# Network-Supported Layered Multicast Transport Control for Streaming Media

Zaichen Zhang, *Member*, *IEEE*, and Victor O.K. Li, *Fellow*, *IEEE*

**Abstract**—Multicast is very efficient in distributing a large volume of data to multiple receivers over the Internet. Layered multicast helps solve the heterogeneity problem in multicast delivery. Extensive work has been done in the area of layered multicast, for both congestion control and error control. In this paper, we focus on network-supported protocols for streaming media. Most of the existing work solves the congestion control and error control problems separately and does not give an integrated efficient solution. In this paper, after reviewing related work, we introduce our proposed protocols, namely, Router-Assisted Layered Multicast (RALM) and Router-Assisted Layered FEC (RALF). The former is a congestion control protocol, whereas the latter is an error control protocol. They work under the same framework and provide an integrated solution. We also extend RALM to RALM-II, which is compatible with Transmission Control Protocol (TCP) traffic. We analyze the complexity of the proposed protocols in the network and investigate their performance through simulations. We show that our solution achieves significant performance gains with reasonable additional complexity.

**Index Terms**—Internet, network protocols, multicast, network-supported protocols, streaming media.

## 1 Introduction

Multicast is an efficient approach to delivering streaming media to multiple receivers over the Internet. Multicast traffic is usually transmitted on top of the User Datagram Protocol (UDP) [1], which lacks congestion control mechanisms. This will lead to unfair usage of network resources vis-à-vis the Transmission Control Protocol (TCP) [2] and other adaptive traffic, congestion collapse [3], and network instability. Multicast transport control protocols, including congestion control and error control protocols, are necessary to enable wide deployment of multicast services on the Internet.

There are three major challenges in Internet multicast congestion control: scaling to a potentially large number of receivers, dealing with heterogeneity in the network and among receivers, and being compatible with other traffic, such as TCP. The Internet consists of heterogeneous networks. Both the bottleneck link capacity leading to receivers in a multicast group and the processing power of receivers can vary greatly. The transmission rate of a multicast group should satisfy faster receivers in the group while not overwhelming the slower ones. Layered multicast [4], [5] is an effective solution for the heterogeneity problem.

A survey of layered multicast can be found in [6]. We may divide the proposed protocols into end-to-end protocols and network-supported ones. The former, such as Receiver-driven Layered Multicast (RLM) [5], Receiver-driven Layered Congestion control (RLC) [7], and Fast-response Receiver-driven Layered Multicast (FRLM) [8], perform all congestion control functions at end hosts (senders and receivers). The network-supported protocols, such as packet Pair receiver-driven cumulative Layered Multicast (PLM) [9], priority dropping [10], Receiver-driven Layered Multicast with Priorities (RLMP) [11], and Active-Layered Multicast Adaptation (ALMA) [12], rely on additional network mechanisms to achieve enhanced performance. The trade-off is the complexity introduced into the network, which may degrade the overall performance and render a protocol impractical to deploy.

Multicast transport error control, which is commonly used for reliable data delivery applications, also helps provide performance enhancements and quality-of-service (QoS) guarantees for streaming media applications [13]. Although many streaming media applications are loss tolerant, excessive packet losses may lead to excessive performance degradation. Furthermore, in cumulative layered multicast congestion control schemes, lower layers are usually more important than higher ones. However, many protocols, such as RLM and RLC, implement uniform dropping, in which routers drop packets from different layers randomly when congestion occurs. Therefore, error protection is often desirable in such protocols, especially to provide error protection for lower layers.

We propose, as an integrated solution, Router-Assisted Layered Multicast (RALM) [14] and Router-Assisted Layered FEC (RALF) [15] for multicast streaming media. The former is a congestion control protocol, whereas the latter is an error control protocol. Both are network supported. In this paper, we will also extend RALM to RALM-II, which works well with TCP traffic at a common bottleneck link.

- *Z. Zhang is with the National Mobile Communications Research Laboratory, Department of Radio Engineering, Southeast University, Nanjing 210096, China. E-mail: zczhang@seu.edu.cn.*
- *V.O.K. Li is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China. E-mail: vli@eee.hku.hk.*

*Manuscript received 20 Mar. 2005; revised 2 Feb. 2006; accepted 19 Sept. 2006; published online 9 Jan. 2007.*
*Recommended for acceptance by C. Raghavendra.*
*For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0212-0305.*
*Digital Object Identifier no. 10.1109/TPDS.2007.1041.*

The rest of this paper is organized as follows: In Section 2, we describe related work. In Sections 3 and 4, we introduce RALM and RALF, respectively. RALM-II is presented in Section 5. We investigate the protocols by computer simulations using Network Simulator version II (NS2) [16]. Simulation results are given in Section 6. Complexity analysis of the proposed protocols is given in Section 7. We discuss some issues related to designing network-supported protocols and conclude the paper in Section 8.

## 2 Related Work

In this section, we introduce previous work in the area of layered multicast transport control for real-time applications. We first introduce representative congestion control protocols, including end-to-end and network-supported ones, and then introduce multicast error control schemes.

RLM is the fundamental layered multicast congestion control protocol. It includes a probing mechanism called "join-experiment," which makes join/leave decisions of multicast groups based on observed packet losses. Basically, a receiver subscribes to a higher layer (if any) when no packet loss is observed for a certain time. It drops a newly joined layer if packet loss occurs. Join-experiments may cause congestion in the network and interfere with other receivers. To scale to a large number of receivers, a "shared learning" mechanism is adopted in RLM. Before conducting a join-experiment, a receiver first multicasts an announcement of the experiment to the entire group. Other receivers will be aware of the experiment and share its result. RLM also maintains a join timer for each layer and a detection-time estimator. Join timers back off (double their values) on failed subscriptions and multiplicatively decrease their values in steady state. They are used in improving protocol stability and minimizing failed joins. The detection-time estimator estimates the latency between the start time of a local action and its feedback from the network. It is refreshed each time a join-experiment fails. Based on the above mechanisms, RLM receivers infer network capacities, subscribe to the proper number of layers, and gracefully adapt to network conditions.

End-to-end protocols are robust and easy to deploy on the Internet. However, they have difficulty in catching up with fast variations in the network and in coordinating receivers, leading to slow convergence and unfair bandwidth distribution among different sessions.

One of the network-supported schemes is priority dropping, in which routers drop packets from higher layers (which have lower priorities) when congestion occurs to protect lower layers. Priority dropping is stable. It adapts to congestion at the packet-level time scale. Moreover, priority dropping shares bandwidth fairly between multicast sessions as long as their layer bandwidth distributions are the same. However, implementing priority dropping in the network is complex. It requires checking the priority level of each incoming packet before it is queued and routed. It also requires a complex queue management scheme to keep priorities and retrieve randomly positioned packets in the queues.

RLMP tries to achieve the good performance of priority dropping while avoiding its high complexity. It uses only two priority levels: The highest subscribed layer has a low priority and other layers have a high priority. A receiver makes a join/leave decision (thus setting priority levels) based on the observed long-term packet loss ratios. At an outgoing link of a router, the priority for a multicast group is the maximum priority required by all downstream receivers/routers. In RLMP, the highest layer serves as a "buffer layer." It protects other layers by absorbing transient losses and monitors network congestion status. RLMP is stable even under bursty traffic and shares bandwidth fairly among competing multicast sessions. By using only two priority levels and first-in, first-out (FIFO) queues at routers, the complexity of RLMP is not excessive.

