameshn@lucent.com.

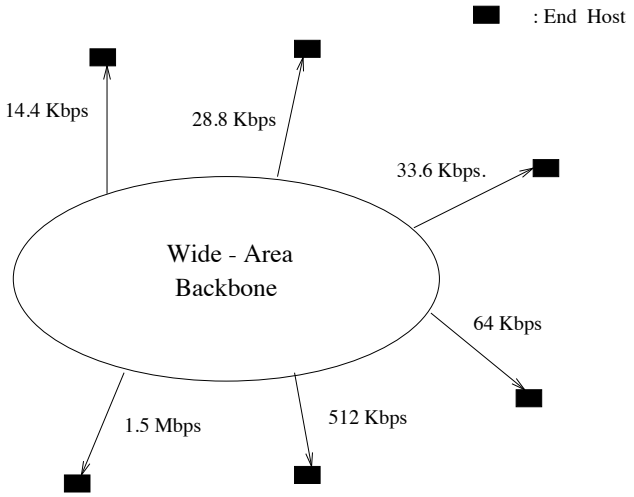

Figure 1: Network with heterogeneous hosts

1 Introduction

Multicasting has enabled a wide range of applications in wide-area networks. Applications requiring reliable dissemination of large volumes of data form an important subclass of these multicast applications. Examples include multicast file transfer, Usenet news distribution, web cache preloading and distribution of databases for interactive virtual environments.

The heterogeneity of computer systems and networks, however, significantly complicates the problem of efficient multicast of bulk data. This is particularly the case for flow control, where the sender is confronted with a multitude of receivers with widely varying rates at which they can receive data. These rates may be limited either by the processing capabilities of the end systems or by bandwidth limitations on the paths between the sender and the receivers. The goal of multicast flow control is deceptively simple – the sender should ideally choose its rate(s) to minimize the time needed to transfer data to the receivers, while at the same time minimizing the amount of bandwidth used. Realizing this goal, however is difficult. If the sender chooses a rate that is too high, certain receivers will lose data, and retransmissions will be required. If the data transfer rate is too low, an inordinate amount of time will be required to transfer the data.

In this paper, we examine an approach towards multicast *flow control* in which the sender uses multiple multicast groups to transmit data. In our approach, the sender transmits data to these multicast groups at different rates, and each receiver joins (receives data from) one or more of these groups, subject to its receive rate limitation. We focus on the case where the flow-controlled sender has a fixed amount of data to send (such as for multicast ftp) to the receivers. We present algorithms for choosing the rate at which the sender transmits data on each multicast channel, and for determining which portions of the data should be sent over which channel, and when. We consider two types of rate assignment policies. In one case, the sender's transmission rate on each channel is fixed. Here, we demonstrate a policy that is shown to be optimal in the sense of minimizing the average time needed to transfer the data to all receivers.

In the second case, the sender can change its transmission rates when some of the receivers (those with higher receive rates) finish receiving all the data. Here, we propose simple heuristics for determining the

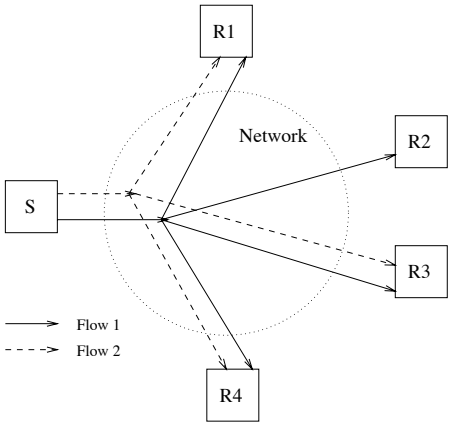

Figure 2: System Model

transmission rates that are shown to be close to optimal, even for large groups of receivers with receive capacities that differ by orders of magnitude. We show that previously proposed algorithms (such as those that choose transmission rates with a heavy bias towards either the slowest or the fastest receivers in a group) perform quite poorly. A significant observation from a practical standpoint is that our flow control scheme performs well even when restricted to use a small number of multicast groups.

The remainder of this paper is organized as follows. Section 2 defines the flow control problem and the system model in detail. Section 3 presents an approach for rate assignment and data transmission with an unlimited number of multicast channels. Section 4 presents algorithms for the case that there are a limited number of multicast channels. Results from the simulation study are described in Section 5. Related work is reviewed in Section 6. Section 7 concludes the paper.

2 Problem Setting

The problem of transmission rate control for reliable multicast communication is complicated by the scale of multicast applications and the heterogeneity of existing networks. Applications like DIS, news dissemination, etc. are expected to have hundreds or even thousands of members in each multicast group. Any transmission rate control scheme must thus scale well to such large groups. The problem would be easier if all of the receivers were identical in all respects. However, today's communication infrastructure is heterogeneous. End-systems exhibit diversity in their processing speeds, buffering capacities and background loads. The network bandwidth on the paths from a sender to receivers in the same multicast group may also show high variability. It is not uncommon to see bandwidth variations from 14.4Kbps for modem connections to a few megabits per second for high-speed $T1$ links (Figure 1). End-system or network constraints (or both) can limit the rate at which a receiver can receive data.

When transmitting data to a group of heterogeneous receivers at a *single* rate, one can choose this rate in any of several ways. However, every choice leads to a tradeoff. For example, when the sender's rate is set to the rate of the slowest receiver, the processing capacity of the faster receivers is underutilized. Conversely, if the sender matches the rate of the fastest receiver, the slower receivers lose data due to overflow and the

resulting retransmissions waste both bandwidth and sender processing capacity.

In this paper we consider the problem of transferring a *fixed* (finite) amount of data from a sender to a number of heterogeneous receivers, each of which must eventually receive all of the data. Multicast ftp is an application with such requirements. We contrast this with the case of long-lived (i.e., infinite-length) sessions, where the sender is necessarily constrained to match the rate of the slowest receiver if all receivers must eventually receive all data. With a fixed amount of data to be transferred, the sender has considerably more flexibility. For example, it could first transmit at a high rate (allowing "fast" receivers to complete quickly) and then retransmit at a low rate (allowing the "slower" receivers to receive the data they had previously missed).

Clearly, in order to quantitatively evaluate the performance of various flow control policies, we will need specific performance criteria. Informally, we define the *completion time* of a receiver as the time taken for the receiver to receive all the data. We will be primarily interested in the *average completion time* of all receivers. As discussed in section 5, we will be considering other performance measures as well. Let us characterize each receiver by a "rate" that represents how fast it can receive data. Given these definitions, the specific problem that we address in this paper is:

How to multicast a data stream of finite length to a group of receivers with heterogeneous "rates" so as to minimize the average completion time.

Recognizing the limitation of a single transmission rate for heterogeneous receivers, we adopt a *multirate* transmission approach in which the sender transmits data at more than one rate. Data is multicast in a *nested* fashion - the sender partitions a data stream into a number of parts and transmits each over a separate multicast group at a certain rate. The slowest receiver(s) will receive from only one group. Faster receivers will be able to receive data at an overall higher rate by receiving data *concurrently* from *several* multicast groups. Our goal is to determine the best way of assigning rates to the channels and partitioning data segments among them, with the constraint that all receivers must eventually receiver all the data.

2.1 System Model

Figure 2 shows a simple multicasting model. Although the issue of error recovery is quite central to reliable multicasting, we are interested in the problem of rate control, not error recovery. Hence we assume that the underlying network for data delivery is *free from loss and error*. We also assume that data can be delivered out of order to a receiver. A justification for this is provided in [9]. We note that this does *not* imply that data will not be retransmitted since (as noted above) a slower receiver may not have been able to receive a previously transmitted data segment due its rate limitations.

