A Hybrid Adaptation Protocol for TCP-Friendly Layered Multicast and Its Optimal Rate Allocation*

Jiangchuan Liu[1**], Bo Li[1], and Ya-Qin Zhang[2]

[1] Department of Computer Science
The Hong Kong University of Science and Technology
{csljc,bli}@cs.ust.hk

[2] Microsoft Research, Asia
No.49, Zhichun Road, Beijing, 100080, China
yzhang@microsoft.com

Abstract- **Layered transmission has been proposed as a solution to video multicast over the Internet. Existing protocols usually perform adaptation at the receiver's side and use static rate allocation techniques at the sender's side. As a result, significant mismatches between the fixed transmission rates and the heterogeneous and dynamic rate requirements from the receivers could still occur.**

In this paper, we show that such mismatches can be minimized by employing dynamic layer rate allocation at the sender's side by taking advantage of the recent development in layered video coding. Specifically, we study the optimization criteria for layer rate allocation, and propose a metric called *Fairness Index*, which fairly reflects the degree of a receiver's satisfaction. We then formulate this into an optimization problem with the objective of maximizing the expected fairness index, and derive an efficient and scalable algorithm to solve it. We further demonstrate that such sender rate adaptation can be seamlessly integrated into an end-to-end adaptation protocol called HALM (Hybrid Adaptation Layered Multicast). This protocol is designed for the current best-effort Internet and is TCP-friendly. Its control overhead is also kept at a low level.

Simulation results show that HALM improves the degree of fairness for receivers with heterogeneous bandwidth requirements, and interacts with TCP flows substantially better than static allocation based protocols. In addition, increasing the number of layers in HALM always leads to a higher degree of fairness and usually, 3 to 5 layers is sufficient. However, this is not true for the static allocation based protocols.

I. INTRODUCTION

Real-time video distribution over the Internet using IP multicast is an important problem. It can be used in webcast, video on demand, and distance learning, etc. In the current Internet, only best-effort service is provided and TCP is the dominant traffic, it is better for a video multicast protocol to be adaptive and *TCP-friendly*, i.e., not overwhelm the congestion-sensitive TCP traffic [13]. In addition, since users differ greatly in their bandwidths and processing capabilities, a multicast protocol should also ensure *intra-session fairness*, which means each user in a multicast session receives video data at a rate compatible with its capacity, regardless of the capacities of other users [3]. However, the Internet's intrinsic heterogeneity and large scale make these two objectives difficult to achieve.

In a traditional unicast protocol, a sender collects feedback from a receiver and adjusts its transmission rate accordingly [13,14,33]. In a multicast scenario, however, feedback from too many receivers can easily cause feedback implosion [3,4]. In addition, if only a single transmission rate is used (like that in the unicast), the conflicting requirements of a set of heterogeneous receivers cannot be satisfied simultaneously, i.e., receivers with lower capacities may suffer congestion while receivers with higher capacities may have their capacities underutilized.

To solve these problems, previously proposed multicast protocols use receive-driven layered transmission [1,3,11]. In this approach, a raw video is compressed into a number of layers. The layer with the highest importance, called *base layer*, contains the data representing the most important features of the video, while additional layers, called *enhancement layers*, contain data that progressively refine the reconstructed video quality. Usually, the layer rates are statically allocated at the sender's side. Each layer is delivered through a separate multicast group, and a receiver joins as many groups as its capacity allows.

Since the rate of each layer is fixed in this approach, significant mismatches between the fixed layer rates and the capabilities (e.g., bandwidths) of the receivers can still occur. In this paper, we show that such mismatches can be significantly reduced by using sender-driven adaptation as a complement to receiver-driven adaptation. For example, the bandwidths of the receivers in a session usually exhibit clustered patterns (i.e., receivers who share the same uplink generally experience the same bottleneck bandwidth), and by dynamically adjusting the layer rates to match these clusters, the overall system performance can be improved. Specifically, we study the optimization criteria for layer rate allocation, and propose a metric called *Fairness Index*, which fairly reflects the degree of a receiver's satisfaction. We then formulate this into an optimization problem with the objective of maximizing the expected fairness index, and derive an efficient and scalable algorithm to solve it.

We further demonstrate that such sender rate adaptation can be seamlessly integrated into an end-to-end adaptation protocol called HALM (Hybrid Adaptation Layered Multicast). This protocol is designed for the current best-effort Internet and is TCP-friendly. In HALM, we also exploit the advantages of several existing TCP-friendly layered multicast protocols [7,11,12]. We do this by incorporating additional information in the receiver feedback messages provided by these protocols. As a result, HALM's control overhead is comparable to the overhead of these protocols.

The performance of HALM has been extensively studied through simulation and statistical analysis under a variety of configurations. Our results show that HALM interacts with TCP traffic better than static allocation based protocols. Its optimal layer rate allocation algorithm usually outperforms

[*] The research was supported in part by grants from Research Grant Council (RGC) under contracts AoE/E-01/99 and HKUST6163/00E.

[**] Jiangchuan Liu's work is partially supported by a Microsoft research fellowship.

traditional static allocation techniques by 10%-20% in terms of the expected fairness index, thanks to its adaptability to the available bandwidth distributions. With the optimal allocation algorithm, increasing the number of layers always leads to better performance and usually 3 to 5 layers is sufficient. However, we found that this is not the case for the static allocation techniques.

The remainder of this paper is organized as follows. Section II presents some related work. Section III gives an overview of our protocol. Section IV formulates the optimal allocation problem and presents efficient allocation algorithms. Section V discusses the parameter settings for HALM and the control overhead. Section VI evaluates the performance of HALM through simulations and statistical analysis. Finally, Section VII concludes the paper and discusses some future directions.

II. Related Work

A. *Scalable Video Coding*

In the coding community, *scalable coding* is frequently used to refer to layered coding. The scalability can be achieved by scaling the frame speed (*temporal scalability)*, frame size (*spatial scalability)* and frame quality (*quality or SNR scalability*) [28]. These scalable coding algorithms have been adopted in advanced compression standards, such as H.263+, MPEG-2 and MPEG-4. HALM does not specify any particular coding algorithm in the application layer. It can cooperate with all these scalable coders. However, a coder with a wide dynamic range and fine granularity in terms of rate control is of particular interest, such as the MPEG-4 *Fine Granularity Scalability* (FGS) [28] or *Progressively FGS* (PFGS) coders [29]. The key technique used in these two scalable coding algorithms is *bit-plane coding* [28]. By this technique, layer rate is allocated after compression using an assembling/packetization procedure, which is different from traditional techniques that perform rate control during compression by adjusting quantizers. Hence, it has very fast responsiveness for rate allocation and incurs low overhead for layer synchronization. Bit-plane coding has been considered as a very promising technique and has been adopted in the MPEG-4 standard [28].

B. *TCP-Friendliness*

Using TCP for real-time video delivery is not practical because these applications usually require a smoothed transmission rate and have stringent demand on delay. However, since a dominant portion of today's Internet traffic is TCP-based, streaming video should have some rate control mechanism to ensure that video traffic will not overwhelm the congestion-sensitive TCP traffic. TCP-friendliness is one solution that has been proposed. On a long-term basis, a TCP-friendly video stream fairly shares the bottleneck bandwidth[1] with the TCP connections running over the same path [13]. Usually this is done by using an equation to estimate the long-term TCP throughput over that path and adjusting the transmission rate accordingly [13,14]. There has been significant research on modeling TCP throughput [10]. The conclusion is that such an equation relies on the packet size, loss event rate, and Round-Trip Time (RTT) [10,13]. The estimations of some parameters, such as RTT, require feedback packets. Therefore, in a heterogeneous multicast environment, how to efficiently estimate these parameters for a large number of sender-receiver pairs becomes a challenging issue. This is one of the key issues that is addressed in HALM.