Our proposed RALM protocol is also a router-assisted layered multicast protocol. In contrast to priority dropping, there is almost no extra burden in the router's fast path. There is no need to check the priority level of each incoming packet. The only additional complexity is that, when looking up the Multicast Forwarding Table (MFT), the router also checks the suspension flag (a one-bit variable) in the MFT entry. This flag indicates whether the packet should be forwarded to the corresponding interface. This is also different from the RLMP approach, in which, for each incoming packet, a priority level (also a one-bit variable) associated with the group is retrieved from the MFT entry, and then a simple comparison is conducted based on the priority level and the current usage of outgoing queues to determine whether there is room in the queues for this packet.

Forward Error Correction (FEC) [17] is an efficient error control mechanism for multicast applications. Packet-level FEC, which deals with erasures instead of bit errors, is commonly used at the transport layer. An erasure is a lost packet with known packet number. The Reed-Solomon Erasure (RSE) [18] code is commonly used for packet-level FEC. An RSE code encodes a block of $k$ data packets into an $n$-packet codeword with $h = n - k$ redundant (parity) packets. Receiving any $k$ out of the $n$ packets in the codeword is enough for recovering the original $k$ data packets. The Tornado code [19] provides faster encoding and decoding time compared with RSE code for a large codeword. FEC is often combined with Automatic Repeat reQuest (ARQ) [20] in error control protocols. FEC helps in layered multicast for both reliable data delivery [21], [22] and real-time applications [23].

Layered FEC schemes [24], [25] are used to provide a certain level of error control on layered multicast congestion control protocols for real-time applications. A scalable and error-resilient compression code is usually adopted for source coding in a layered FEC. Here, "scalable" means that the coding scheme allows decoding at multiple rates. An error-resilient code has the ability to prevent error propagation from lost packets. Layered FEC works with pseudo-ARQ (ARQ is simulated by receivers subscribing and unsubscribing to FEC layers) [25] in an integrated way (a receiver requests more parity packets when necessary) [26]. After encoding the original data stream into several cumulative layers, an FEC code, such as an RSE code, is applied on the data layers and several FEC layers that contain the parity packets are generated. The data and FEC

layers are sent in separate multicast groups, with FEC layers delayed for certain times after the data layers. This delaying trades off between the level of protection and the receiving latency. It also helps alleviate burstiness in packet losses. A receiver tolerant to larger latency will subscribe to more FEC layers if the data layers are susceptible to errors.

Existing layered FEC schemes try to achieve optimal bandwidth allocation between data and FEC layers [27] so that user utility can be maximized with the given available bandwidth and packet loss ratio. However, optimal bandwidth allocation is determined by source coding and channel coding schemes and is also related to a utility function that maps the received service to user utilities. Finding such an optimal solution is still an open research problem and, generally, very complicated algorithms are necessary. Furthermore, given that the available bandwidth and packet loss ratio are both measured values and that there are variable time delays in joining/leaving multicast groups in the Internet environment, whether an optimal solution can achieve its desired performance in real implementation is quite questionable. Under these considerations, RALF does not attempt to achieve the overall optimal performance, which involves source coding and user utility issues. Instead, it provides a thin transport layer error control service that can be tailored for different upper layer schemes by adjusting a set of parameters.

Most proposed layered multicast error control solutions conduct error control together with congestion control, usually through an equation-based approach [28]. In this approach, a receiver estimates available bandwidth or its fair share of bandwidth by some equations using measured values like average packet loss ratios. The receiver then determines how many and which data and FEC layers should be subscribed. Existing algorithms often assume independent packet losses and maintain a long-term average loss ratio for control purposes. This average loss ratio is necessary in congestion control. It helps achieve protocol stability and the fair sharing of bandwidth among receivers (sessions) downstream of the same bottleneck link. However, packet losses actually occur in bursts, so the smoothed long-term average loss ratio is not suitable for determining proper FEC protection levels. We solve this problem by using RALM for congestion control and RALF for error control. RALM does not use the equation-based approach. Instead, receivers probe the available bandwidth through join-experiments. RALF uses instantaneous observed packet losses for error control. The protocols work in the same framework. Network mechanisms adopted by RALM are also useful for RALF, and RALF introduces no additional complexity in the network.

## 3 The RALM Protocol

In RALM, the sender encodes the original data stream into a fixed number of layers and sends each layer to a separate multicast group. In a session, the cumulative bandwidth from layer 1 (the basic layer) to layer $k-1 (k>1)$ is $B_d^k$, which we call the Lower End Bandwidth (LEB) of layer $k$. For the basic layer, $B_d^1$ is 0. The value of $B_d^k$ should be communicated to receivers joining the group carrying layer $k$. This could be achieved through out-of-band mechanisms. For example, LEB values of a multicast session could be announced together with multicast addresses of this session at a Web site, and receivers get this information from the Web site before they start the multicast session.

A basic idea of RALM is router-initiated suspension/retry. A RALM-aware router monitors the buffer status at each of its outgoing links using a two-threshold mechanism [14]. If congestion occurs at an outgoing link, the router will immediately suspend some of the current transmitting groups, that is, stop sending packets of the groups to that outgoing link temporarily. A suspended group will be retried when congestion disappears. Suspended groups that are not likely to successfully transmit later will be "dropped" by the router. The router then deletes all states related to the group. No further retry will be conducted for a dropped group unless it is subscribed by a downstream receiver again.

We set two thresholds, $P_h$ and $P_l$, of the buffer of each outgoing link, and $P_h > P_l$. They divide buffer usage into three states: High, Normal, and Low. We define two kinds of buffer state changes: "Low to High" and "High to Low." A "Low to High" occurs when the previous change is "High to Low" and the state changes to High again.[1] Similarly, a "High to Low" occurs when the previous change is "Low to High" and the state changes to Low again. We say that congestion occurs when a "Low to High" occurs and disappears when a "High to Low" occurs. In the former case, we suspend a group, and in the latter case, we retry a group. If the buffer state is still High (Low) after a group is suspended (retried), we suspend (retry) another group. The two-threshold scheme helps improve stability of the protocol.

Three timers are maintained at each outgoing link in RALM: a suspension timer $T_s$, a retry timer $T_r$, and a back off timer $T_u$. $T_s$ and $T_r$ define the minimal time intervals between two successive suspensions and two successive retries, respectively. These minimal time intervals are needed to observe the effects of suspensions or retries. $T_u$ is set to a minimal value $T_{u,\min}$ initially. Its value is doubled each time a failed retry is detected until a maximum value $T_{u,\max}$ is reached. A retry is called failed if it causes a "Low to High" event. When a RALM-aware router makes a decision of retrying a suspended group, it must wait for $T_u$ to do so. Therefore, when $T_u$ reaches $T_{u,\max}$, the protocol enters a steady state in which it retries a suspended group approximately every $T_{u,\max}$. When some events, for example, a new group joining at the outgoing interface or a burst of traffic arriving, break the steady state, $T_u$ is reset to $T_{u,\min}$. This breaking of steady state is detected by observing a "Low to High" event when a $T_u$ timer is running. Fig. 1 illustrates the basic operations of the suspension algorithm.

The choice of which group to suspend is based on group priorities. In the same multicast SESSION, groups carrying higher layers have lower priorities and will be suspended before those carrying lower layers. Priorities of groups from different sessions are determined by their LEB values. A group with a smaller LEB has higher priority. A RALM-aware router maintains a bandwidth list, as shown in

1. Initially, the state is Low and a "Low to High" occurs when the state changes to High.

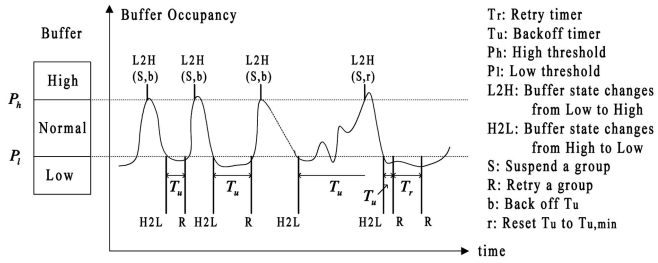


Fig. 1. Illustration of the suspension algorithm.