Data transmission across the network takes place over *channels*, each of which can be thought of as a multicast group. The *channel rate* is the maximum amount of data that a sender will transmit over the channel per unit time. A sender can transmit data on multiple channels concurrently and a receiver can receive from multiple channels concurrently. For example, in Figure 2, receivers $R1$, $R2$, $R3$ and $R4$ receive data from sender S via channel 1, while $R1$, $R3$ and $R4$ also receive data via channel 2. We will see shortly that a receiver should subscribes to a channel only if it can receive all the data that is transmitted on the channel.

Data is modeled as a continuous, infinitely divisible fluid stream. A *data segment* refers to a finite portion of a data stream and is identified uniquely by a *(head, length)* pair. The head is the position of the segment with respect to the beginning of the stream and the length is the size of the data segment. A data

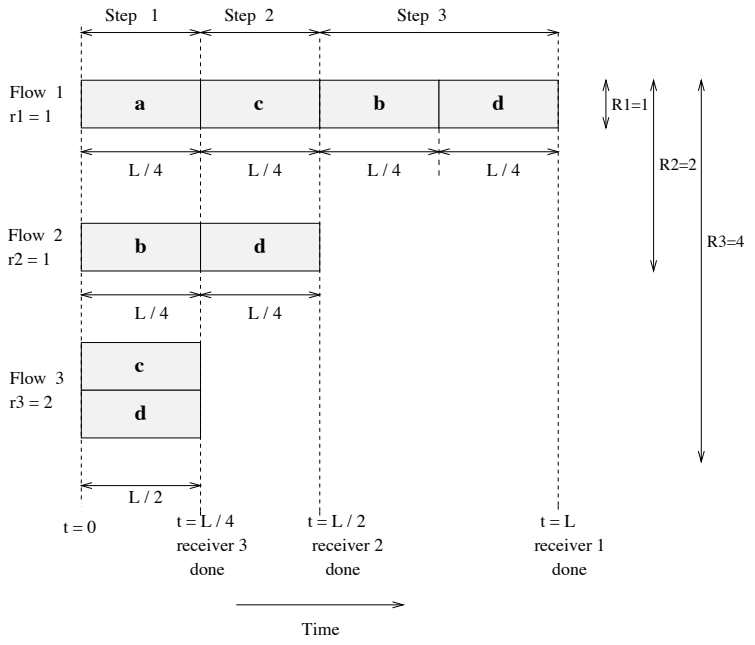

Figure 3: Nested Transmission Example

stream of length L can itself be considered to be a segment, represented by $(0, L)$.

Each receiver is characterized by a maximum rate at which it can receive data. We assume this rate is determined by a single, fixed, time-invariant bottleneck and that the sender can determine the receiver's rate (e.g., either by directly querying the receiver or by estimating this rate via measurement) before data transfer begins. We consider a set of N receivers with rates $R_1, R_2, \ldots, R_N$. The *ideal completion time* of a receiver is defined as the amount of time the receiver needs to receive the entire data stream when receiving at its maximum rate. With a data stream of length L, the ideal completion time of receiver n is given by $I_n = L/R_n$.

A *rate assignment policy*, r, is a policy for associating transmission rates with channels. The subset of channels that each receiver subscribes is determined by a *subscription policy* $\pi(r)$. Let there be K channels with rates $(r_1, r_2, \ldots r_K)$. Then the set of channels that receiver n subscribes to is its *subscription set* $S_n^{\pi(r)}$, where $S_n^{\pi(r)} \subseteq \{1, 2, \ldots, K\}$. The maximum rate at which receiver n receives data is thus $\sum_{k \in S_n^{\pi(r)}} r_k \leq R_n$. The *actual completion time* of a receiver, under the assumption that it can receive all the data sent on all the subscribed channels, is defined as the actual time it takes to receive a data stream. For receiver n and for a stream length L, this is given by $T_n = L / \sum_{k \in S_n^{\pi(r)}} r_k$.

As an illustration of our nested multicasting scheme, consider a data stream of length L that is transmitted over 3 channels to 3 receivers with maximum rates $R_1 = 1, R_2 = 2$ and $R_3 = 4$ (Figure 3). The channel rates are set to $r_1 = 1, r_2 = 1$ and $r_3 = 2$. Receiver 1 subscribes to channel 1, receiver 2 to channels 1 and 2 and receiver 3 to all the channels.

In this example, the sender partitions the data stream into 4 equal-sized segments a, b, c and d. First a is transmitted over channel 1, b over channel 2 and c and d over channel 3. These concurrent transmissions finish at the same time $L/4$. At this time, only R_3 has received the entire stream. We then transmit segment

c over channel 1 and d over channel 2 concurrently. When these transmissions complete (at time $L/2$), receiver 2 has received all four segments, but receiver 1 still has not received segments b and d. These are thus then transmitted over channel 1. At time L, receiver 1 completes. In this example, by using three channels for three receivers, each receiver completes in its *ideal* completion time.

The features of our model can now be summarized as follows :

- The network is loss-free and error-free.
- Data is a continuous stream delivered over channels,
- Data can be delivered out of order to a receiver.
- Each channel k is assigned a fixed rate r_k.
- Each receiver n is characterized by a maximum processing rate R_n and this rate is known to the sender.

As noted earlier, in the algorithms we will consider, a receiver will only subscribe to a channel if it can receive *all* of the data that is transmitted on that channel. We close this section by seeing informally why this should be so.

Consider a receiver with a maximum processing rate of 4 and two channels with rates $r_1 = 3$ and $r_2 = 2$. If a receiver only subscribes to a channel if it can receive all of the data, then the receiver can subscribe to only one channel, thereby receiving at a maximum rate of 3. Suppose now that a receiver were to be allowed to receive a portion of the data transmitted on a channel. If the receiver were allowed to discard half of the data that was transmitted on channel 2, then it would be able to receive data at a cumulative rate of 4 – its maximum rate – by subscribing to both the channels. This would seem ideal.

However, the second approach leads to a waste of network bandwidth for *multicast* communication, particularly when there are a large number of receivers. Consider a channel that has N subscribing receivers. If each receiver independently drops a fraction p of the data on that channel (i.e., because the receiver's rate is such that it can only receive a fraction $1 - p$ of the data), the probability that a given data segment is correctly received by all N receivers is $(1-p)^N$. Hence the probability that the sender must retransmit that segment is $(1-(1-p)^N)$, or almost .99 with $p = .2$ and $N = 20$. As N grows large, the probability that one or more of the receivers will require a retransmission (due to a rate-limited packet drop at a receiver) approaches 1, and the expected number of times a segment must be transmitted consequently approaches infinity. Given these consideration, we conclude that a receiver should only join a multicast group if it is capable of receiving all of the data transmitted on that channel.

3 Multirate Transmission With Unlimited Channels

When the number of channels available for data transfer is unlimited, it is possible to attain the ideal completion time for every receiver by using as many channels as there are receivers. We describe a scheme to accomplish this by generalizing the idea in Figure 3.

Before proceeding further, we provide some definitions to set the context for the subsequent discussion. A *data transmission schedule*, Δ, determines the way that data is divided among the channels and the order in which the segments are transmitted. A schedule is defined to be *work-conserving* if every channel always transmits data at the channel rate as long as it is in use. A schedule is defined to be *duplicate-free* if it never delivers the same data segment to any receiver more than once. In order that every receiver receives data at a rate that is equal to the sum of the rates of all the channels that it subscribes to, the transmission schedule has to be work-conserving and duplicate-free. A transmission policy (Δ, r) has two components :

- a transmission schedule (Δ).
- a channel rate assignment policy (r).

A *multirate transmission scheme* $M = \{\Delta, r, \pi(r)\}$, is combination of a transmission policy $\{\Delta, r\}$ and a subscription policy $\pi(r)$.