C. *Layered Multicast*

Layered multicast has been a hot topic in the networking community as well as the coding community for several years [1,3,5,7,18,22,23]. McCanne, Jacobson and Vetterli [1] proposed the first practical receiver-driven adaptation algorithm for layered video multicast over the Internet. This algorithm, known as Receiver-driven Layered Multicast (RLM), sends each video layer over a separate multicast group. A receiver periodically joins a higher layer's group to explore the available bandwidth. If packet loss is detected after the join-experiment, the receiver will leave the group. This control loop continues during the transmission. RLM has been considered as a promising approach for adaptive video transmission. First, it is fully compatible with the current best-effort Internet infrastructure. Second, it is scalable and works well with heterogeneous receivers because adaptation is performed by the receivers.

However, it is well known that the original RLM is not TCP-friendly [3,7,11]. Some improvements have been proposed which use equation-based rate control on the receiver's side [7,11]. As shown in [26], these static allocation based approaches are still not fair to some receivers because their choices are restricted to a discrete set of layer rates which may not match their requirements. An effective way to improve fairness is hybrid adaptation which uses sender-based dynamic layer rate allocation in conjunction with receiver-driven adaptation. Shacham [8] in his pioneer work has shown a rate allocation algorithm to maximize the aggregate (or equivalently, average) signal quality of all the receivers for hierarchically encoded data transmission. His target network employs a fixed, optimal routing scheme for a given traffic mix. Hybrid adaptation protocols for the best-effort Internet include the Multicast Enhanced Loss-Delay based Adaptation (MLDA) protocol [12] and the SIM protocol [23]. However, the allocation algorithms in MLDA and SIM are not optimal. For example, in the current implementation of MLDA, the layer rates are uniformly allocated between the maximum and minimum bandwidth requirements. The distinct feature of our HALM protocol is

[1] Note that there is still no consensus on the exact definition of *fairness* between layered multicast traffic and unicast traffic [31,32]. It is still a question whether the layered video and a TCP session should get the same bandwidth share (*absolute fairness*) [31] or, compared with the TCP bandwidth *b*, the video bandwidth has a bounded value *Cb*, for some constant *C* (*bounded fairness*) [31]. HALM adopts the notion of absolute fairness. However, by using the similar techniques in [32], it can be extended to the scenario where bounded fairness is adopted. In addition, it is hard for a layered multicast protocol to be totally fair because the adaptation unit is a layer at the receiver's side [26]. Therefore, the objective of HALM is to achieve a fair share as close as possible by using optimal rate allocation at the sender's side and layer adaptation at the receiver's side.

that it embeds an optimal allocation algorithm which is based on the distribution of the receivers' bandwidths. It also adopts an efficient feedback mechanism for both sender-based layer rate allocation and receiver-based TCP-friendly bandwidth estimation.

Another attractive layered multicast approach is the prioritized transmission [5,18]. In this approach, the sender assigns different priorities to the layers according to their levels of importance, and, during congestion, routers drop low priority packets first. This approach is very stable because the high priority packets are always well protected. In addition, if flow-isolation is implemented in routers, being TCP-friendly is not a requirement any more [6]. The Source Adaptive Multi-layered Multicast (SAMM) protocol [5] is based on this assumption. A SAMM sender adjusts the rates of the layers according to the receivers' bandwidth reports. To avoid feedback implosion, these reports are not sent directly to the sender but to some feedback mergers which use a heuristic algorithm to merge the information before forwarding them to the sender. Nevertheless, these scheduling policies are more complex than the current FIFO drop-tail policy. HALM only requires drop-tail routers, but its optimal rate allocation algorithm can also be combined with prioritized transmission.

III. Overview of the Hybrid Adaptation Protocol for Layered Multicast (HALM)

HALM works on top of the RTP protocol [2]. The underlying packet delivery model is the group-oriented IP multicast model [20,21] which provides only best-effort service. Moreover, HALM uses pure end-to-end control in which all functionalities are implemented at end systems (sender or receiver). Therefore, HALM is fully compatible with the current Internet infrastructure.

A. Sender functionality

The sender encodes the raw video into l cumulative layers using a layered coder, where layer 1 is the base layer and layer l is the least important enhancement layer. Let c_j denote the cumulative layer rate up to layer j, and ρ_l denote the rate vector, $\rho_l = (c_1, c_2,\dots, c_l)$. With the cumulative subscription policy, this discrete set offers all possible video rates that a receiver in the session could receive. In particular, the maximum rate that a receiver with an expected bandwidth r can receive is given by $\Gamma(r,\rho_l)=\max\{c : c \le r, c \in \rho_l\}$. Note that there is a gap between this receiving rate and the expected bandwidth of the receiver. To minimize this gap, the sender also collects the bandwidth reports from the receivers. Assume the session size is N and the receivers' expected bandwidths are $\{r_1,r_2,\dots, r_N\}$. The sender will adjust the layer rates based on the bandwidth distribution with a control period of T_{ctrl} sec.

The sender generates reports to all the receivers every T_{SR} sec. A report packet, SR, includes the RTP synchronization source identifier (SSRC) [2], a timestamp, the current rate vector and a response list for receivers' requests. Since a rate vector is different from the one in the previous control period (if they are the same, we can offset the current vector by a small value.), this adjustment can serve as an implicit synchronization signal to trigger the receivers' joining/leaving actions. Note that this implicit signal can be detected even if some SR packets are lost.

B. Receiver functionality

In order to be friendly with TCP, a receiver directly uses a TCP throughput function to calculate its expected bandwidth. There has been significant research on modeling TCP throughput. The conclusion is that such an equation relies on the packet size, loss event rate, and RTT [10,13]. HALM provides a general framework to measure these parameters; hence, different equations can be easily embedded in HALM, e.g., the one from [10]. The following control loop is performed at a receiver's end:

1: Measures or estimates loss event rate, RTT;

2: If receives an SR with a new rate vector, goto 3, else goto 1;

3: Stores the rate vector to ρ_l , and calculates the equivalent TCP throughput B.

4: Calculates K using $K=\max\{k : c_k \le B, c_k \in \rho_l\}$. Joins or leaves layers until the subscription level is K.

5: Goto 1.

Note that the average rate at which the video stream is delivered from the source to the receiver is equivalent to or less than a long-term TCP connection running over the same path. Thus, we have obtained a TCP-friendly scheme suitable for multicast delivery in a heterogeneous environment. In addition, this scheme is scalable because the receivers' joining/leaving actions are synchronized so that no coordination, or shared learning [1], is needed for join-experiments. It is also very robust because the implicit signal will be detected even if some SR packets are lost.

A receiver also generates report packets every T_{RR} sec. A report packet, RR, contains the SSRC of the receiver and its expected bandwidth. It also serves as a request for RTT estimation.

IV. Sender-based Dynamic Rate Allocation

In this section, we consider the layer rate allocation strategy on the sender's side. Two fundamental issues are addressed. First, what is an optimal allocation? Second, how is the optimal allocation achieved? We present a fairness index that is suitable for evaluating the receivers' satisfaction in a heterogeneous environment. We then formulate the problem of optimal allocation and provide efficient algorithms for solving this problem.