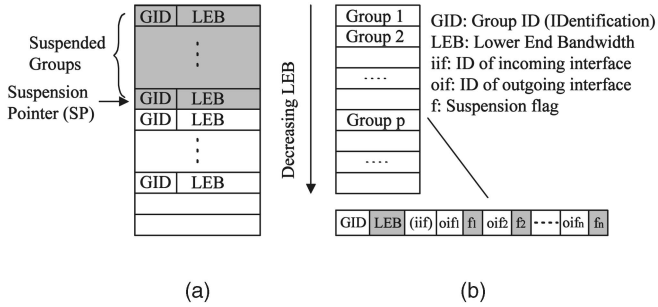


Fig. 2. The bandwidth list and MFT.

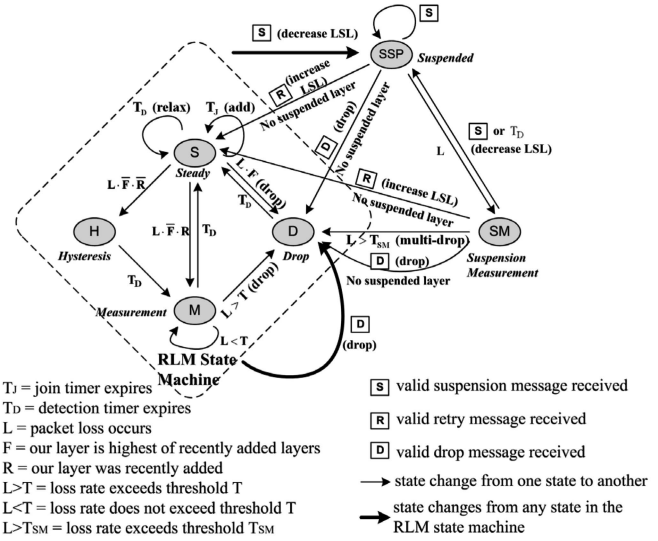


Fig. 3. The state machine of the RALM receiver protocol.

Fig. 2a, at each outgoing link. The list caches the IDentifications (IDs) and LEBs of groups that are likely to be suspended or retried. A RALM receiver puts a group's LEB value in an Internet Protocol (IP) option in the IP header of each join packet sent for this group. A RALM-aware router checks the LEB value and updates the bandwidth list if necessary. In the MFT at the router, there is a suspension flag associated with each outgoing interface in each group's entry, as shown in Fig. 2b. A set flag indicates that the group is now suspended at this outgoing interface. Packets will only be sent to outgoing interfaces with cleared suspension flags. When a group is suspended or retried at an outgoing interface, the corresponding suspension flag is set or cleared accordingly. Through this approach, group priorities are maintained in the control plane. Packet delivery is almost not affected—the only additional burden is checking the suspension flags.

When a router suspends, retries, or drops a group at an outgoing link, it will send through subcasting[2] a suspend packet $S_n$, retry packet $R_n$, or drop packet $D_n$ to all receivers in the group downstream to the link, where subscript $n$ is the index of the layer which is carried by the suspended, retried, or dropped group. RALM receivers perform all RLM operations for supporting incremental deployment. They also react to control (suspend, retry, and drop) packets from RALM-aware routers.

A receiver keeps a Lowest Suspended Layer (LSL) value, which indicates the lowest layer currently suspended. If no layer is suspended, LSL is set to $L_{\max}+1$, where $L_{\max}$ is the number of the highest layer currently subscribed by the receiver. Fig. 3 shows the state machine of the RALM receiver protocol. It includes the state machine of the RLM receiver protocol [5], which is shown in the dashed diamond. $S$, $H$, $M$, and $D$ denote four states: steady state, hysteresis state, measurement state, and drop state, respectively. Reasons for state changes, such as $T_J$, $T_D$, and $L$ (their meanings are explained in the figure), are labeled beside the arrows. A bar on a reason denotes when this reason is not true, and several reasons concatenated by dots means that these reasons occur simultaneously. Actions associated with a state change are indicated in parentheses. "Add" refers to subscribing to a higher layer, "relax" refers to multiplicatively decreasing the join timer for the current layer, and "drop" refers to dropping the current layer and backing off the join time for that layer. In addition to the RLM state machine, there are two RALM-specific states, namely, a "Suspended $(SSP)$" state and a "Suspension Measurement $(SM)$" state. When a valid[3] suspension packet $S_n$ is received at a receiver, the receiver sets its LSL to $n$ and changes into the $SSP$ state if it is not in this state. If a packet loss is observed at the $SSP$ state, the receiver changes to the $SM$ state and a $T_D$ timer (see [5]) is set. The $SM$ state is similar to the $M$ (Measurement) state in the RLM state machine, which logs losses for a longer time to see whether the loss rate exceeds a predefined threshold $T_{SM}$. If it does, the highest not suspended layer and all layers above it will be dropped (the "multidrop" operation in Fig. 3) and the receiver changes to the $D$ (Drop) state. Otherwise, after the $T_D$ timer expires, the state changes back to $SSP$. When a valid retry packet $R_n$ is received, LSL is increased to $n+1$. After this increase, if $LSL = L_{\max}+1$, which indicates that no layer is suspended, the receiver will change to the $S$ (Steady) state. When a valid drop packet $D_n$ is received, the receiver drops layer $n$ and all subscribed layers above it. After this, if no layer is suspended, the receiver will change to the $D$ state.

By coordinating receivers using network mechanisms, RALM achieves a fair share of bottleneck link bandwidth among multiple multicast sessions and avoids congestion caused by failed join-experiments. The protocol is stable since it keeps priorities in the network. It reacts fast to

2. Subcasting refers to multicasting in the subtree of a multicast distribution tree.

3. Considering loss or reordering of control packets, some of the received control packets are considered "invalid" and will be ignored by the receiver. Details about this can be found in [29].

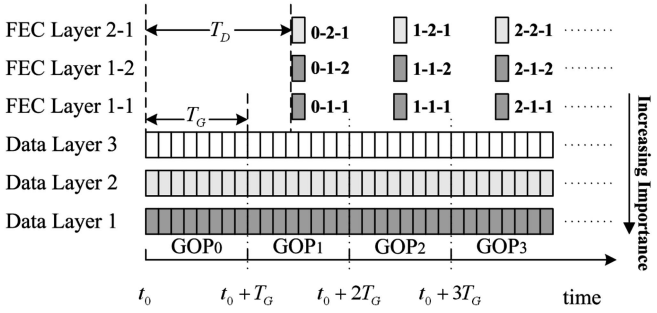


Fig. 4. Data organization in RALF.

network condition changes by monitoring network status at the routers.

## 4 The RALF Protocol

RALF is proposed to provide error protection on top of RALM. It is tailored for Internet streaming media applications. Its design embodies two principles: 1) decoupling multicast transport congestion control and error control from upper layers and 2) decoupling error control from congestion control at the transport layer, as explained in Section 2.

The data organization of RALF, as shown in Fig. 4, is similar to that of the layered FEC scheme proposed in [24]. Different from [24], where FEC layers are delayed with different time values, RALF sends all FEC layers together. We believe that this is preferred, since it simplifies the operation and avoids the possibility of receivers missing their deadlines due to excessive FEC layer delays.