Now consider a set of K receivers, $K \geq 2$ with rates $R_1 < R_2 < \cdots < R_K$. To match the rates of all K receivers, we use K channels. Channel rates $(r_1, r_2, \ldots, r_K)$ are chosen as :

$$\begin{aligned} r_1 &= R_1, \\ r_n &= R_n - R_{n-1}, \quad 1 < n \leq K. \end{aligned}$$

We define a subscription policy π_{ideal} such that the subscription set of receiver k is given by

$$S_k^{\pi_{ideal}} = \{1, 2, \cdots k\}, \quad 1 \leq k \leq K$$

. The transmission schedule A_{ideal} can be described as follows. Data segments are transmitted in a number of steps such that in step i, channels 1 through $K - i + 1$ are used for transmission. Let $\mathcal{B}_{k,i}$ be the set of segments that are transmitted on channel k in step i. Let $\mathcal{Q}_{k,i}$ be the set of segments that are transmitted on channel i in steps 1 through i. Then

$$\mathcal{Q}_{k,i} = \bigcup_{j=1}^{i} \mathcal{B}_{k,j}$$

Let us define a class of functions $\{g_{m,n}\}_{m,n=1}^{K}; g_{m,n} : R_+^2 \rightarrow R_+^2$, where R_+^2 is the set of positive real numbers. For every $a = (h, l)$, $g_{m,n}(a) = (h', l')$ where

$$\begin{aligned} h' &= h + (\sum_{j=1}^{m-1} r_j / \sum_{j=1}^{n} r_j) * l, \\ l' &= (r_m / \sum_{j=1}^{n} r_j) * l. \end{aligned}$$

This means that whenever a data segment (h, l) is partitioned among channels 1 through n, then the length of the sub-segment assigned to channel m is proportional to the channel rate r_m.

In step 1, the entire data stream $(0, L)$ is partitioned among and transmitted concurrently over K channels, i.e., for $k = 1, \cdots K$,

$$\mathcal{B}_{k,1} = \{((\sum_{m=1}^{k-1} r_m / \sum_{m=1}^{K} r_m) * L, (r_k / \sum_{m=1}^{K} r_m) * L)\}$$

In step i, $(i > 1)$, all of the segments that were transmitted over channels $K - i + 2$ in step 1 through $i - 1$, are partitioned among and transmitted concurrently over channels 1 through $K - i + 1$. That is, for $i = 1, 2, \ldots K$ and $k = 1, \cdots K - i + 1$,

$$\mathcal{B}_{k,i} = \{g_{k,K-i+1}(a), | a \in \mathcal{Q}_{K-i+2,i-1}\}$$

A_{ideal} has the following important properties :

Property P1 *All K receivers receive the complete data stream at the end of step K.*

Property P2 *A_{ideal} is duplicate-free.*

Property P3 *A_{ideal} is work-conserving.*

The proofs are provided in the appendix.

Since A_{ideal} is duplicate-free and work conserving, receiver k receives data at the cumulative rate of all the channels in $S_k^{\pi_{ideal}}$, i.e., at rate $\sum_{i=1}^{k} r_i = R_k$. Therefore, every receiver finishes in its ideal completion time.

A_{ideal} is also efficient in terms of bandwidth usage. Let us define the *transmission volume* as the total volume of data transmitted by a sender in the course of delivering a data stream to all receivers. Then A_{ideal} is optimal in the sense of minimizing the transmission volume :

Property P4 *Of all multirate transmission schemes that attain the ideal completion time for every receiver, A_{ideal} is optimal in terms of transmission volume.*

The proof of $P4$ is also provided in the appendix.

4 Multirate Transmission with Limited Channels

In the previous section, we have derived an ideal multirate transmission scheme by using as many channels as receivers. However, for real applications with a large number of receivers and high variability in rates, it is unrealistic to assume enough channels to match the rates of all receivers. Hence, in this section, we explore flow control approaches when the number of channels K is smaller than the number of receivers N. We examine two types of rate assignment policies - *static* assignment, when the rates are assigned once and for all prior to the start of data transmission, and *dynamic* assignment, when the sender readjusts transmission rates whenever some of the faster receivers finish receiving data.

Given N receivers and K channels, $K < N$, we would like to find a transmission scheme $M = \{A^*, r^*, \pi^*(r)\}$, that minimizes the average actual completion time of all the receivers. With a duplicate-free and work-conserving schedule, the actual completion time of receiver n is $T_n = L/\sum_{k \in S_n^{\pi^*(r)}} r_k$. So the problem is to choose a set of channel rates $r_1, r_2 \cdots r_K$ so as to

minimize $\quad \sum_{n=1}^{N} T_n$

s.t. $\quad \sum_{k \in S_n^{\pi^*(r)}} r_k \leq R_n, \quad j = 1, \ldots, N.$

However it is difficult to ensure that in general there is a work-conserving and duplicate-free schedule for a given set of of channel rates and a given subscription policy. Hence we focus on a class of nested subscription policies Π in which, for every policy $\pi \in \Pi$, a receiver must subscribe to channels $1, 2, \cdots (k-1)$ in order to subscribe to channel k, $(k = 2, \cdots K)$, i.e.

$$S_n = \{1, \cdots k_n\}, \quad 1 \leq k_n \leq K,\ 1 \leq n \leq N.$$

It follows that with K channels, the sender offers K aggregate rates, $F_1, F_2, \cdots, F_K$, to receivers, where

$$F_k = \sum_{i=1}^{k} r_i, \quad k = 1, \ldots K.$$

Since we are interested in every receiver receiving data at the highest possible rate, we choose a specific policy $\pi_c \in \Pi$ for which

$$(1) \quad S_n = \{1, \ldots, k_n\}, \quad 1 \leq k_n \leq K, \; S_1 \subseteq S_2 \subseteq \cdots S_N,$$
$$(2) \qquad \sum_{k=1}^{k_n} r_k \leq R_n < \sum_{k=1}^{k_n+1} r_k, \quad n = 1, \ldots N.$$

This states that every receiver subscribes to as many channels as possible without the cumulative rate of the subscribed channels exceeding its own maximum capacity.

As we shall show later, each policy $\pi \in \Pi$ has a work-conserving and duplicate-free schedule associated with it, irrespective of the channel rates. So the problem now is to choose channel rates so as to optimize performance. In section 4.1, we present a static rate assignment policy that is optimal for π_c while in section 4.2, we examine dynamic channel rate adjustment heuristics for π_c.

4.1 Static Channel Rate Assignment

Under a static rate assignment, channel rates are assigned only once, prior to the start of data transmission, and under π_c, each receiver subscribes to the same set of channels for the entire duration of data transmission. Lemma 1 states an important property, viz., the existence of a conflict-free and work conserving transmission schedule for any policy $\pi \in \Pi$, and hence for π_c :

Lemma 1 *For any subscription policy $\pi \in \mathcal{B}$ and any static channel rate assignment policy, there exists a conflict-free and work-conserving transmission schedule.*

Schedule A_{ideal} (section 3) is one such schedule. We have already shown that A_{ideal} is conflict-free and work-conserving for π_{ideal} and for any set of channel rates $(r_1, r_2, \ldots, r_K)$. It is easy to infer that properties $P1, P2$ and $P3$ of A_{ideal} are valid for any $\pi \in \mathcal{B}$.

It follows from Lemma 1 that every receiver n always receives data at rate $\sum_{k=1}^{k_n} r_k$. Hence the actual completion time of receiver n is $L/\sum_{k=1}^{k_n} r_k$. Our goal is to choose $r_1, r_2, \cdots r_K$ so as to

$(C1)$ Minimize $\quad \sum_{n=1}^{N} L/\sum_{k=1}^{k_n} r_k$

s.t. $\quad 1 \leq k_1 \leq \cdots \leq k_N,$
$\sum_{k=1}^{k_n} r_k \leq R_n, \quad j = 1, \ldots, N.$

The following result expedites the search for an optimal solution by greatly reducing the size of the solution search space :

Lemma 2 *For every optimal solution to C1, $F_k \in \{R_1, \ldots, R_N\}$.*

See appendix for proof.