A. Optimization Criteria

The optimization objective for many multicast protocols is to maximize the aggregate bandwidth of a session [8]. It is, however, not a suitable metric for optimizing the degree of fairness in a heterogeneous environment. For example, such algorithms always try to satisfy a receiver with huge bandwidth, and at the same time, might sacrifice a number of receivers with relatively narrow bandwidth requirements.

Since, with a cumulative subscription policy, the subscription level of a receiver relies on its expected bandwidth and the set of cumulative layer rates, we define the *Fairness Index* $F(.)$ for a receiver with expected bandwidth r as $F(r,\rho_l)=\Gamma(r,\rho_l)/r$. This definition can be used to access the satisfaction of a receiver when there is performance loss incurred by a mismatch between the discrete set of the possible receiving rates and its expected bandwidth. Since the expected bandwidth is estimated as the throughput of a TCP connection over the same path, this index also reflects the degree of fairness when HALM traffic competes with TCP traffic. For a multicast session, the optimization objective is to maximize the expected fairness index, $\overline{F}(r, \rho_l)$, for all the receivers by choosing an optimal rate vector. We formally state the optimization problem as follows:

$$\text{(P1)} \qquad \text{Maximize } \overline{F}(r, \rho_l) = \frac{1}{N}\sum_{i=1}^{N} F(r_i, \rho_l), \qquad (1)$$

$$\text{Subject to } \; l \leq L,$$
$$0 < c_{i\text{-}1} < c_i, \; i=2,3,\ldots,l.$$

where L is the maximum number of layers that the sender can manage.

The complexity of this optimization problem can be further reduced by considering some characteristics of a practical layered coder. First, every lossy data compression scheme has only a finite set of quantizers. Therefore, there are only a finite number of possible rates for any given source. Second, to avoid the undesired situation where a receiver cannot join any layer, the base layer should adapt to the minimum expected bandwidth. However, the rate of the base layer usually has a lower bound [28,29]. Taking these factors into account, we assume there are M operational points. The set of operational rates is given by $\pi = \{ R_1, R_2, \ldots, R_M : R_i < R_{i+1}\}$, and R_1 is the lower bound of the base layer rate. We can then re-formulate the optimization problem as follows:

$$\text{(P2)} \qquad \text{Maximize } \overline{F}(r, \rho_l) = \frac{1}{N}\sum_{i=1}^{N} F(r_i, \rho_l), \qquad (2)$$

$$\text{Subject to } \; l \leq \text{L},$$
$$c_1 = \max_j \{R_j : R_j \leq \min_i \{r_i : r_i \geq R_1\}\},$$
$$c_i \in \pi, \; c_{i\text{-}1} < c_i, \; i=2,3,\ldots,l.$$

B. *Optimal Allocation Algorithms*

Note that the receivers can be divided into l sets according to their subscription levels; in each set the receivers have the same receiving rate. Assume $c_{l+1} \to \infty$, the expected fairness index can be calculated as follows,

$$\overline{F}(r, \rho_l) = \frac{1}{N}\sum_{j=1}^{l} \sum_{c_j \leq r_i < c_{j+1}} F(r_i, \rho_l)$$
$$= \frac{1}{N}\sum_{j=1}^{l} \sum_{c_j \leq r_i < c_{j+1}} \frac{\Gamma(r_i, \rho_l)}{r_i} = \frac{1}{N}\sum_{j=1}^{l} \left[c_j \sum_{c_j \leq r_i < c_{j+1}} \frac{1}{r_i} \right]. \quad (3)$$

Let $\varphi(m,l) = \max_{c_l = R_m} \overline{F}(r, \rho_l)$, i.e., the maximum expected fairness index when c_l is set to the mth operational point, R_m. we have the following recurrence relation,

$$\varphi(m,l) = \begin{cases} \frac{1}{N} R_m \sum_{r_i \geq R_m} \frac{1}{r_i}, & \mathit{if}\; l=1, R_m = c_1, \\ \max_{1 \leq j < m} \left\{ \begin{array}{l} \varphi(j, l-1) + \\ \frac{1}{N}(R_m - R_j) \sum_{r_i \geq R_m} \frac{1}{r_i} \end{array} \right\}, & \mathit{if} \;\; l > 1, \\ 0, & \mathit{Otherwise} \end{cases} \qquad (4)$$

Lemma 1. $\overline{F}(r, \rho_l^*)$ is non-decreasing with respect to l, where ρ_l^* is an optimal allocation with l layers.

Proof:

$$\overline{F}(r, \rho_l^*) = \max_{1 \leq m \leq M} \varphi(m, l)$$
$$= \max_{1 \leq m \leq M} \max_{1 \leq j < m} \left\{ \varphi(j, l-1) + \frac{1}{N}(R_m - R_j) \sum_{r_i \geq R_m} \frac{1}{r_i} \right\}$$
$$\geq \max_{1 \leq m \leq M} \max_{1 \leq j < m} \varphi(j, l-1)$$
$$= \max_{1 \leq j < M} \varphi(j, l-1) = \overline{F}(r, \rho_{l-1}^*). \qquad (5)$$

Remark 1. For heterogeneous receivers, increasing the number of layers never decreases the expected fairness index if the allocation scheme is optimal. From the viewpoint of improving fairness, a source with more layers is thus more desirable. Though the conclusion is intuitive, we will show in Section VI that this appealing property does not hold with static allocation schemes.

Theorem 1: $\overline{F}(r, \rho_L^*) = \max_{1 \leq m \leq M} \varphi(m, L)$ is the solution to the optimization problem P2.

Proof: From Lemma 1, $\overline{F}(r, \rho_L^*) \geq \overline{F}(r, \rho_l^*), l < L$.

Remark 2. Note that (4) relies only on the aggregate features of the receiver bandwidths, such as $\sum_{r_i \geq R_m} 1/r_i$, which can be pre-calculated during the bandwidth collection process. Therefore, the above result directly leads to a dynamic programming algorithm with time complexity O(LM^2) and auxiliary storage space O(LM).

Next, we show that with some modifications, the algorithm for P2 can also be applied to solve problem P1.

Lemma 2. $\rho_L^* \subseteq \{ r_1, r_2, \ldots, r_n \}$.

Proof: Assume $\rho_L^* = \{c_1^*, \ldots, c_k^*, \ldots, c_l^*\}$ where $c_k^* \notin \{r_1, r_2, \ldots, r_N\}$. We can construct another layer rate vector, $\rho_L^{'} = \{c_1^*, \ldots, c_k^{'}, \ldots, c_l^*\}$, where $c_k^{'} = \min\{c : c \in \{r_1, r_2, \ldots, r_N\}, c_k^* < c\}$. It is easy to show that $\overline{F}(r, \rho_L^{'}) \geq \overline{F}(r, \rho_L^*)$, which contradicts that ρ_L^* is optimal.

Theorem 2. Optimization problem P1 can be solved by a dynamic programming algorithm with time complexity O(LN^2) and auxiliary storage space O(LN).

Proof: Based on Lemma 2, using $\{r_1, r_2, \ldots, r_n\}$ instead of π in the algorithm for problem P2.