In Fig. 4, there are three data layers. Each data layer has eight packets for each Group of Pictures (GOPs), which is a block of frames compressed together in the source encoder. Data layer 1 is the most important one and is protected by two FEC layers labeled 1-1 and 1-2. Data layer 2 is protected by one FEC layer labeled 2-1. There is no FEC protection for data layer 3. In this example, there is one parity packet in each FEC layer for each GOP, but more packets are also possible. In Fig. 4, index $i{-}j{-}k$ means that the parity packet is for GOP $i$ and in the $k$th FEC layer for data layer $j$. $T_G$ is the time duration of one GOP, which is determined by source coding. $T_D$ is the delay between data and FEC layers. If $T_D$ is too small, when a data layer requests its FEC layer, the FEC layer has already started. In this case, some useful FEC packets will not be received. On the other hand, the $T_D$ value cannot be too large to exceed the replay time required by the applications. For these reasons and considering delay jitter in the network, we choose the $T_D$ value as follows [29]:

$$\begin{cases} T_{D,\min} \le T_D < T_R - t_{0,\max}, & \text{if } T_{D,\min} < T_R - t_{0,\max} \\ T_D = T_R - t_{0,\max}, & \text{if } T_{D,\min} \ge T_R - t_{0,\max}, \end{cases} \quad (1)$$

where $T_{D,\min} = T_G + t_{d,\max} + t_{rtt,\max} + t_{0,\max} \cdot t_{d,\max}$ is the maximal value of a detection time—the time taken by a receiver to detect a packet loss. $t_{rtt,\max}$ is the maximal Round-Trip Times (RTTs) between the receivers and the sender (or an on-tree router). $T_R$ is the replay time, which is the maximal delay allowed by the source decoder. $t_{0,\max}$ is the maximal delay jitter in the network.

Among the above values, $T_G$ and $T_R$ are given by the employed source coding scheme, $t_D$ depends on packet loss detection mechanisms and traffic patterns, and $t_{rtt}$ and $t_0$ are related to network topology and congestion status. In a best effort network, there are generally no upper bounds for $t_D$, $t_{rtt}$, and $t_0$. However, when the sum of estimated $t_D$, $t_{rtt}$, and $t_0$ values is smaller than $T_G$ and $T_R$, $T_D$ can be chosen over a relatively wide range in which RALF's performance is not sensitive to the $T_D$ value. This condition holds for disseminating audio/video streams in a typical Internet environment. RALF is not efficient in an environment where $t_D$, $t_{rtt}$, and $t_0$ are large, such as in a satellite network or when the source coding requires very small $T_R$, as in interactive real-time applications.

As shown in Fig. 4, each data layer is protected by zero, one, or more FEC layers. In RALF, FEC layers for the same data layer have almost the same priority, which is lower than that of the data layer. When congestion occurs, this approach ensures that an FEC layer is suspended/dropped before its corresponding data layer. In the RALM framework, each layer in a layered multicast session has an LEB value, which reflects the layer's priority. A higher LEB value corresponds to a lower priority. The LEB value of the $m$th FEC layer for data layer $k$ is set to $B^k_{f,m} = B^k_d + \varepsilon_m$, where $B^k_d$ is the LEB of data layer $k$, $\varepsilon_m$ is a very small positive number, and $\varepsilon_{m1} > \varepsilon_{m2}$ if $m1 > m2$. As a result, we have $B^k_d < B^k_{f,m} < B^{k+1}_d$. In RALM, the bandwidth list maintenance algorithm [14] swaps groups with the same LEB values so that they can be served fairly. This is not necessary for FEC layers, since their bandwidths are much smaller than those of data layers. Therefore, we set slightly different $\varepsilon_m$s for different FEC layers to avoid such swapping and reduce the processing burden on the network.

RALF provides error protection in a greedy way, that is, it joins or keeps FEC layers as long as they may be needed. This greedy approach makes the protocol simple and robust, helps reduce fluctuations of joining/leaving FEC groups, and provides better error protection. The introduced redundancy is not excessive, since RALF uses thin FEC layers and packet losses are relatively rare when RALM is adopted for congestion control. A receiver maintains "holding timers" and a "loss counter" at each data layer for the greedy error protection. (In (1), setting $T_D$ to compensate for the maximum delay between lost data packets and the corresponding parity packets is also a greedy approach.)

Since FEC packets are delayed, when a receiver joins an FEC layer very soon after a packet loss, it should keep this layer until it receives the FEC packets for the GOP where the loss occurs. At the transport layer, RALF does not know which FEC packet is for which GOP, so we need to hold a joined FEC layer for a reasonable time $T_H$. In a greedy approach, $T_H$ should satisfy $T_H \ge T_D + t_{0,\max} - t_{l,\min}$, where $t_{l,\min}$ is the minimal time for leaving a multicast group. When an FEC layer is joined, a holding timer for this layer is set with value $T_H$. The receiver can only leave this layer after the timer expires.

The loss counter records the number of measured lost packets in one GOP. When $n_0$ packet losses are detected, the loss counter is increased by $n_0$. After $T_{G1} = T_G + t_{0,\max}$, it

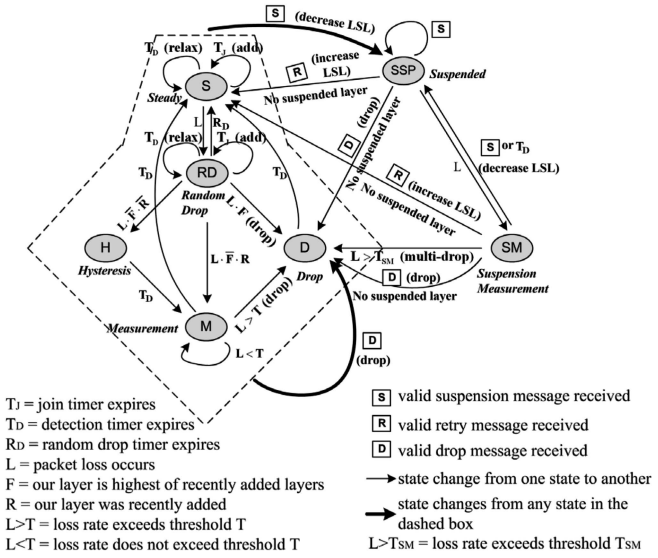


Fig. 5. The state machine of the RALM_II receiver protocol.

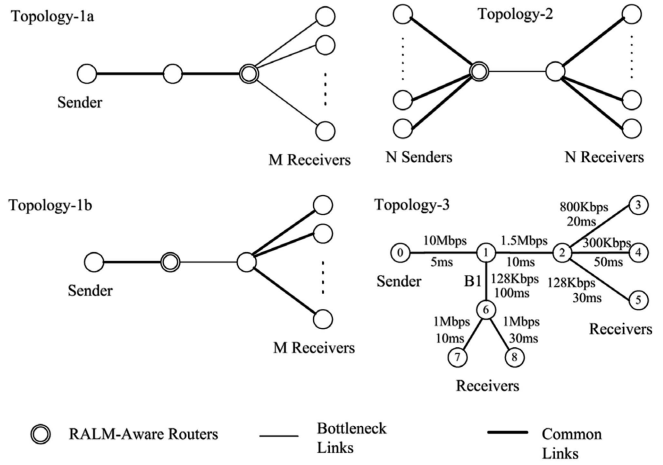


Fig. 6. Network topologies simulated.

will be automatically decreased by $n_0$. A receiver joins or leaves the FEC layers for a data layer based on holding timers and the loss counter. Supposing that an FEC layer contains one FEC packet for each GOP, $n_{FEC}$ is the number of joined FEC layers for the data layer, $n_0$ is the original value of the loss counter, and $n_1$ is the newly detected packet loss number, then the receiver acts as follows:

- If $n_0 + n_1 > n_{FEC}$, Joins $n_0 + n_1 - n_{FEC}$ FEC layers and Refreshes $n_{FEC} - n_0$ FEC layers.
- If $n_0 + n_1 \leq n_{FEC}$, Refreshes $n_1$ FEC layers.