Since the actual completion time of every receiver n is given by L/F_{k_n}, our aim is to choose $F_1, F_2, \cdots F_K$ so as to minimize the total completion time $\sum_{n=1}^{N} L/F_{k_n}$. Thus we can restate $C1$ as :

$(C2)$ Minimize $\quad \sum_{n=1}^{N} \sum_{k=1}^{K} 1(F_k \leq R_n < F_{k+1})/F_k$

s.t. $\quad F_k \in \{R_1, \ldots, R_N\}, \quad k = 1, \ldots, K$

	Single rate (14.4 Kbps)	Nested Scheme with 3 channels	Simulcast with 3 channels
Average completion time	5555.6s	2165.67s	2165.67s
Transmission volume	10MB	22.7MB	30MB

Table 1: Comparison of transmission schemes for a 10Mb software.

where $F_{K+1} = \infty$ and $1(P)$ takes value 1 if the predicate P is true and value 0 otherwise.

An exhaustive search for an optimal solution to $C2$ involves $\binom{N}{K}$ possibilities. However, we observe the following structure for the optimal solution. If $T(n,k)$ is the optimal solution for the n slowest receivers ($n = 1, 2, \ldots, N$) and k channels, then $T(n,k)$ satisfies the following recursions :

$$\begin{aligned} T(n,k+1) &= \min_{1 \le m \le n} \{m/R_{n-m+1} + T(n-m,k)\} \\ T(n+1,k) &= \min\{1/R_{n+1} + T(n,k-1), T(n,k) + 1/F_k^*(n,k)\} \end{aligned}$$

where we are interested in finding $T(N,K)$ and the rate policy $F_1^*(N,K), F_2^*(N,K), \ldots, F_K^*(N,K)$ that go along with it. Dynamic programming enables us to solve this problem in time $O(N^3K)$ with a space complexity of $O(NK)$.

We have identified an optimal static rate assignment policy. Let us now briefly examine its performance by comparing it with to other possible flow-controlled transmission schemes.In one case, data is transmitted at the rate of the slowest receiver. In the second scheme, receivers are split into mutually exclusive subgroups and the sender transmits data to each group independently over a separate channel. We refer to this as *simulcast*. Suppose that the goal is to download a 10MB file from a server to 13 clients capable of receiving at the following rates (in Kbps) : $14.4, 28.8, 28.8, 28.8, 33.6, 33.6, 64, 64, 128, 128, 256, 512$ and 1000. The client rates are representative of the rates that we can expect to see on the Internet today.

The performance of the above three schemes are compared with respect to two performance metrics - the average completion time and the transmission volume (Table 1). We see that a large (about 60%) reduction in completion time is obtained by using our nested multirate scheme instead of transmitting data at the rate of the slowest client. If the optimal channel rates are r_1, r_2 and r_3, then simulcast attains the same average completion time with channel rates F_1, F_2 and F_3 given by $F_i = \sum_{j=1}^{i} r_i, \quad i = 1, 2, 3$. However simulcast is inefficient in terms of bandwidth utilization. Since simulcast multicasts data independently to the 3 sub-groups, the total volume of data transmitted is three times the size of the software, i.e., 30MB. The nested scheme transmits only 22.7MB, representing a significant performance gain over simulcast in terms of bandwidth use.

4.2 Dynamic Channel Rate Assignment

Under a dynamic rate assignment policy, the sender is allowed to readjust the channel rates while transmitting data. When assigning channel rates based on the rates of receivers, we must keep in mind that the set of unfinished receivers keeps changing. This is because different receivers receive data at different rates and complete at different times. Though channel rate readjustments can be done at any time, the benefits are likely to be the maximum if adjustments are done whenever the set of unfinished receivers changes. Hence we focus on a subclass of dynamic policies, Δ, that readjust channel rates whenever one of more receiver(s) finish receiving data and drop out.

Consider N receivers and K channels where $K < N$. An initial set of channel rates is chosen based on the rates of all N receivers. Subsequently, when one or more receivers finish(es), the channel rates are adjusted based on the rates of the *unfinished* receivers. This is continued until there are only K unfinished receivers. At this point, the channel rates can be optimally set to match the rate of each of these K receivers (as in section 3).

A *transmission stage* is defined as the interval between successive rate assignments. Since the subscription set of a receiver depends on the channel rates, every unfinished receiver must recompute its subscription set at the beginning of each stage.

We now consider the problem of finding an optimal policy $d^* \in \Delta$ for the nested subscription policy π_c. Let $S_n^t = \{1, 2, \cdots k_n^t\},\ 1 \leq k_n^t \leq K$, be the subscription set of receiver n in stage t. Let there be u_t unfinished receivers at the beginning of stage t. Let the channel rates for stage t be $r_1^t, r_2^t, \cdots r_K^t$. Then, for every stage t and for $1 \leq n \leq u_t$,

$$(1) \quad S_n^t = \{1, \ldots, k_n^t\}, \quad 1 \leq k_n^t \leq K, \quad S_1^t \subseteq S_2^t \subseteq \cdots S_{u_t}.$$
$$(2) \qquad \sum_{k=1}^{k_n^t} r_k^t \leq R_n < \sum_{k=1}^{k_n^t+1} r_k^t \quad j = 1, \ldots u_t.$$

The following lemma (proved in the appendix) states that every dynamic policy in Δ has a work-conserving and conflict-free schedule associated with it, for any subscription policy $\pi \in \mathcal{B}$:

Lemma 3 *For any subscription policy $\pi \in \mathcal{B}$ and any dynamic rate assignment policy $d \in \Delta$, there exists a conflict-free and work-conserving transmission schedule.*

Since the above result is true for $\pi_c \in \mathcal{B}$, the problem now is to find a set of optimal channel rates for every transmission stage. However, this problem is much harder to solve than the static rate assignment problem. There are a number of reasons - there are multiple sets of channel rates to be chosen, a receiver receives different portions of the data stream at different rates and the number of transmission stages is not known beforehand. This makes the search for the optimal solution computationally intractable. Hence, as a practical alternative, we study heuristics for assigning channel rates.

4.2.1 Rate Assignment Heuristics

We note that the difficulty in finding an optimal dynamic rate adjustment policy lies in having to consider the channel rate assignments for all stages together. To circumvent this difficulty, we consider heuristic-based approaches where rate assignment for each stage is independent of every other stage. This leads to simple algorithms, though we can show by counterexample that none are optimal :

1. *Equal Partitions* (EQ) : Let there be u^t unfinished receivers at the beginning of stage t and let $r_1^t, \ldots . r_K^t$ be the channel rates for the stage. We define $F_k^t = \sum_{j=1}^{k} r_k^t, \quad k = 1, \ldots, K$. Under EQ, $F_k (1 \leq k \leq K)$ is set at the $(100 * (k-1)/K)$th percentile of the range of rates of unfinished receivers $1, \ldots, u^t$. That is, we choose channel rates so that the aggregate rates available, F_k, are "smoothly" distributed between the maximum and minimum receiver rates. Let $W = R_{u^t} - R_1$ and let us define

$$w = W/K$$

Then

$$F_k^t = R_1 + w * (k-1), \quad k = 1, 2, \cdots K.$$

For example, with four channel and with receiver rates between $R_1 = 1$ and $R_{u^t} = 101, F_1 = 1,\ F_2 = 26,\ F_3 = 51$ and $F_4 = 76$.

Since EQ considers only the range of rates but not the rates of the individual receivers, it is a naive approach. Still, it is very simple to use and can be expected to perform well when receiver rates are evenly distributed between two known extreme values.