Note that bandwidth is a network measurement; the fairness definition in this section implies that to optimize the perceptual video quality is the same as to optimize the network utilization. In other words, they have a linear relationship. However, this is found not true because existing studies show that these two assessments generally exhibit a somewhat non-linear relationship [30]. Normally, this non-linearity can be characterized by a utility function $U(r)$, which maps the rate r delivered by the network into an application-aware performance measure [18], and consequently an *application-aware fairness index* can be calculated [34,35]. In the context of video transmission, the utility function can be obtained from the well-established rate-distortion relations [30], or other perceptual video measures [29,30]. We have proved that, with some modifications, the algorithms presented in this section still work in this context, see [34,35].

C. Computation Overhead

Let $T_N^{P_1}$ denote the execution time for solving problem P1 with N receivers and 5 layers. On a Pentium III 450 MHz PC, the execution times are $T_{500}^{P_1}$ = 32 ms, $T_{1000}^{P_1}$ = 126 ms, $T_{2000}^{P_1}$ = 514 ms, and $T_{3000}^{P_1}$ = 1154 ms. The execution time of the algorithm for solving P1 depends on the number of the receivers, which means this algorithm is not scalable. However, for a small group and a layered coder with fine-granular rate control (M>>N), it is still an efficient algorithm for optimal rate allocation.

On the other hand, given the number of layers and the number of operational points, the execution time for solving problem P2 is constant. For example, when M=500 and L=5 (note that this setting corresponds to the fine granular rate control in state-of-art layered video coders [29]), the execution time is always 32 ms. As the complexity does not depend on the number of receivers, it is highly scalable and can be applied to large sessions for real-time adaptation. Moreover, since it does not need to explicitly know the bandwidth of each receiver, sampling or feedback aggregation mechanisms [5,19] can be used to avoid the implosion problem and minimize the collection time. We will discuss the feedback mechanism in detail in the next section.

V. Parameter Measurements and Discussions of Control Overhead

A. Calculation of Loss Event Rate

The update of loss event rate in HALM is done similarly to the method recommended in [13]. The only difference is that the loss event rate of a HALM receiver should be calculated across all the received layers because these layers act as a 'single' stream to compete for bandwidth with TCP connections. The difficulty here is that each layer has it own RTP sequence number space [9], and we cannot distinguish the order of packets from different layers by using sequence numbers only. To do this, we resort to some application-level semantics, such as using timestamps in conjunction with sequence numbers to distinguish the order.

B. Estimation of Round-Trip Time

Obtaining an accurate and stable measurement of the round-trip time is of primary importance for HALM. To find the 'true RTT', we must use a feedback loop which follows the definition of RTT. However, the use of feedback may cause implosion at the sender if there are many receivers sending estimation requests at a high frequency. On the other hand, low frequency requests may result in inaccurate conclusions. Motivated by the previous work on RTT estimation in multicast environments [11,12], we use a hybrid strategy to solve this problem, which combines a low frequency closed-loop method and a high frequency open-loop method. Our results show this strategy works well in most cases. Furthermore, it does not require synchronization between the sender and receivers' clocks.

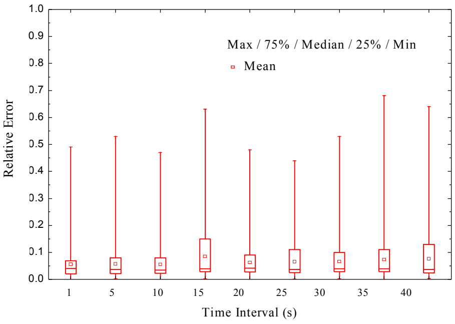


Figure 1. The relative errors of open-loop RTT estimation over different time intervals.

Closed-loop Estimation. The closed-loop method is based on the definition of RTT. As mentioned in Section III, a receiver report, RR, also serves as a request for closed-loop RTT estimation, and a sender report, SR, serves as a response. To reduce the overhead of packet headers, the sender does not give a response to each request but uses a batch process. Suppose the sender has sent an SR at time t and received K requests with identifiers $SSRC_i$ and arrival times t_i^{arrive}, i=1,2,…,K, in time slot [t,t+T_{SR}]. At time t+T_{SR}, it will multicast a new SR packet to all the receivers. The packet contains the list of $SSRC_i$ and corresponding delays t_i^{delay}, where t_i^{delay} =t+$T_{SR} - t_i^{arrive}$, i=1,2,…,K. When the receiver with $SSRC_i$ receives the response packet, it will generate a closed-loop RTT estimate τ^0 using $\tau^0 = t^0 - t' - t_i^{delay}$, where t^0 and t' are the current local time and the local time that the request was initiated, respectively.

We will show later that T_{RR} >> T_{SR}. If the receiver does not receive a response for its request after time T_{SR}+RTO, it will assume the response packet is lost and clear the record for that request. Therefore, the probability of a mismatch between a request and a response is very low. In addition, this kind of mismatch can be eliminated by a simple filtering mechanism, which filters the estimates with very large deviations, as shown in [12].

Open-loop Estimation. We also adopt an open-loop estimation method as a complement to the closed-loop estimation. The method tracks the one-way trip time from the sender to the receiver and transforms it to an estimate of the round-trip time. It is applicable to symmetric links as well as asymmetric links. We do not introduce extra control messages for this mechanism but use the existing sender report packets.

Note that an RTT estimate τ can be expressed as $\tau = \tau_{S\to R} + \tau_{R\to S}$, where $\tau_{S\to R}$ is the one-way trip time from the sender to the receiver, and $\tau_{R\to S}$ is the time from the receiver to the sender. Let $\tau_{R\to S} = \tau_{S\to R} + 2\delta$ where δ reflects the link asymmetry, we have $\tau = 2(\tau_{S\to R} + \delta)$. Suppose at local time t_R^0, the receiver updates its closed-loop RTT estimate with value τ^0, and the timestamp of the SR packet is t_S^0. At local time t_R' ($> t_R^0$), a new SR packet arrives with timestamp t_S'. If the receiver is not in the response list, it

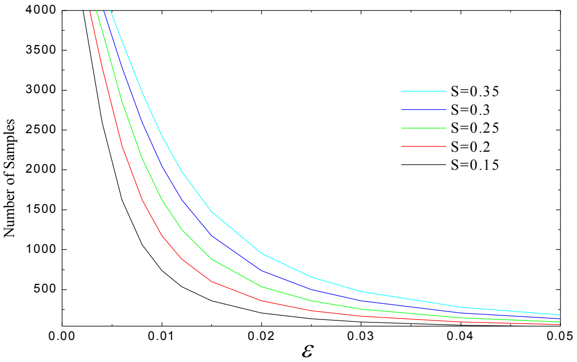


Figure 2. Number of samples vs. confidence interval.