Here, refreshing refers to resetting the holding timer with value $T_H$. When an FEC layer's holding timer expires, the receiver leaves this layer.

The above scheme is greedy. If $n_t$ losses occur within one $T_{G1}$, the loss counter will increase $n_t$ and $n_t$ FEC layers will be joined or refreshed. It is possible that the $n_t$ lost packets belong to two consecutive GOPs, in which case, less than $n_t$ FEC layers need to be joined or refreshed.

## 5 Extended RALM—RALM-II

In this section, we propose an extended version of RALM—RALM-II. RALM-II achieves a fair allocation of bottleneck link bandwidth, not only among different multicast sessions, but also between multicast and TCP sessions.

In a RALM-aware router, at each outgoing interface, two thresholds of the queue are set to monitor congestion status [14]. The gap between the two thresholds helps reduce traffic burstiness and improve protocol stability. However, if the gap is too large, the average queuing delay, which increases with the gap, will become intolerable for real-time applications. Too large a gap will also lead to slow convergence of the protocol. On the other hand, a small gap, which is usually adopted in RALM, has difficulty accommodating bursty traffic, leading to performance degradation when RALM shares the same bottleneck link with TCP.

The problem can be solved by adopting Random Early Detection (RED) [30] for queue management at RALM-aware routers. RED sets two thresholds, $P_{r,h}$ and $P_{r,l}$, for a queue, with $P_{r,h} > P_{r,l}$. When a packet arrives at the queue, an average queue size $avg$ is calculated. If $avg < P_{r,l}$, then the packet is enqueued. If $P_{r,l} < avg < P_{r,h}$, the packet is randomly marked with a possibility $P_a$, which increases with $avg$. Compared with unmarked packets, a marked packet has higher priority to be dropped when congestion occurs. If $avg \geq P_{r,h}$, then all incoming packets will be marked. RED provides congestion avoidance by detecting incipient congestions. By setting $P_{r,h}$ and $P_{r,l}$ to appropriate values, the average queue size (and, therefore, the average queuing delay) can be controlled and high-link utilization can still be maintained.

Since RED marks packets randomly when the average queue size falls between RED's two thresholds, to prevent receivers from overreacting to the loss of randomly marked packets, RALM-II adds an additional state, which is the "Random Drop ($RD$)" state, in the RALM receiver state machine, as shown in Fig. 5. If a packet loss is observed by a receiver in the $S$ state (the steady state), before changing to other states as in the original protocol, the receiver will first change to the $RD$ state and set an $R_D$ timer. If another packet loss is observed at the $RD$ state, the receiver will proceed to other states. Otherwise, if no packet loss is observed before the $R_D$ timer expires, the receiver changes back to the $S$ state.

## 6 Simulation Results

We simulated RALM based on the network topologies shown in Fig. 6. In topology 1, there is one RALM session with one sender and $M$ receivers. In topology 1a, each of the $M$ receivers is at a separate outgoing interface of the RALM-aware router. By using this topology, we test the scalability of RALM to the number of receivers, as well as the number of outgoing interfaces. In topology 1b, all the $M$ receivers are downstream to the same interface of the RALM-aware router. This topology is used to test the scalability of RALM to the number of receivers sharing the same outgoing interface. In topology 2, there are $N$ RALM sessions, each with one sender and one receiver. By increasing $N$, we show that RALM scales to a large number of competing sessions. In topology 3, we test RALM's performance in a heterogeneous environment with various link bandwidth

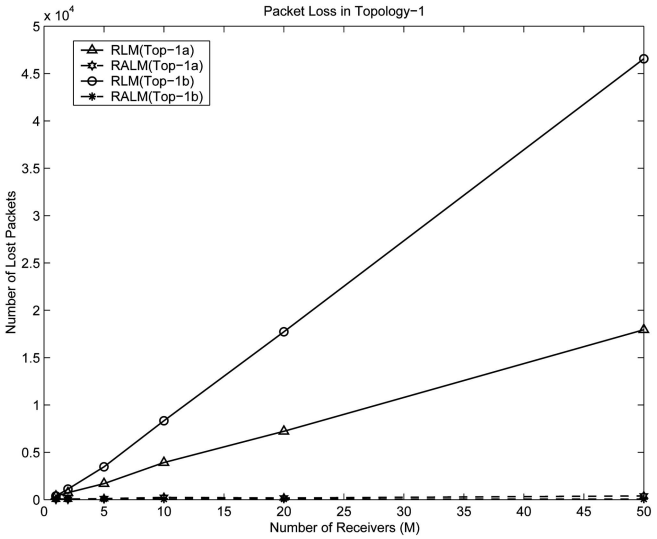


Fig. 7. Packet loss in topology 1.

and delays. In addition, the effect of RALM is illustrated by enabling RALM at node 1, at node 2, or at both nodes. For comparison, we also simulate RLM on the same topologies.

The bandwidth list on each outgoing interface contains 10 entries, no matter how many groups are active on that interface. Packet size is fixed at 1 Kbyte. Drop Tail is used for queue management. Common link bandwidth is 1 megabit per second and bottleneck link bandwidth is 500 kilobits per second times $N$, the number of sessions sharing the bottleneck link. If not specified, the delay of each link is 10 ms. Each queue's limit is 20 packets. For queues at outgoing interfaces of RALM-aware routers, their high threshold and low threshold are set as 15 and 5 packets, respectively. Constant Bit Rate (CBR) sources with variable coding delays [5] are simulated in each layer. The CBRs are $32 \times 2^{m-1}$ Kbps, $m = 1, \cdots, 6$, for layer $m$. Variable Bit Rate (VBR) sources with exponential distribution and Pareto distribution are also investigated. We run each simulation for 2,000 seconds of simulation time. If there is only one session, it is started at 1 second. If there are multiple sessions, they are started randomly on the interval [1, 5] seconds. Start times of receivers are randomly chosen on the interval [5, 60] seconds.

### 6.1 Basic Simulations on RALM

Fig. 7 plots the total number of lost packets versus session size in topology 1. We run each simulation 10 times with receivers joining at randomly chosen times and plot the average values. Although the total packet loss is proportional to the number of receivers in RLM, it remains constant in RALM. The majority of RALM losses are undelivered packets at RALM-aware routers. This kind of loss occurs when a packet arrives at a RALM-aware router and finds that all outgoing interfaces of its group are suspended. The router will then free the packet and send a leave message upstream. Therefore, the number of undelivered packets is proportional to the product of the bandwidth of the group and the delay of the incoming link to the router and will not increase with the number of receivers. Since undelivered packets are freed before being put into queues at bottleneck links, they will not waste bottleneck link bandwidth. Therefore, we conclude that RALM has a near-optimal loss property and scales very well to the number of receivers in a session. Furthermore, in our simulations, in topology 1, throughputs of RLM and RALM are almost the same, so that the desirable loss property of RALM is not achieved by sacrificing throughput.