2. *Maximize Utilized Capacity* (MUC) : This approach attempts to maximize the sun of the aggregate rates at which all the unfinished receivers are receiving data. Let there be u^t receivers at the beginning of stage t and let the channel rates be $r_1, \ldots, r_K$. With a work-conserving and conflict-free schedule, every receiver n $(1 \leq n \leq u^t)$, receives data at a rate $\sum_{i=1}^{k_n^t} r_i^t$. Hence the goal is to choose $r_1^t, r_2^t, \ldots, r_K^t$ so as to

$$\begin{array}{ll} \text{maximize} & \sum_{n=1}^{u^t} \sum_{k=1}^{k_n^t} r_k^t \\ \\ \text{s.t.} & 1 \leq k_1 \leq \cdots \leq k_{u^t}, \\ & \sum_{k=1}^{k_n^t} r_k \leq R_n < \sum_{k=1}^{k_{n+1}^t} r_k^t, \quad n = 1, \ldots, u. \end{array}$$

Channel rates are chosen so as to match the rates of K of the u^t unfinished receivers. Defining $F_k^t = \sum_{j=1}^{k} r_k^t, \quad k = 1, \ldots, K$, our goal is to choose $F_1^t, F_2^t, \ldots, F_K^t$ so as to

$$\begin{array}{ll} (C3) \text{ maximize} & \sum_{n=1}^{u^t} \sum_{k=1}^{K} 1(F_k^t \leq R_n < F_{k+1}^t) * F_k^t \\ \\ \text{s.t.} & F_k^t \in \{R_1, \ldots, R_{u^t}\}, \quad k = 1, \ldots, K. \end{array}$$

where $F_{K+1}^t = \infty$.

It is clear from the formulation of $C3$ that an optimal solution for it has the same structure as one for $C2$. Hence we can obtain an optimal solution for $C3$ in polynomial time using dynamic programming.

5 Performance Evaluation

In this section, we use a simulator based on the system model described in section 2 to study the benefits of using multiple multicast groups for flow control and also to compare the performance of the rate assignment algorithms described earlier.

Our simulations have been performed with normalized values for receiver rates and data stream lengths. Since 14.4Kbps is a realistic estimate of the slowest receiver rate, we use this as the normalization factor. It is not uncommon to observe a variation of two orders (or even more) of magnitude in receiver rates. Hence we consider normalized receiver rates between 1 and 100. The length of the data stream is taken to be 5000 seconds worth of data at 14.4Kbps.

Receiver rates have been chosen from the following two distributions:

1. **UNI** : Receiver rates are integer values following a uniform random distribution between 1 and 100. Hence $R_i \in \{1, \ldots, 100\}$ and $P(R_i = k) = 0.01, \quad k = 1, 2, \ldots, 100$.

2. **SKEW** : Receiver rates are integer values following a skewed distribution between 1 and 100 such that

$$\begin{array}{rl} P(R_i = k) = 0.3/78, & k = 1, \ldots, 29, \\ = 0.4/11, & k = 30, \ldots, 40, \end{array}$$

$$= 0.3/78, \quad k = 41, \ldots, 69,$$
$$= 0.3/11, \quad k = 70, \ldots, 80,$$
$$= 0.3/78, \quad k = 81, \ldots, 100.$$

When evaluating the rate assignment algorithms, a number of independent replications are made so as to generate a confidence interval of 6% or less of the point value for the 95% confidence level of the performance metric.

In addition to the above algorithms, we simulate two other simple rate assignment approaches :

1. *Slowest Receivers First* (SRF) : Channel rates are assigned to match the rates of the K slowest unfinished receivers, i.e.

   $$F_i = R_i, \quad 1 \leq i \leq K.$$

   Note that the $(N - K)$ fastest receivers subscribe to all the channels and hence receive the entire data stream at the end of the first stage. This leaves only $(K - 1)$ unfinished receivers. Since the rates of these receivers are already matched by $F_1, \ldots F_{K-1}$, no readjustment is necessary. In that sense, SRF is a static policy.

2. *Fastest Receivers First* (FRF) : Channel rates are set to match the rates of K fastest *unfinished* receivers. If at the beginning of a transmission stage, receivers 1 through u are unfinished, then

   $$F_{K-i} = R_{u-i}, \quad i = 0, 1, 2, \cdots K - 1$$

   As we shall see later, these two extreme scenarios provide interesting insights into the behavior of rate assignment algorithms.

5.1 Comparison of Receiver Completion Times

In the first set of graphs (Figures 4 and 5), we compare the performance of various rate assignment algorithms in terms of the average completion time(T). The number of receivers(N) varies from 5 to 400 while the number of channels is fixed at $K = 4$. The average completion time is normalized by the completion time of the receiver whose rate is the expected value of the distribution in question (UNI for Figure 5 and SKEW for Figure 6). Thus the value on the y-axis can be interpreted as the number of times that the average completion time has increased due to receiver heterogeneity.

We observe that the performance of FRF and SRF scales very poorly with increasing values of N. For SRF, this is understandable because only the four slowest receivers receive at their maximum capacity and the capacities of all other receivers are highly underutilized. The result for FRF can be explained by observing that in any stage, all but the 4 fastest unfinished receivers are completely starved. The high utilization of the few fastest receivers does not compensate for the non-utilization of all other receivers.

The optimal static rate assignment policy (STATIC) is found to improve T by a factor of about 3 over FRF and SRF. Both MUC and EQ improve performance further by more than 15%. From the nature of the curves for STATIC, EQ and MUC, we can infer that all three exhibit very good scalability for large values of N. MUC is better than EQ, the difference between the two being 8% or less in the two graphs shown above. We expect the performance of EQ to degrade as the distribution of receiver rates becomes more and more skewed, since EQ assigns channel rates uniformly over a range of receiver rates. We have found that its performance is closer to MUC for UNI than for SKEW. A more careful comparison of the two has shown that the MUC can perform upto 12% better than EQ.

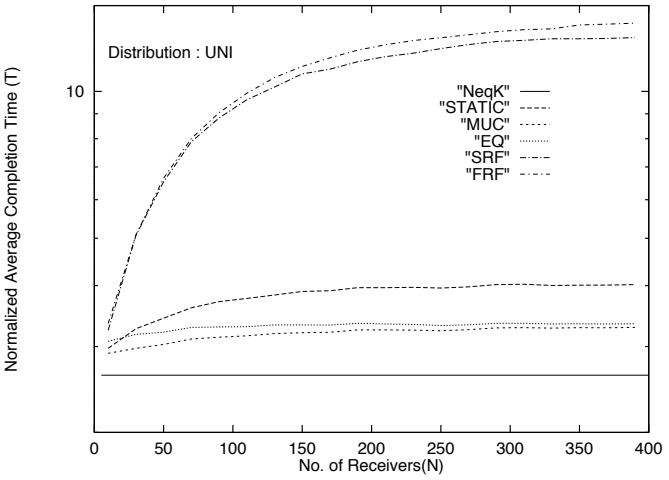

Figure 4: T vs. N for UNI

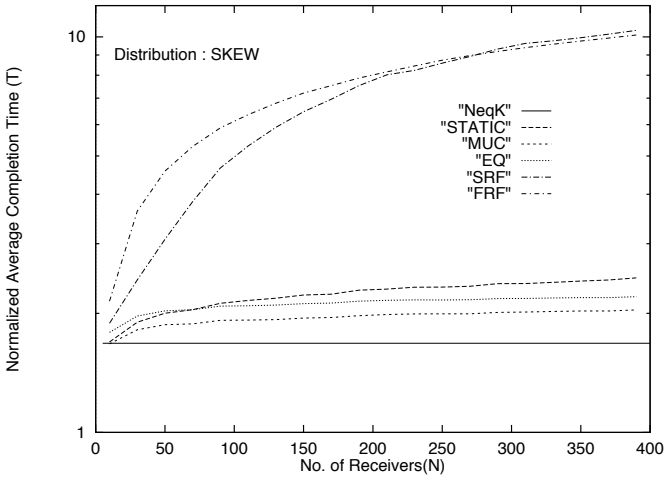

Figure 5: T vs. N for SKEW

The algorithm that performs best, viz. MUC, is about 25% worse than the lower bound NeqK shown in Figures 4 and 5. However, NeqK has been computed for the unlimited channel case ($N = K$), where each receiver completes in its the ideal completion time. This is impossible to achieve with four channels, hence NeqK represents an unattainable lower bound. We have computed the optimal average completion time by an exhaustive search for upto 10 receivers (further search is computationally intractable). For $N = 10$, the optimal value of T has been found to be within $15 - 20\%$ of the performance of MUC. Our intuition is that the optimal value of T is monotonically non-decreasing in N. Hence we expect MUC to perform no worse than 20% from the optimal even for higher values of N.