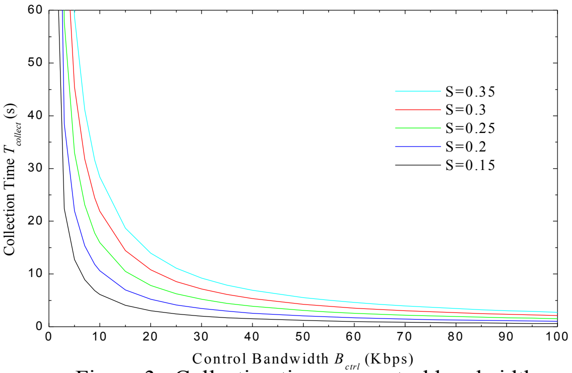


Figure 3. Collection time vs. control bandwidth.

will generate an open-loop RTT estimate τ' using the following relation,

$$\begin{aligned}\tau' &= 2(\tau'_{S\to R} + \delta') \\ &= 2[t'_R - (t^0_R - \tau^0_{S\to R} + t'_S - t^0_S) + \delta'] \\ &= 2[t'_R - (t^0_R - \frac{\tau^0}{2} + \delta^0 + t'_S - t^0_S) + \delta'] \\ &\approx 2[t'_R - (t^0_R - \frac{\tau^0}{2} + t'_S - t^0_S)].\end{aligned} \quad (6)$$

In (6), we make the assumption that $\delta' = \delta^0$. However, $\delta' - \delta^0$ may vary over time and it is also affected by the skew between the sender's clock and the receiver's clock. We have conducted a series of experiments over the Internet to examine the variations of $\delta' - \delta^0$. Figure 1 shows the results of normalizing $2|\delta' - \delta^0|$ by the corresponding RTT, that is, the relative errors, over different time intervals. The data are gathered from 10 connections from Hong Kong to United States and Europe where each connection lasts 1000 sec. It can be seen that the relative errors are usually less than 10%, and do not accumulate over time.

Finally, a smoothed round trip time is calculated by the weighted moving average method for TCP [15]. In our experiments, we find that, after smoothing, the maximal error is limited to about 15% by using T_{SR} = 1 sec, which is good enough for bandwidth estimation. Another parameter *RTO* can be estimated from *RTT*. Practically, the simple heuristic of *RTO*=max{1,4*RTT*} works reasonably well to provide fairness with TCP [13].

C. Control Overhead and Adaptation Frequency

Another key question of HALM is how frequently the sender should re-allocate the layer rates; that is, how to determine the control period, T_{ctrl}. Note that this parameter depends on the control bandwidth which is used to collect the receivers' feedback. For a fixed control bandwidth, the collection time scales linearly with the number of the receivers, as in RTCP [2]. Therefore, if the sender makes a decision based on the expected bandwidths of all the receivers' reports, the convergence time can be very long for large sessions. Since the optimal allocation algorithm depends only on the bandwidth distribution, we resort to sampling; that is, making decisions based on a controlled number of reports. The sample size, *n*, is the minimum number of reports that the sender requires to calculate the expected fairness index within a given confidence interval ε and a confidence level $1-\alpha$ [24]. This criterion is formally expressed as follows,

$$n = \min\{k : \Pr(|\overline{F}_k - \overline{F}| < \varepsilon) \geq 1-\alpha\}, \quad (7)$$

where $\overline{F}$ is the average fairness index based on the bandwidth distribution of all the receivers, and $\overline{F}_k$ is the one based on k reports. Since the receivers generate reports independently, we can assume that the samples are independent and identically distributed. From the statistical theory [24], we have

$$n_0 = \left(\frac{Z_{\alpha/2}S}{\varepsilon}\right)^2 \text{ and } n = \frac{n_0}{1+n_0/N}, \quad (8)$$

where $Z_{\alpha/2}$ is the upper $\alpha/2$ percentage point of the standard normal distribution and S is an estimate of the standard deviation of the fairness indices. This result holds for $n \geq 30$ regardless of the shape of the index distribution [24]. Given B_{ctrl}, the control bandwidth, W_{RR}, the payload size of a RR, W_{SR}, the size of each RTT response in a SR, H_{RR}, other overhead for a RR, and H_{SR}, other overhead for a SR, we have the following relation for $T_{collect}$, the collection time of *n* samples,

$$T_{collect} = \frac{n\cdot(W_{RR} + H_{RR} + W_{SR})}{B_{ctrl} - H_{SR}/T_{SR}}. \quad (9)$$

Here the receiver report period is $T_{collect}\cdot N/n$. The sender estimates it and then informs all the receivers. Each receiver uses an exponentially distributed feedback timer to avoid feedback implosion [25].

In HALM, we adopt the simplified 32-bit RTCP header for SR and RR packets, as described in [11], and 16 bits for bandwidth report and delay. Other overheads include the UDP and IP header (224 bits), SSRC field (32 bits), SR timestamp field (32 bits) and layer rates in SR (48 bits if 3 layers). Hence, we have H_{RR}=288 bits, H_{SR}=368 bits, W_{RR} = 16 bits and W_{SR} = 48 bits.

Figure 2 shows the required sample size for a confidence level of 95%, which provides a high enough confidence from a statistical point of view. The number of receivers is 5000 and the standard deviation varies from 0.15 to 0.35, which covers a broad dynamic range. Figure 3 shows the relationship between the control bandwidth and collection time for ε=0.02. It can be seen that the collection time is within 15 sec for a reasonable control bandwidth (about 20 Kbps). Note that, a short control period may result in inaccurate bandwidth estimation and highly oscillative adaptation behavior, which is not suitable for video transmission. It also increases the computation overhead. Hence, we set the control period to 15 sec in the current version.

D. Local Coordination

Usually the receivers in the same LAN have homogeneous parameters, such as RTT and loss event rate. We can use this homogeneous nature to speed up convergence and reduce the protocol overhead. First, when a receiver gets a closed-loop RTT estimate, it should distribute the new estimate to other receivers in the same LAN. Second, a newly joined receiver can initiate a request, and the receiver that updated the closed-loop estimate most recently in the LAN should respond to this request by providing the current RTT, loss event rate, and subscription level. This local coordination is very suitable for the scenarios where more than one receivers in the same LAN. It is worth noting that this scheme can be extended to the regions with TTL>1. We can use loss bit-map comparisons [16] to locate receivers with homogeneous parameters.

VI. PERFORMANCE EVALUATION

In this section, we examine the performance of HALM under a variety of configurations. We also compare it with layered multicast protocols with static allocation (LMSA). Two commonly used static layer rate allocation schemes are as follows:

Uniform allocation (LMSA-U). The rates of all enhancement layers are equal, i.e., $c_i = c_{i-1} + \beta$ for some constant β. An example is found in [11]. Note that MLDA [12] evenly allocates the layer rates between the minimum and maximum receiver bandwidths, which is analogous to the uniform allocation scheme except that MLDA changes allocation every control period.

Exponential allocation (LMSA-E). The cumulative layer rates are exponentially spaced by a constant factor $\gamma > 1$, i.e., $c_i = \gamma\, c_{i-1}$. This is the scheme adopted in the original RLM [1] and many other experiments [7,18].

A. Simulations Results

We have simulated HALM and LMSA protocols in a large number of topologies and configurations using the LBNL network simulator *ns*-2 [17]. Here we present a subset that explores their performance under some typical configurations that have been used in many previous studies [3,12].

We use the following default parameters for our simulations unless a new parameter is explicitly specified. The routing protocol is DVMRP. All the queues use FIFO drop-tail scheduling discipline with the maximum queuing delay of 0.15 sec. The link delay is set to 20 ms between two switches and 10 ms between a switch and an end system (a receiver or a sender). The TCP connections are modeled as FTP flows that always have data to send and last for the entire simulation time. A TCP-Reno flavor is used for simulating the congestion control behavior of TCP. To exclude the influence of the TCP congestion control window, we choose a max-window of 4000 packets which is sufficiently large to ensure TCP connections remain in the well-behaved mode. The packet size is 500 bytes for both TCP and HALM traffic.