TABLE 1
Fairness Indices of Simulations in Topology 2

| Number of Sessions | 2 | 5 | 10 | 20 |
|---|---|---|---|---|
| RLM | 0.9476 | 0.7500 | 0.7130 | 0.7218 |
| RALM | 1.0000 | 0.9999 | 0.9998 | 0.9997 |

TABLE 2
Throughputs and Packet Losses in Topology 3

| ID | RLM | RALM (1 enabled) | RALM (2 enabled) | RALM (1,2 enabled) |
|---|---|---|---|---|
| | Rcvd/Lost | Rcvd/Lost | Rcvd/Lost | Rcvd/Lost |
| 3 | 116960/93 | 118941/100 | 118468/0 | 118707/0 |
| 4 | 55751/256 | 55753/272 | 55701/0 | 55714/0 |
| 5 | 24386/288 | 24290/265 | 24293/0 | 24297/0 |
| 7 | 23972/370 | 24173/0 | 23919/326 | 24254/0 |
| 8 | 23946/360 | 24259/0 | 23984/334 | 24340/0 |
| Total | 245015/1369 | 247416/637 | 246365/660 | 247321/0 |

In topology 2, we focus on fairness between multiple multicast sessions. Here, we define the fair allocation of resources as the equal allocation of bandwidth among all sessions. Fairness is measured by adopting the fairness index [31] as

$$F(\vec{x}) = \Big(\sum_{i=1}^{n} x_i\Big)^2 \Big/ \Big(n \sum_{i=1}^{n} x_i^2\Big), \tag{2}$$

where $n$ is the number of sessions, $\vec{x} = (x_1, x_2, \cdots, x_n)$, and $x_i$, $i = 1, 2, \cdots, n$ is the total number of received packets of each session. $F(\vec{x}) = 1$ indicates that all sessions share bandwidth fairly, and $F(\vec{x}) = 1/n$ indicates that all resources are given to one session. Table 1 compares the fairness indices of RLM and RALM when different numbers of concurrent sessions are simulated. We run each simulation 10 times with receivers starting at random times and record the average values. From this table, we can see that RALM shares bandwidth fairly among different RALM sessions.

Table 2 records the observed number of received and lost packets of each receiver in topology 3, which consists of links with heterogeneous bandwidths and delays. When we enable RALM at node 1, receivers 7 and 8, which are downstream of the bottleneck link B1, observe no loss. Similarly, when node 2 is enabled, receivers 3, 4, and 5 observe no loss. When both nodes 1 and 2 are enabled, all receivers take advantage of RALM. The number of undelivered packets at RALM routers in the above three experiments are 0, 39, and 42, respectively.

By simulations, we studied RALM's sensitivity to settings of the parameters $P_H$ and $P_l$ under CBR traffic, where $P_H$ and $P_l$ are the high and low queue thresholds of the RALM protocol, respectively. Simulation results show that RALM is not sensitive to these parameters.

Fig. 8 plots the throughputs, RALM loss ratios (the NUMBER of RALM undelivered packets divided by the total number of packets sent), and fairness indices under

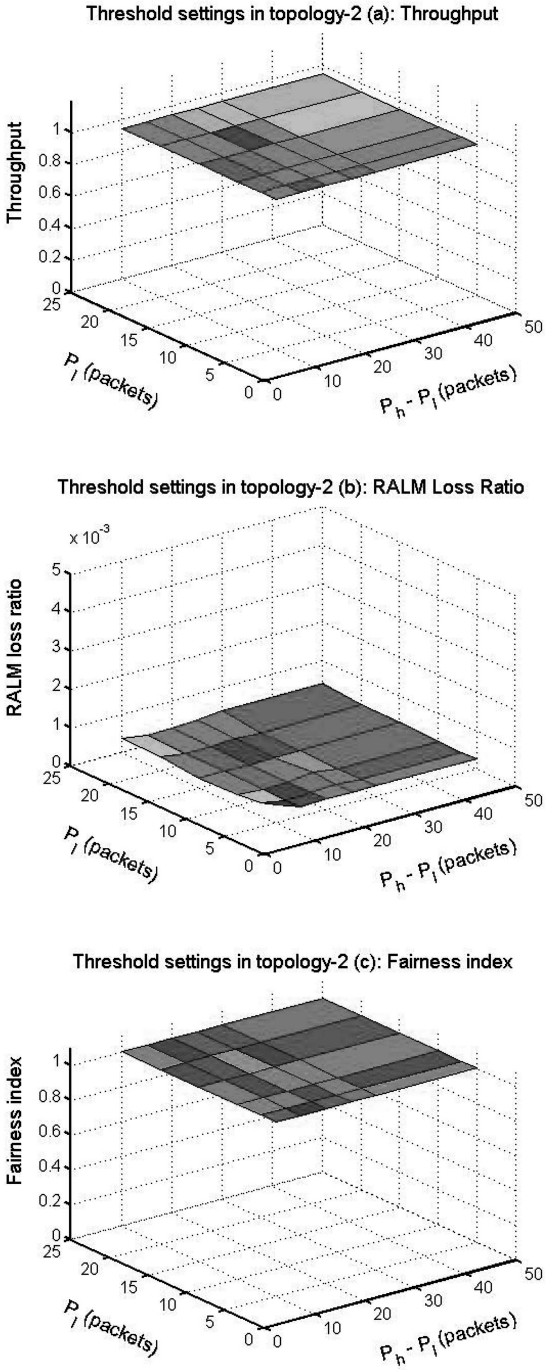


Fig. 8. Queue threshold settings in topology 2. (a) Throughput. (b) RALM loss ratio. (c) Fairness index.

different queue threshold settings in topology 2. The number of sessions is 10, and other values give similar results. In the range [10, 25] packets for the low threshold and [10, 50] packets for the gap, the obtained throughput, RALM loss ratio, and fairness index are almost the same.

### 6.2 Fairness between TCP and RALM/RALM-II Sessions

In this section, we investigated RALM and RALM-II's performance when they share a common bottleneck link with TCP traffic. Topology 2 in Fig. 6 is used in the simulations, where 10 RALM/RALM-II sessions and

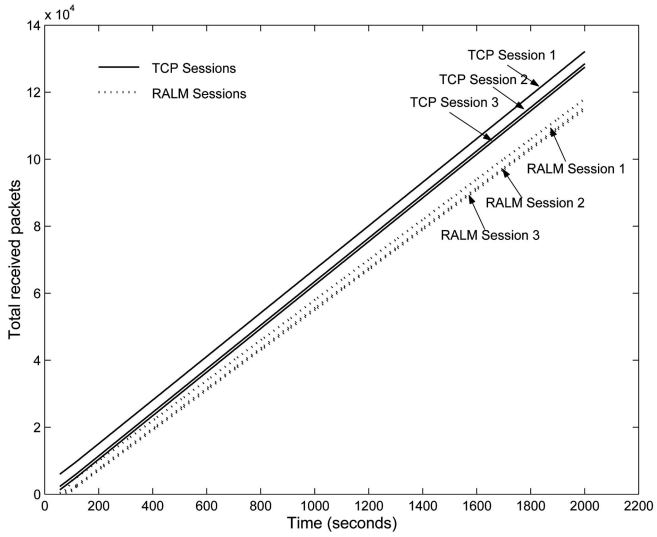


Fig. 9. Fairness between TCP and RALM sessions.

10 TCP sessions are simulated. For the simulation of RALM, the queue management scheme of the RALM-aware router is simply Drop Tail, and for RALM-II, RED is adopted. For RALM and RALM-II sessions, as in previous simulations, CBR sources with variable coding delays are simulated in each layer. The maximum window size at each TCP sender is 15 packets.

We first simulated RALM with TCP traffic. To accommodate TCP's bursty traffic, we set a large gap between the two thresholds of RALM—the high and low thresholds are set as 225 and 5 packets, respectively, and the queue size is set as 250 packets. We record the total number of received packets of each session every second and show the result in Fig. 9.[4] We can see that all sessions share the bottleneck link bandwidth fairly on a one-second time scale. In addition, the overall fairness index between TCP sessions and RALM sessions is 0.9979, which indicates that TCP and RALM sessions share bandwidth fairly. However, the recorded average queue occupation of the bottleneck link is 187 packets. With 10 Mbps bandwidth and 1 Kbyte packet size, this corresponds to an average delay of 150 ms, which is too large for some real-time applications.