To summarize, we make the following observations :

- An "extreme" approach such as SRF or FRF is undesirable for channel rate assignment. A more "balanced" approach, exemplified by STATIC, EQ and MUC is good.
- Our nested multirate transmission scheme scales well to large receiver groups.
- Though a static channel rate assignment policy performs reasonably well, dynamic rate adjustment is beneficial.

5.2 Comparison of Bandwidth Utilization

An important concern for any rate assignment policy is how efficiently it uses network bandwidth. Hence we next consider the transmission volume V incurred by the same set of five algorithms. Figures 6 and 7 show the results for UNI and SKEW respectively. The value plotted on the y-axis is the transmission volume normalized by the data stream length, i.e., 5000. Thus the y-axis represents the transmission volume as a multiple of the stream length.

Our intuition that slower transmission rates result in lower transmission volume and vice-versa, is upheld by the results for SRF and FRF. In the case of SRF, all but the slowest three receivers are able to subscribe to all the channels, hence they complete after the entire stream has been transmitted only once. That leaves only three unfinished receivers. Thus, V is only about twice the stream length. On the other hand, FRF

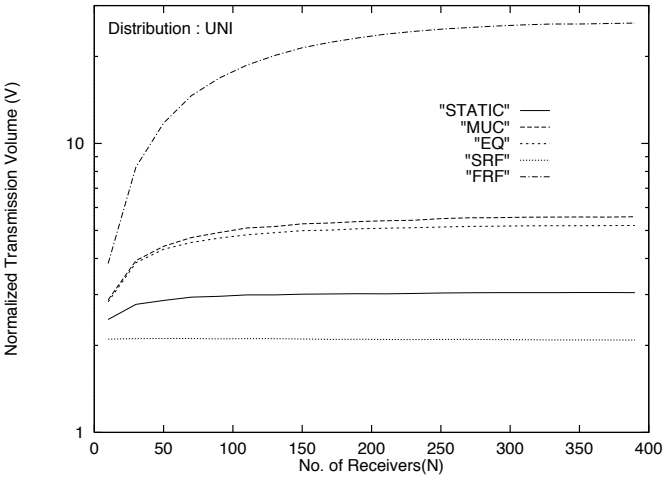

Figure 6: V vs. N for UNI

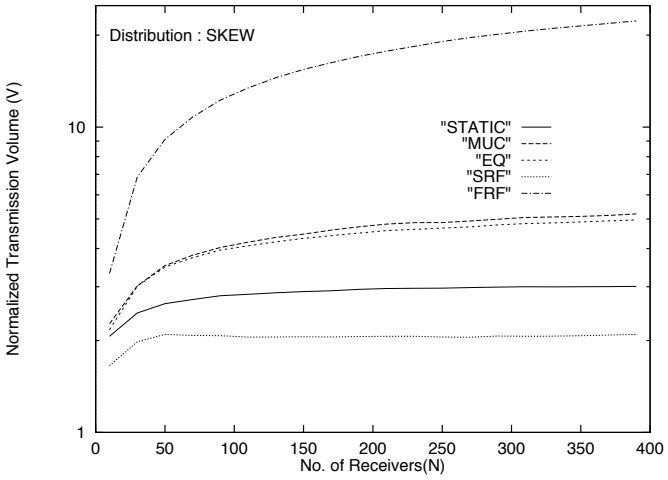

Figure 7: V vs. N for SKEW

starves all of the receivers except the K fastest unfinished ones, hence the data transmitted at each stage will be received by only a few receivers. This explains its poor performance.

Of greater interest is the performances of the STATIC, MUC and EQ algorithms that were found to achieve low average completion times. We observe that the performance scales very well with N in each case. The performance of MUC and EQ are comparable - about five times the size of the data stream, while STATIC requires only about three times the stream size. The transmission volume increases with the number of channels used, since the higher the number of channels used, the smaller the number of receivers receiving all the data segments sent out in any transmission stage. With dynamic policies EQ and MUC, all K channels are used until there are fewer than K unfinished receivers. However, for STATIC, some of the channels are released earlier. This explains why the transmission volume is higher for EQ and MUC than STATIC.

5.3 Required Number of Channels

One interesting question is how the efficiency of transmission is affected by the number of available channels. To address this, we have varied the number of channels from $K = 2$ to $K = 15$, keeping the number of receivers fixed at $N = 50$. The results are shown in Figures 8 and 9. The value of T on the y-axis is normalized as before.

The horizontal line NeqK shown in the graphs is an asymptotic lower bound for the performance of any rate assignment policy. It has been computed by setting $K = N = 50$, in which case it is possible to finish every receiver in its ideal completion time. Different policies approach the asymptotic bound at different rates - the rate of decrease is much faster for STATIC, MUC and EQ than for FRF and SRF. But more significantly, the curve is convex in each case. This implies that the performance improves rapidly for small values of K while the rate of improvement decreases at higher values of K. Hence most of benefits of using multiple channels can be realized with a small number of channels. MUC, for example, shows near optimal performance with only five channels. This result greatly enhances the viability of our nested transmission scheme from a practical standpoint. We have verified this result for other values of N though we have not presented the results here.

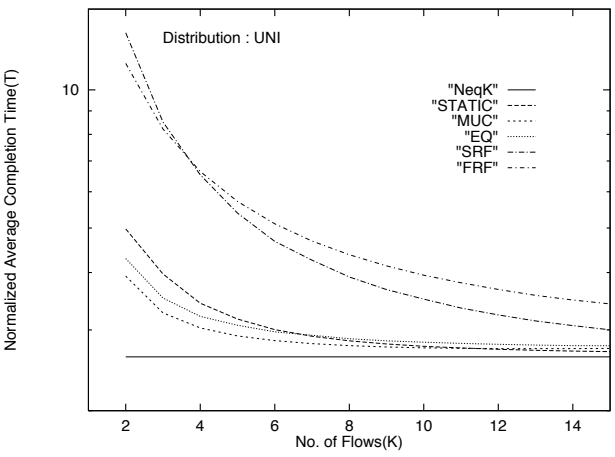

Figure 8: T vs. K for UNI

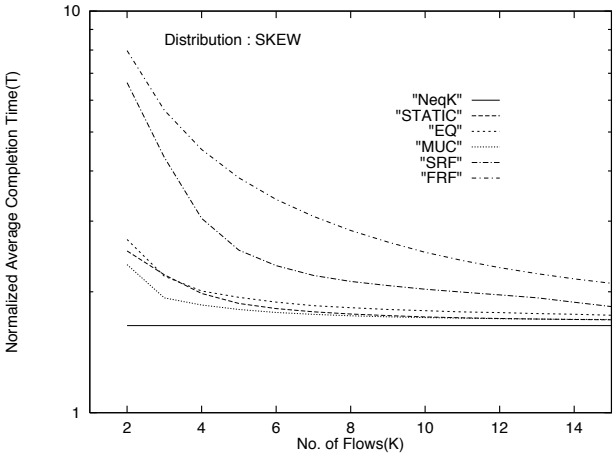

Figure 9: T vs. K for SKEW

6 Related Work

Transmission rate control for multicast communication has come to be recognized as an important issue ([1, 2]). Proposals for high-speed local area networks include the use of packet-level admission control and buffer reservation in the network([3]) and the use of a tree-like hierarchy of multicast groups to control the sender rate ([4]).

Real-time applications like video have received particular attention in the study of multicast rate control in wide-area networks ([5, 6, 7]). [5] describes mechanisms for soliciting receiver response in a scalable manner and adjusting video transmission rate based on it. The issue of how to satisfy receivers with diverse requirements (which we consider here) was left open in [5]. ([6]) has proposed algorithms for handling heterogeneous receivers - throttling the sender according to the needs of the majority of receivers, according to the bottleneck receiver, etc. However. similar approaches for non real-time loss-sensitive approaches are largely unexplored.