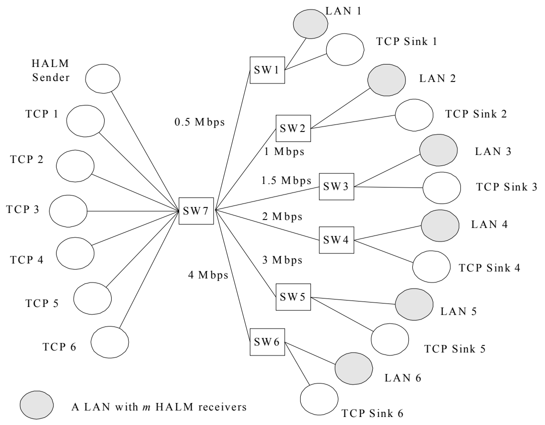


Figure 4. Simulation topology.

All simulations were run for 1000 seconds, which is long enough for observing transient ad steady-state behaviors. The cumulative layer rates of a HALM source are initialized to {256,512,1024 Kbps}, and the lower bound of the base layer rate is 220 Kbps. On the receiver's side, the initial settings are 100 ms for *RTT*, and 0 for *p*.

Figure 4 depicts one topology for our simulation. There is a HALM sender and 6*m* receivers belonging to 6 LANs, where each LAN has *m* receivers. The bottleneck links are (SW_7, SW_i), i=1,2,…,6, and each one is shared between the HALM flow and a TCP connection from TCP_i to $Sink_i$. Other links are sufficiently provisioned to ensure that any drops that occur are only due to congestion at the bottleneck links.

In the first simulation, we set *m* to 5. The receivers stay in the session throughout the whole period. We observe that the receivers in the same LAN receive data at identical rates when local coordination is adopted. Therefore, from each LAN, we choose one receiver as a representative and denote the receiver from LAN_i as receiver *i*. Figure 5 shows the cumulative layer rates allocated in the simulation, and Figure 7 shows the bandwidth distribution between the competing HALM and TCP flows at different switches. We can see that basically the rates of layer 1, 2, 3 adapt to the expected bandwidths of receiver 1, 3, 5. This adaptation setting is just the one that maximizes the expected fairness index. However, because of the inaccuracy in bandwidth estimations, we can see some oscillations of the layer rates, e.g., in Figure 7(b), receiver 2 oscillates between layer 1 and layer 2 and influences the rate allocation of layer 2. Since the rate allocation is based on the receivers' statistical behavior, this kind of transient changes is expected to be reduced in large sessions, as discussed in Section V.

We also simulate LMSA-U and LMSA-E on this topology by replacing the corresponding HALM sender and receivers. The bandwidth distributions between TCP and layered traffic are compared in Table 1. We observe that HALM generally increases the degree of TCP-friendliness and intra-session fairness than the two static allocation-based schemes. Although some receivers, such as LMSA-U receiver 6, get better bandwidth share when competing with TCP, the average fairness index of HALM is 0.83, which is higher than that of LMSA-U (0.67) and LMSA-E (0.69). Thus, as expected, HALM achieves higher performance in terms of the degree of the overall receiver satisfaction in a session.

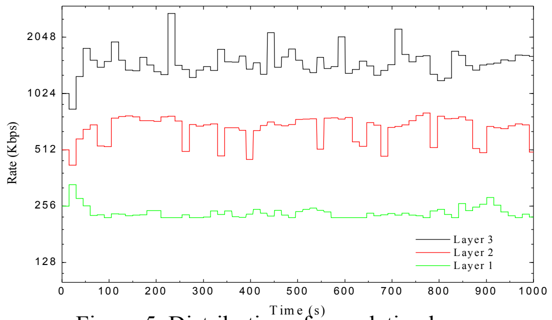


Figure 5. Distribution of cumulative layer rates without receiver joining and leaving.

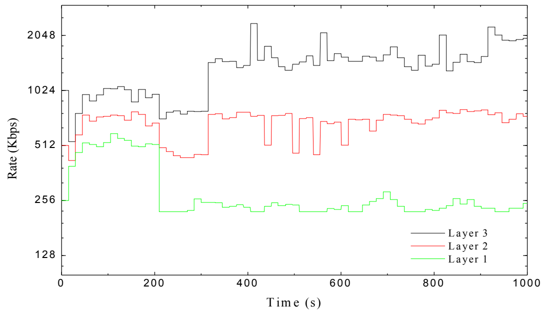
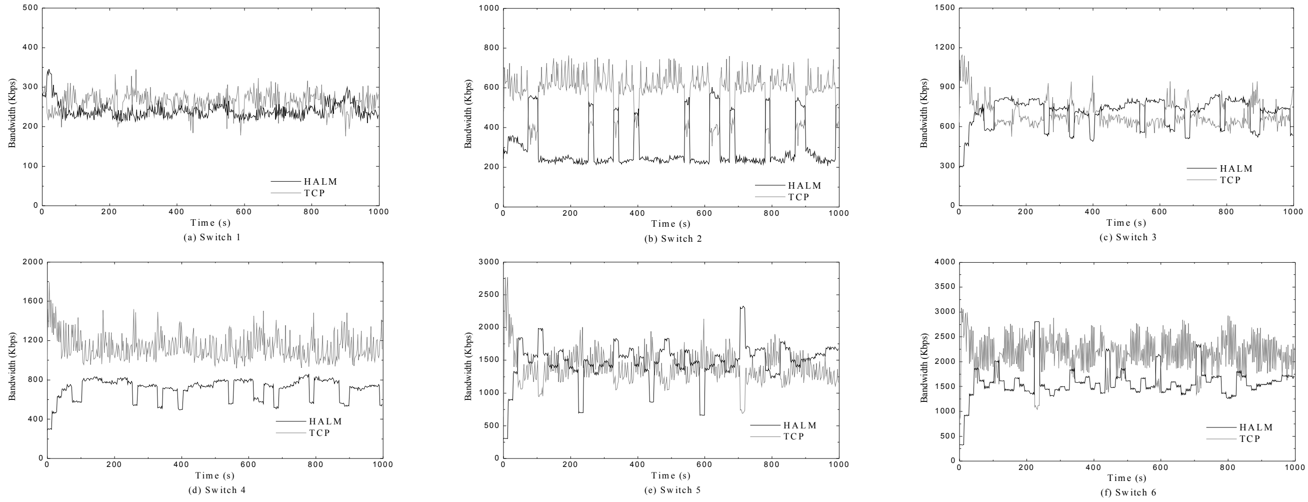


Figure 6. Distribution of cumulative layer rates with dynamic joining and leaving.



(a) Switch 1

(b) Switch 2

(c) Switch 3

(d) Switch 4

(e) Switch 5

(f) Switch 6

Figure 7. Bandwidth distribution between HALM and TCP at different switches.

Receiver	HALM			LMSA-U			LMSA-E		
	B_{HALM}	B_{TCP}	Ratio	B_{LMSA-U}	B_{TCP}	Ratio	B_{LMSA-E}	B_{TCP}	Ratio
1	227.6	258.1	0.88	193.4	277.5	0.69	246.3	226.7	1.09
2	331.8	602.5	0.55	219.3	721.5	0.30	337.4	607.2	0.56
3	704.3	696.2	1.01	388.7	1025.2	0.38	487.7	895.5	0.54
4	705.9	1136.2	0.62	573.7	1303.9	0.43	701.2	1164.6	0.60
5	1472.4	1389.5	1.05	1317.8	1475.8	0.89	1003.6	1789.4	0.56
6	1582.1	2169.3	0.72	1725.8	1992.9	0.87	1004.3	2757.8	0.36

Table 1. Distribution of the received bandwidths (Kbps). The ratio is obtained by dividing the layered stream bandwidth by the TCP bandwidth.