We use RALM-II to solve this problem, and the simulation result is shown in Fig. 10. In this simulation, the two queue thresholds of RALM-II are 70 and 5 packets, respectively. RED is adopted with queue thresholds 50 and 5 packets, respectively. Although there is performance degradation compared with Fig. 9, the fairness index is still as high as 0.9962. However, the recorded average queue size is only 19.9, which corresponds to a 16 ms queuing delay, which is acceptable for most real-time applications.

### 6.3 Simulations with VBR Sources

We also investigated RALM performance with two types of VBR sources. In the first VBR source model, the packet interarrival time of each multicast layer follows the exponential distribution, with Probability Density Function (PDF)

4. In Fig. 9 and Fig. 10, in order to make the curves readable, we have only shown the results for Sessions 1, 2, and 3. The behavior of the other sessions is similar.

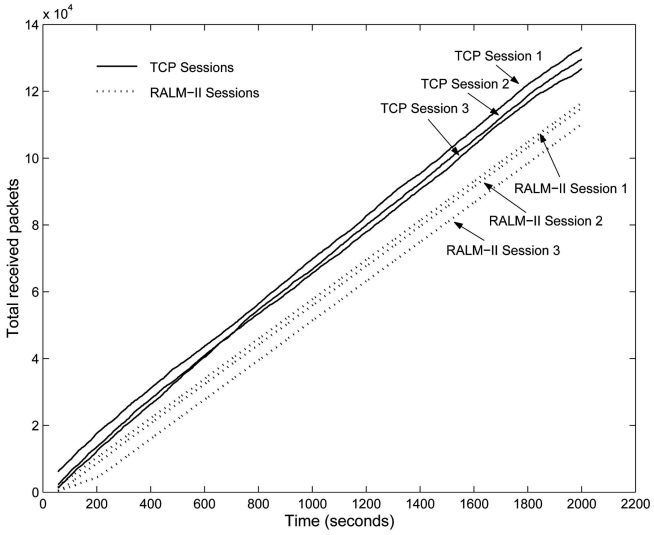


Fig. 10. Fairness between TCP and RALM-II sessions.

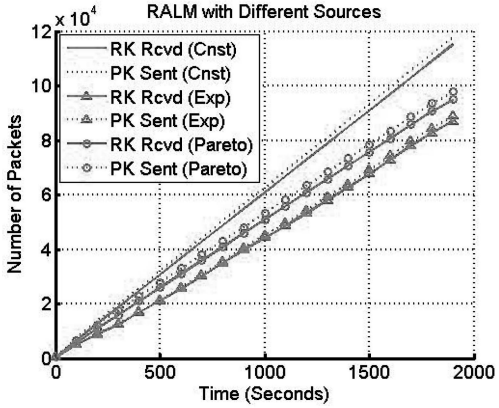


Fig. 11. RALM with different sources.

$f(x) = a \cdot e^{-ax},\ x > 0$, where the expectation $E(x) = 1/a$. We choose $a$ so that the average rate of each layer $m$ equals $32 \times 2^{m-1}$ Kbps, $m = 1, \cdots, 6$. In the second VBR source model, the packet interarrival time of each multicast layer follows the Pareto distribution, which is a heavy-tail distribution. The PDF is $f(x) = ca^c/x^{c+1}$, $a \leq x < \infty$. We set $c = 2.5$ so that the mean and variance of $x$ exist. $E(x) = ca/(c-1)$. We then choose $a$ so that the average rate of each layer $m$ equals $32 \times 2^{m-1}$ Kbps, $m = 1, \cdots, 6$.

Fig. 11 plots the number of packets sent at the sender and received by a receiver in topology 1a, with the CBR source and the two VBR sources, respectively. We run the simulation 10 times and show the average values in Fig. 11. In the case of VBR sources, the received number of packets is less than that of the CBR case, since the VBR sources introduce high burstiness. However, in all cases, the protocol works well. More importantly, the differences between packets sent and packets received (that is, the number of lost packets) in all cases are almost the same,[5] which means that the VBR sources waste no extra network resources.

5. When a burst of packets arrive at a RALM-aware router, the router suspends congested outgoing interfaces, and if all outgoing interfaces are suspended, the router will notify the upstream node to stop sending so that bursty traffic will not introduce extra packet loss.

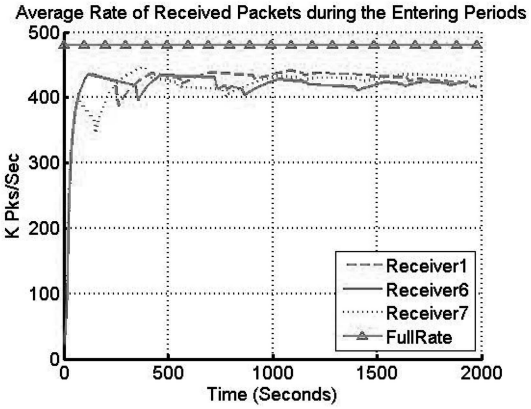


Fig. 12. RALM performance with dynamic enter/quit receivers (average rate of receivers during the entering periods).

### 6.4 Simulations with Dynamic Receiver Joins/Leaves

We evaluate RALM performance when group members dynamically join and leave frequently using topology 1a. To distinguish this from joining/leaving a layer, we say a receiver "enters" when it starts to receive service (begins to join the basic layer) and "quits" when it stops receiving service (leaves all layers). There are 10 receivers. When a receiver enters a session, it schedules a timer with a value randomly chosen in the range [10, 300] seconds. When the timer expires, it quits the session and schedules another timer in the same way. When the timer expires, it enters the session again.

Fig. 12 records the average rate of the received packets during the entering periods (the number of received packets divided by the sum of time of entering periods). To make the figure clear, we plot the performance of three randomly chosen receivers from the 10 receivers, noting that similar performance is observed for the other receivers. We also plot in the figure the "full rate," which is the optimal rate a receiver can achieve. With the simulation setup, a receiver can join four layers at most, with bandwidths of 32, 64, 128, and 256 Kbps, respectively. As a result, the full rate is 480 Kbps. From the figure, we find that all the receivers can approach the full rate. Since the minimal join interval is 5 seconds, a receiver needs at least 15 seconds to achieve the full rate after it enters the session, at which period, the average rate is low. This explains the gap between the full rate and the receiver performance in the figure.

To make the enter/quit procedure clear, we plot in Fig. 13 the average rate of each enter/quit period of receiver 1 with its enter/leave status. At the lower figure, the dashed line changing from 0 to 1 indicates that the receiver enters the session, and the changing from 1 to 0 indicates that the receiver quits the session. We find that, when the receiver enters the session, the average rate begins to increase to the full rate, and after it quits the session, the average rate is 0. In the case that the entering period is short, the average rate cannot approach the full rate due to the necessary minimal join periods.

### 6.5 Simulations on RALF

We simulated RALF using topology 2 in Fig. 6, where there are 10 RALF sessions and 10 TCP sessions. The bottleneck

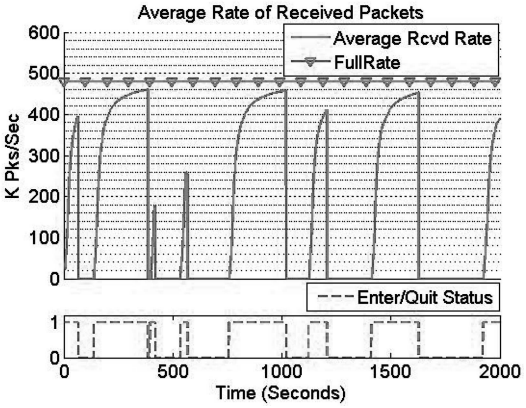


Fig. 13. RALM performance with dynamic enter/quit receivers (average rate of receiver 1).

link bandwidth is 10 Mbps. The propagation delay of each link is 20 ms. Each simulation runs for 1,000 simulated seconds.