Multilevel coding has been proposed as a possible approach for accommodating heterogeneity among receivers of encoded video ([2, 8, 7].The receiver-driven layered multicast (RLM) protocol([8]) has been designed to combine a layered source encoding approach with a receiver-oriented layered transmission scheme using multiple multicast groups. Though our work has drawn inspiration from RLM, we have studied a very different application domain. Eventual delivery of all data to all receivers is a requirement for our target applications but not so for loss-tolerant video applications as considered for RLM.

The use of multiple multicast groups has been proposed for both error recovery([10, 13]) and congestion/flow control ([11, 12, 13]). To the best of our knowledge, [13] is the only ongoing work in the design of multicast congestion control mechanisms for bulk data transfer applications. The key idea in this research is to improve efficiency through a "TCP friendly" congestion control mechanism using multiple multicast groups. Receivers are grouped on the basis of data loss correlation; loss statistics are used to drive the congestion avoidance algorithm. [12] has proposed the idea of splitting a set of receivers into mutually exclusive subsets in order to improve the throughput of positive acknowledgment based point-to-multipoint protocols. The sender carries on independent conversation with each subset over a separate multicast channel - this is the simulcast approach described in section 4.1. Our work differs from this in several respects. First, we have separated the issues of flow control and error control by not considering any specific

error-recovery mechanism. Second, we have proposed a nested transmission scheme that has been shown to achieve significant bandwidth gains over simulcast. Third, we have explored solutions with a restricted number of multicast channels, whereas [12] implicitly assumes the availability of an arbitrary number of channels.

In the realm of reliable multicasting, there has been a proliferation of transport protocols targeting applications such as file transfer ([15, 16, 14]), DIS ([17]), whiteboards ([9]), etc. Proposals for transmission rate control ranges from traditional window-based flow control([15]) to newer approaches like negative acknowledgment based control([16]). A survey of these protocols reveals the weakness of the "one size fits all" paradigm, as has been reiterated in ([9]).

7 Conclusions

This paper describes an approach towards multicast flow control using multiple multicast groups. Data is transmitted at different rates to receivers with different capacities, so as to minimize the average completion of all the receivers. Our algorithms for choosing transmission rates for the multicast channels are simple and demonstrate good performance for large and heterogeneous receiver groups. In addition, several of our proposed transmission schemes use bandwidth efficiently and perform well even with a small number of multicast groups.

There are a number of directions in which this work can be extended. In order to achieve a clear separation between flow control and error control, we have assumed a loss-free network. It remains to be seen how our flow control approach can be combined with various error-recovery techniques. In particular, [20] shows that multicast losses in the MBone are spatially and temporally correlated. It may be beneficial to take this into account when grouping receivers ([13]).

We have assumed that the sender has perfect knowledge about receiver rates. Though we have not discussed specific mechanisms for gaining this knowledge, one possibility is to use to probe tools described in [21]. Also, we have examined flow control only for static, time-invariant bottleneck conditions. End-to-end rate control for dynamic variations in the network condition (i.e., congestion) remains an open problem. Finally, other types of long-lived applications and applications with data priority (e.g., whiteboard) need to be considered. We anticipate that the performance criteria for evaluating flow control schemes for these applications will be different from the ones that we have considered, thereby necessitating flow control algorithms that are potentially different from the ones presented in this paper.

References

[1] C. Diot, W. Dabbous, J. Crowcroft. Multipoint Communication: A Survey of Protocols, Functions and Mechanisms. *To appear in IEEE JSAC*.

[2] T, Turletti, J.C. Bolot. Issues with Multicast Video Distribution in Heterogeneous Packet Networks. *In Proceedings of 6th International Workshop on PACKET VIDEO, Portland, Oregon, 26-27 Sept. 1994, pp. F3.1-3.4.*

[3] X. Chen, L.E. Moser, P.M. Melliar-Smith. Flow Control Techniques for Multicasting in Gigabit Networks. *In Proceedings of International Conference on Network Protocols, 1996.*

[4] D.H.H. Guerney. Multicast Flow Control in Local Area Networks. *Technical Report TR95-1479, Cornell University, Ithaca, NY.*

[5] J.C. Bolot, T. Turletti, I. Wakeman. Scalable Feedback Control for Multicast Video Distribution in the Internet. *In Proceedings of ACM Sigcomm, 1994*.

[6] R. Yavatkar, L. Manoj. Optimistic Strategies for Large-Scale Dissemination of Multimedia Information. *In Proceedings of ACM Multimedia, 1993*.

[7] H. Kanakia, P.P. Mishra, A. Reibman. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. *In Proceedings of ACM Sigcomm, 1993*.

[8] S. McCanne, V. Jacobson, M. Vetterli. Receiver-driven Layered Multicast. *In Proceedings of ACM Sigcomm, 1996*.

[9] S. Floyd, V. Jacobson, S. McCanne, C Liu, L. Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *Extended version of paper in Proceedings ACM Sigcomm, 1995*..

[10] S.K. Kasera, J.F. Kurose, D. Towsley. Scalable Reliable Multicast Using Multiple Multicast Groups. *In Proceedings ACM Sigmetrics, 1997*.

[11] S.Y. Cheung, M.H. Ammar, X. Li. On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. *Tech Report: GIT-CC-95-25, Georgia Institute of Technology, Atlanta, GA. July 1995*.

[12] M.H. Ammar, L. Wu. Improving the Throughput of Point-to-Multipoint ARQ Protocols Through Destination Set Splitting. *In Proceedings of IEEE Infocom, 1992*.

[13] L. Vicisano, M. Handley, J. Crowcroft. B-MART, Bulk-data (non-real-time) Multiparty, Adaptive Reliable Transfer Protocol. *Draft available from ftp://cs.ucl.ac.uk/darpa at University College, London*.

[14] StarBurst MFTP Compared to Today's File Transfer Protocols : A White Paper. *Draft available from http://www.starburst.com/white.htm*.

[15] J.C. Lin, S. Paul. RMTP : A Reliable Multicast Transport Protocol. *In Proceedings IEEE Infocom, 1996*.

[16] A. Koifman, S. Zabele. RAMP : A Reliable Adaptive Multicast Protocol. *In Proceedings IEEE Infocom, 1996*.

[17] H.W. Holbrook, S.K. Singhal, D.R. Cheriton. Log-Based Receiver Reliable Multicast for Distributed Interactive Simulation. *In Proceedings of ACM Sigcomm, 1995*.

[18] S. Keshav. A Control Theoretic Approach to Flow Control. *In Proceedings of ACM Sigcomm, 1991*.

[19] P.B. Danzig. Flow Control for Limited Buffer Multicast. IEEE Trans. on Software Engineering. Vol. 20, no. 1, Jan. 1994.

[20] M. Yajnik, J.F. Kurose, D, Towsley. Packet Loss Correlation in the MBone Multicast Network. *In Proceedings of IEEE Global Internet Conference, 1996*

[21] R.L.Carter, M.E. Crovella. Measuring Bottleneck Link Speed in Packet Switched Networks. *In Performance Evaluation, Vol. 27 & 28, 1996*.

[22] D.P. Bertsekas. Dynamic Programming : Deterministic and Stochastic Models. *Prentice-Hall, Inc*.

Appendix

A Proofs of Properties of A_{ideal}

First, we make the following useful observations about A_{ideal} :

Lemma 4 *No part of the data stream is transmitted more than once over channels* 1 *through* $K-i+1$ *in steps* 1 *through* $i-1$, $i = 1, 2, \cdots K$.