Receiver	1	2	3	4	5	6
Joining Time (s)	200	0	0	0	300	400
Leaving Time (s)	-	600	-	800	900	-

Table 2. Schedule of joining and leaving.

We also let the receivers dynamically join and leave the session to observe the effects of these actions and the responsiveness of HALM. To preclude the effect of local coordination, we set *m* to 1. In Table 2, we show a joining/leaving schedule in our simulation. The allocated source layer rates are shown in Figure 6. We can see that HALM always tries to maximize the overall system performance according to the current bandwidth distribution of the session members. Because the joining and leaving actions are synchronized, the convergence time of HALM is very short. Usually, the time for a receiver to get the optimal share is within one control period, or about 15 sec in this simulation. For example, after receiver 1 joins the session at 200 sec, it waits for the control signal for 10 sec. It then reports the expected bandwidth to the sender and the rate of the base layer is adjusted to about 200 Kbps, which is the requirement of receiver 1. The rates of layer 2 and 3 are also adjusted to maximize the overall system performance. We find that, after receiver 4 leaves the session at 800 sec, the layer rates do not change significantly because the original allocation still maximizes the expected fairness index under the new distribution. To the contrary, the leave of receiver 5 at 900 sec triggers a totally new allocation. This is because, originally layer 3 is adapted to the expectation of receiver 5, but now it can adapt to that of receiver 6 to achieve a higher fairness index.

B. *Statistical Results for Large Sessions*

For large sessions, we directly model the receiver bandwidths in a session coming from different distributions. To emulate the heterogeneous nature, a commonly used tool is the mixture Gaussian model [27], which consists of *k* clusters where each cluster follows a Gaussian distribution. In our study, the cluster means are chosen from 100 Kbps to 3

Order	Distribution	Parameter Settings			
		k	M_i (Kbps)	N	N_i
1	Clustered-1	3	200, 1000,2200	1000	333,333,334
2	Clustered-2	6	150, 500,900,1400,2000,2800	1000	166,166,167, 167,167,167
3	Top-heavy	5	150,550,1100,1750,2650	1000	150,150,150,400,150

Table 3. Receiver bandwidth distributions. k: number of clusters, M_i: mean of cluster i, N: total number of receivers, N_i: number of receivers in cluster i.

Mbps. This range covers the bandwidths of many available network access and video compression techniques. It is also a typical dynamic range of the existing layered coders, such as the MPEG-4 PFGS coder [29]. The standard deviation of a cluster is set to 10% of the cluster mean. Therefore, most bandwidth fluctuations are within ±10%, yet some are more than ±50%, which reflects the dynamic nature of the Internet traffic. We present the results of three representative mixture models, as listed in Table 3.

The relations between the expected fairness index and the number of layers are shown in Figure 8. We observe that all these layered multicast schemes can significantly improve the expected fairness index, as compared with single-rate (non-layered) multicast. One important question is how many layers should be used for a layered transmission system. From our results, it can be seen that the improved performance, when using more than 5 layers, is marginal. In addition, using a large number of layers increases the computational complexity on both sender and receiver's sides [28]. Hence, it is clear that 3 to 5 layers is a reasonable choice under a variety of session conditions.

The optimal allocation algorithm in HALM exhibits much better performance in terms of fairness and often outperforms the two static schemes by 10-20% or more. This is because the static schemes often set the layer rates naively, e.g., set to a point with few receivers. Furthermore, as shown in Figure 9, with the optimal allocation scheme, the variances of individual fairness indices are reduced as well because the optimal algorithm always tries to make the individual fairness indices close to 1, the maximum value.

An interesting phenomenon due to the non-adaptability of the static schemes is that, the expected fairness index does not monotonically increase with the increase of L. For example, in Figure 8(c) with the exponential allocation scheme, the fairness of L=5 is higher than that of L=6, and even of L=7, 8. This also gives a justification for the use of sender-adaptation as a complement to the static allocation based adaptation algorithms.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a TCP-friendly hybrid adaptation protocol for layered multicast. The protocol, known as HALM, performs adaptations on both the sender and receiver's sides to improve intra-session fairness and TCP-friendliness. Our main contribution is a formal study on the sender-based optimal layer rate allocation problem. We have defined the optimization metrics and objectives, and derived a scalable algorithm to solve it. We have also studied the mechanisms for parameter measurements in HALM. These mechanisms are carefully designed to exploit the potential of existing protocols so that the overall system performance is improved whereas the control overhead is kept at a low level.

The performance of HALM has been evaluated under a variety of configurations. We also compared it with traditional static allocation based protocols. Our results show that HALM interacts with TCP substantially better than

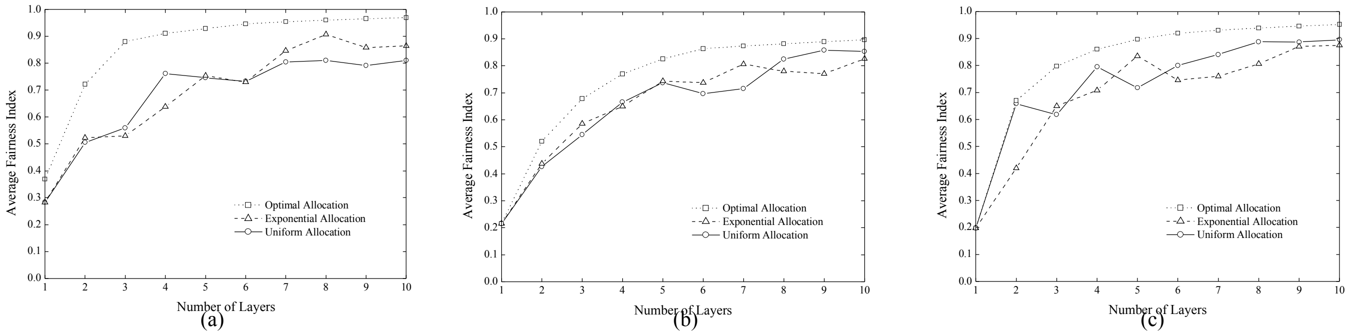


Figure 8. Average fairness indices of different allocation schemes. (a) Clustered-1; (b) Clustered-2; (c) Top-heavy.