The proof is by induction. The result is trivially true for $i = 1$. For $i = 2$, $\mathcal{Q}_{k,i-1} = \mathcal{B}_{k,i-1}$, hence the result is obviously true for $i = 2$. Let it be true for $i = m, \quad 2 \leq m \leq K$. This implies that sets $\mathcal{Q}_{k,m-1}$ are disjoint, for $k = 1, 2, \ldots, K-m+1$. In step $m+1$, the set of segments transmitted on channel k, $k = 1, 2, \cdots K-m$, is

$$\mathcal{B}_{k,m+1} = \{g_{k,K-m}(a) | a \in \mathcal{Q}_{K-m+1,m-1}\}$$

Since $\mathcal{Q}_{K-m+1,m-1}$ is disjoint with every $\mathcal{Q}_{k,m-1}$, $k = 1, \ldots, K-m$, the lemma is true for $i = m+1$. Hence the proof.

Lemma 5 *Receiver* $K-i+1$ *completes at the end of step* i.

The proof is by induction. Receiver K obviously receives the entire stream in step 1, hence the result is true for $i = 1$. Let it be true for $i = m$, i.e., receiver $K-m+1$ completes at the end of step m. The only channel that receiver $K-m+1$ subscribes to but receiver $K-m$ does not, is channel $K-m+1$. So at the end of step m, the only set of segments that has yet to be received by receiver $K-m$ is $\mathcal{Q}_{K-m+1,m-1}$. From Lemma 1, it follows that none of the data segments in set $\mathcal{Q}_{K-m+1,m-1}$ has been received by receiver $K-m$ till the end of step m. In step $m+1$, the set $\mathcal{Q}_{K-m+1,m-1}$ is transmitted over channels 1 through $K-m$. So receiver $K-m$ completes at the end of step $m+1$. Hence the proof.

Based on lemmas 4 and 5, we now prove properties $P1$ through $P4$:

Property P1 *All* K *receivers receive the complete data stream at the end of step* K.

This follows directly from lemma 2.

Property P2 A_{ideal} *is duplicate-free*.

Receiver $K-i+1$ subscribes to channels 1 through i. From Lemma 1, it follows that this receiver receives no duplicates till the end of step $i-1$. The set of segments that it receives in step i is $\mathcal{Q}_{K-i+2,i-1}$. This is disjoint with $\mathcal{Q}_{k,i-1}$ for $k = 1, 2, \cdots K-i+1$ (lemma 1). Also, receiver $K-i+1$ completes at the end of step i. So it never receives any duplicate data. This is true for every receiver. Hence the proof.

Property P3 *A_{ideal} is work-conserving.*

When a given data segment of size l is partitioned among n channels, then the size of the sub-segment assigned to channel m $(1 \leq m \leq n)$, is $(r_m / \sum_{j=1}^{n} r_j) * l$. The time taken to transmit this segment is $(l / \sum_{j=1}^{n} r_j)$. This means that every channel takes exactly the same amount of time to transmit the sub-segment assigned to it and this is true for every data segment. So at every step i, none of the channels 1 through $K - i + 1$ is ever idle. Hence A_{ideal} is work-conserving.

Property P4 *Of all transmission schedules that attain the ideal completion time for every receiver, A_{ideal} is optimal in terms of transmission volume.*

Consider receivers $K - i + 1$ and $K - i$ at the end of step i. Receiver $K - i + 1$ has finished receiving the entire stream (Lemma 2) while $K - i$ is still missing $\mathcal{Q}_{K-i+1,i-i}$. From Lemma 1, we can deduce that no part of this data has been delivered to receiver $i - 1$. So this is the minimum amount of data that needs to be delivered to finish it. Under A_{ideal}, this is the exact amount of data that is sent in step $i + 1$. Applying this reasoning at each step, we can conclude that the total volume of data transmitted under A_{ideal} is the minimum required to complete every receiver in its ideal completion time. Hence A_{ideal} is optimal in terms of transmission volume.

B Proof of Lemma 2

The proof is by contradiction. Let us assume that there is an optimal channel rate assignment for which the result does not hold, i.e., there is at least one i ($<= i <= K$), such that $F_i \notin \{R_1, ...R_N\}$. Let receivers $n_1, n_1 + 1, ...n_2$ subscribe to channels 1 through i, i.e. $F_i < R_{n1} < ... < R_{n2} < F_{i+1}$. This implies that each of them receives at rate F_i. Now let us change the channel rates so that $F_i = R_{n1}$ but $F_j,\ j \neq i$, remains the same. With this new rate assignment, each of receivers n_1 through n_2 receive at a higher rate and hence finish in less time. However, this change in F_i *does not* affect the completion time of any of the other receivers. Hence there is an overall decrease in the average completion time of all the receivers. This contradicts the assumption that the original assignment was optimal. Hence the proof.

C Proof of Lemma 3

We prove the existence of a conflict-free and work-conserving schedule by constructing one. Let $\mathcal{B}_{k,t}$ be the set of segments that are transmitted on channel k in stage t and let function $g_{m,n}$ be defined as before.

With a subscription policy π_c, we build a transmission schedule A that has two phases :

- Phase 1 : There are more than K unfinished receivers, so all K channels are used in each stage. The first stage is identical to that of A_{ideal} described in section 3, i.e.

$$\mathcal{B}_{k,1} = \{((\sum_{j=1}^{k-1} r_j / \sum_{j=1}^{K} r_j) * L, (r_k / \sum_{j=1}^{K} r_j) * L)\}$$

  For stage $t > 1$, let receivers v^t through u^t subscribe to all K channels, where $(R_{v^t} < R_{v^t+1} < \cdots < R_{u^t})$. v^t and u^t are determined by the channel rate assignment algorithm. Starting with the fastest

```
Sent = {};
For (n = u^t downto v^t) do
    For (τ = 1 to t − 1 do
        For (j = k_n^τ + 1 to K) do
            Curr = B_{j,τ} - Sent;
            For k = 1 to K Do
                B_{k,t} = B_{k,t} ⋃ {g_{k,K}(a) | a ∈ Curr};
            Sent = Sent ⋃ Curr;
            endfor;
        endfor;
    endfor;
endfor;
```

Table 2: Algorithm for transmission stage $t > 1$ in phase 1 of schedule A.

receiver u^t, the sender transmits all the segments required to finish each of the receivers subscribing to channel K. At the end of the stage, receivers 1 through $v^t - 1$ are left unfinished. The algorithm for Phase 1 is presented in Table 2.

- Phase 2 : There are K or fewer unfinished receivers. If there are u^t unfinished receivers at the beginning of stage t, then it is possible to match the rates of all of them by using only u^t channels (as shown in section 3). All the remaining channels $u^t + 1$ through K are released. In each stage, the sender transmits the data segments that have yet to be received by receiver u^t (Table 3). Therefore, at the end of stage t, receivers 1 through $u^t - 1$ are left unfinished.

We observe that every data segment to be transmitted is partitioned among channels so that the length of the segment assigned to each channel is proportional to the channel rate. This ensures that schedule A is work-conserving.

The following result is important for showing that schedule A is conflict-free. Let D_n^t be the set of data segments that receiver n has received till the end of stage t.

Lemma 6 *For every transmission stage t and for every receiver pair (x, y) :*

$$R_x < R_y \implies D_x \subseteq D_y.$$

Proof : It follows from the definition of π_c that for every pair of unfinished receivers (x, y) :

$$R_x < R_y \implies S_x \subseteq S_y.$$

This implies that in each stage, a slower receiver receives only a subset of the data segments received by a faster receiver. Hence the proof.

```
For (τ = 1 to t − 1 do
    For (j = k^τ_{u^t} + 1 to K) do
        For k = 1 to u^t Do
            B_{k,t} = B_{k,t} ⋃ {g_{k,K}(a) | a ∈ B_{j,τ}};
        endfor;
    endfor;
endfor;
```

Table 3: Algorithm for transmission stage t in phase 2 of schedule A.

We observe that in schedule A, the next segment chosen for transmission is always one that has not been received by the fastest unfinished receiver. From Lemma 6, we conclude that it has not been received by any of the other unfinished receivers. This ensures that A is conflict-free.