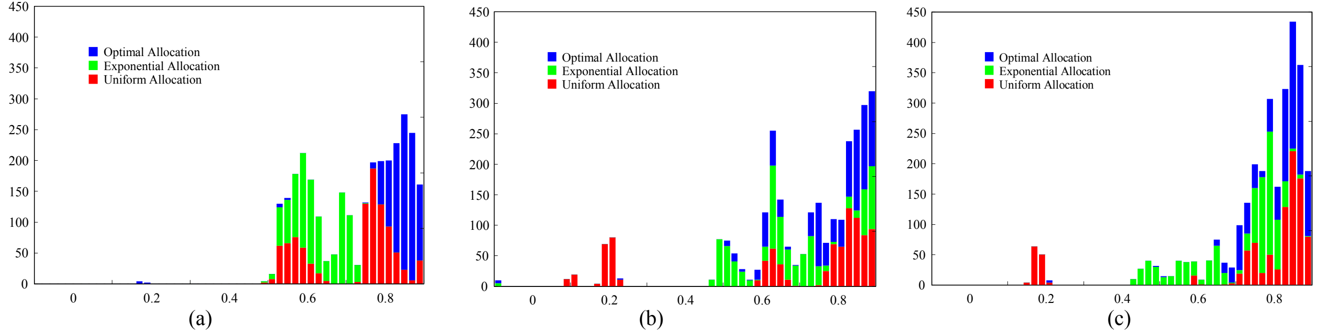


Figure 9. Distributions of fairness indices of different allocation schemes. X-axis: fairness index; Y-axis: number of receivers. Note that the figures are stacked histograms. (a) Clustered-1, L=4; (b) Clustered-2, L=4; (c) Top-heavy, L=4.

traditional protocols and outperforms them by 10-20% or more in terms of the expected fairness index. With the optimal allocation algorithm, increasing the number of layers always leads to better performance, and the use of 3 to 5 layers usually offers satisfactory degree of fairness. However, we found that this appealing property does not hold for the traditional static allocation schemes. Given these potential gains, we believe that the use of hybrid adaptation for layered multicast is worth consideration. However, issues such as its stability in a real network, implementation with real scalable coders, and comparisons with other advanced schemes [5,7,11,12,22,23], are not fully covered in this paper. Some results can be found in [35], and we will make further investigations in the future work.

ACKNOWLEDGMENTS

The authors would like to thank the researchers in Wireless and Networking Group, Microsoft Research, Asia (MSRA), for discussions on sender-adaptation, and the researchers in Internet Media Group, MSRA, for discussions on PFGS coding.

.

REFERENCES

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," in *Proceedings of ACM SIGCOMM 96*, pp.117-130, August 1996.

[2] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC 1889*, January 1996.

[3] X. Li, M. Ammar, and S. Paul, "Video Multicast over the Internet," *IEEE Network Magazine*, Vol. 13, No. 2, pp. 46-60, April 1999.

[4] J. Bolot, T. Turletti, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," *Computer Communication Review*, Vol. 24, No. 4, pp. 58-67, October 1994.

[5] B. Vickers, C. Albuquerque, and T. Suda, "Source-Adaptive Multi-Layered Multicast Algorithms for Real-Time Video Distribution," *IEEE/ACM Transaction on Networking*, Vol. 8, No. 6, pp. 720-733, December 2000.

[6] A. Legout and E. W. Biersack, "Beyond TCP-friendliness: A New Paradigm for End-to-end Congestion Control," *Technical Report*, Institute Eurecom, June 1999.

[7] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," in *Proceedings of IEEE INFOCOM 98*, pp. 996-1003, April 1998.

[8] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," in *Proceedings of IEEE INFOCOM 92*, pp. 2107-2114, May 1992.

[9] M. Speer and S. McCanne, "RTP Usage with Layered Multimedia Streams," *Internet Draft*, draft-speer-avt-layered-video-02.txt, December 1996.

[10] 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*, September 1998.

[11] T. Turletti, S. Parisis, and J. Bolot, "Experiments with a Layered Transmission Scheme over the Internet," *Technical Report*, INRIA, N'3296, November 1997.

[12] D. Sisalem and A. Wolisz, "MLDA: A TCP-friendly Congestion Contrl Framework for Heterogeneous Multicast Environments," in *Proceedings of the 8th Internatial Workshop on Quality of Service (IWQoS)*, June 2000.

[13] S. Floyd, M. Handley, J.Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," in *Proceedings of ACM SIGCOMM 2000*, pp. 43-57, August 2000.

[14] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A Model Based TCP-Friendly Rate Control Protocol, " in *Proceedings of NOSSDAV 99*, June 1999.

[15] W. Stevens, *TCP/IP Illustrated, Vol.* 1*: The Protocols*, 10th printing, Addison-Wesley, 1997.

[16] I. Kouvelas, V. Hardman, and J. Crowcroft, "Network Adaptive Continuous-Media Applications Though Self Organised Transcoding," in *Proceedings of NOSSDAV 98*, July 1998.

[17] S. McCanne and S. Floyd, *The LBNL Network Simulator, ns-2*, http://www.isi.edu/nsnam/ns/.

[18] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus Priority Dropping for Layered Video," in *Proceedings of ACM SIGCOMM 98*, pp. 131-143, September 1998.

[19] X. Xu, A. Myers, H. Zhang, and R. Yavatkar, "Resilient Multicast Support for Continuous-Media Applications," in *Proceedings of NOSSDAV 97,* May 1997.

[20] S. Deering, "Multicast Routing in a Datagram Internetwork," Ph.D. Thesis, Stanford University, 1991.

[21] K. Almeroth, "The Evolution of Multicast: From the MBone to Inter-domain Multicast to Internet2 Deployment," *IEEE Network, Special Issue on Multicasting*, Vol. 14, pp. 10-20, January/February 2000.

[22] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *Proceedings of ACM SIGCOMM 98*, September, 1998.

[23] S. Gorinsky, K. K. Ramakrishnan, and H. Vin "Addressing Heterogeneity and Scalability in Layered Multicast Congestion Control," *Technical Report TR2000-31*, The University of Texas at Austin, November 2000.

[24] D. Montgomery and G. Runger, *Applied Statistics and Probability for Engineers*, Wiley, New York, 1994.

[25] J. Nonnenmacher and E.W.Biersack, "Optimal Multicast Feedback," in *Proceedings of IEEE INFOCOM 98*, April 1998.

[26] D. Rubenstein, J. Kurose, and D. Towsley, "The Impact of Multicast Layering on Network Fairness," in *Proceedings of ACM SIGCOMM 99*, pp. 27-38, September 1999.

[27] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*, Wiley, New York, 1985.

[28] W. Li, "Fine Granularity Scalability in MPEG-4 for Streaming Video," in *Proceeding of ISCAS*, May 2000.

[29] S. Li, F. Wu and Y.-Q. Zhang, "Experimental Results with Progressive Fine Granularity Scalable (PFGS) Coding," *ISO/IEC JTC1/SC29/WG11*, *MPEG99/m5742,* March 2000.

[30] H.-M. Hang and J.-J. Chen, "Source Model for Transform Video Coder and its Application," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 2, April 1997.

[31] H. Wang and M. Schwartz, "Achieving Bounded Fairness for Multicast and TCP Traffic in the Internet," in *Proceedings of ACM SIGCOMM 98*, September 98.

[32] X. Li, S. Paul, and M. Ammar. "Multi-Session Rate Control for Layered Video Multicast," in *Proceedings of International Symposium on Multimedia Computing and Networking*, January 1999.

[33] Y.-T. Hou, D.-P. Wu, B. Li, W. Zhu, and Y.-Q. Zhang, "An End-to-End Approach for Optimal Mode Selection in Internet Streaming Video: Theory and Application," *IEEE Journal on Selected Areas in Communication*, Vol. 18, No. 6, June 2000.

[34] J. Liu, K.-M. Cheung, B. Li, and Y.-Q Zhang , "On the Optimal Rate Allocation for Layered Video Multicast," in *Proceedings of IEEE ICCCN 01*, October 2001.

[35] J. Liu, B. Li, and Y.-Q. Zhang, "An End-to-End Adaptation Protocol for Layered Video Multicast Using Optimal Rate Allocation," *Technical Report*, HKUST, May 2001.