Source data are encoded into 10 data layers with a bandwidth of 80 Kbps for each layer. The GOP duration is $T_G = 1$ sec, and the packet (payload) size is 1 Kbyte. Therefore, a data layer has 10 packets in each GOP. The 10 data packets are encoded using an RSE code, and three FEC (parity) packets are generated for them. The three FEC packets are divided into three FEC layers with one packet in each layer for each GOP. With this setting, the bandwidth of each FEC layer is 8 Kbps and the maximum number of FEC layers for each data layer is three.

Fig. 14 illustrates the effect of FEC protection on the basic layer at one of the receivers. Detailed numerical results for all subscribed layers are given in Table 3.

The top figure in Fig. 14 plots the received packets in a RALM session, where RALF is not enabled. The number of received packets in each GOP is plotted. Since one GOP contains 10 data packets, a value less than 10 in the figure indicates that one or more packets are lost in the corresponding GOP. The next figure plots received packets when RALF is used for error protection. From these two figures and Table 3, we see that RALF enjoys a little lower packet loss probability than RALM. This is because, in RALF, there are many thin FEC layers which adapt to congestion better than the data layers. The third figure in Fig. 14 plots the received

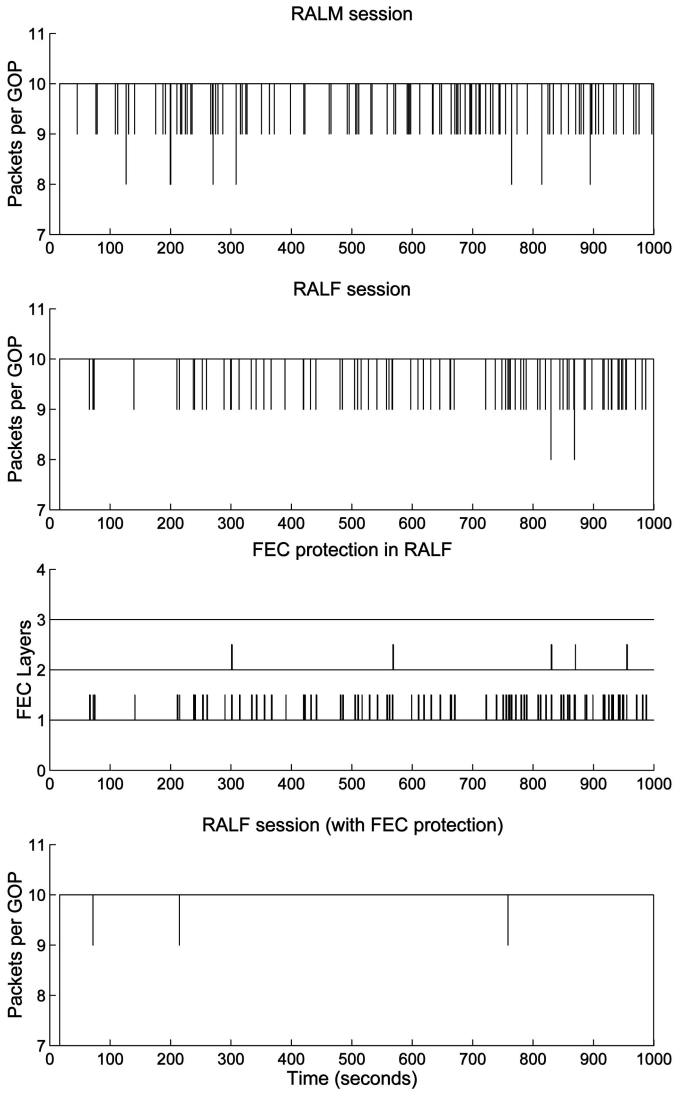


Fig. 14. FEC protection in RALF.

TABLE 3
Error Protection in RALF

| | Layer | Data Rcvd | Lost Data | FEC Rcvd | Redund-ant FEC | Unrecover-able Loss |
|---|---|---|---|---|---|---|
| RALM | 1 | 9748 | 112 | | | |
| | 2 | 9612 | 96 | | | |
| | 3 | 9445 | 120 | | | |
| | 4 | 9202 | 118 | | | |
| | 5 | 9048 | 117 | | | |
| | 6 | 8923 | 143 | | | |
| | 7 | 8777 | 134 | | | |
| | Total | 64791 | 840 | | | |
| RALF | 1 | 9778 | 82 | 149 | 70 | 3 |
| | 2 | 9633 | 75 | 138 | 65 | 2 |
| | 3 | 9466 | 91 | 170 | 79 | 0 |
| | 4 | 9394 | 70 | 131 | 61 | 0 |
| | 5 | 8800 | 119 | 185 | 81 | 15 |
| | 6 | 8420 | 122 | 208 | 97 | 11 |
| | 7 | 8085 | 108 | 187 | 81 | 2 |
| | Total | 63576 | 667 | 1168 | 534 | 33 |

FEC packets from each FEC layer for the data layer. We see that the first FEC layer is adequate most of the time, and the second layer is occasionally joined when data packets are lost in burst. In this simulation, the third FEC layer has never been subscribed by this receiver for its basic data layer. The bottom figure plots the received packets after error recovery using the RSE code. The result is near optimal: Only three packets are lost during the 1,000 seconds simulation time. For other active (subscribed and not suspended) data layers, the loss patterns and error protection effects are similar, as reflected in Table 3.

Table 3 also records the total number of redundant FEC packets for each data layer. When the sum of the received data packets and FEC packets for one GOP exceeds the total data packet number in a GOP (10 in our simulation), their difference is defined as the number of redundant FEC packets. Due to RALF's greedy error protection, nearly half of the FEC packets are redundant, as shown in Table 3. However, since the packet loss ratio is relatively small in RALM/RALF (compared with other probing-based layered multicast protocols, such as RLM), the redundant FEC packets are not excessive. Taking the first row of RALF in Table 3 as an example, there are only 70 redundant FEC packets with the 9,778 received data packets.

## 7 Complexity Analysis

The complexity of the proposed protocols include two parts: complexity in the network and complexity at the end hosts. Since end hosts usually have high computing and storage capabilities, which are enough for the protocols' operations, in this section, we focus on the complexity in the network.

Additional complexity in the network due to RALM includes: the signaling process, processing burden for each incoming packet, and state information maintained in routers.

RALM's signaling burden is small. The control (suspend, retry, and drop) packets are only sent sparingly in steady state. For example, in the simulation of Fig. 10, where there are 10 TCP and 10 RALM sessions sharing the same bottleneck link, only 18 RALM control messages from the RALM-aware router are observed during the 2,000-seconds simulation time. Adding an IP option for carrying LEB information in each join message and processing the header at RALM-aware routers are also reasonable.

Although RALM suspends or drops groups from the lowest priority, in contrast to priority dropping, there is almost no extra burden in the router's fast path. There is no need to check priority level in each incoming packet. The only additional complexity is that, when looking up the MFT, the router also checks the suspension flag (a one-bit variable) in the MFT entry. This flag indicates whether the packet should be forwarded to the corresponding interface. This is also different from the RLMP approach in which, for each incoming packet, a priority level (also a one-bit variable) associated with the group is retrieved from the MFT entry, and then a simple comparison is conducted based on the priority level and the current usage of outgoing queues to determine whether there is room in the queues for this packet.

RALM maintains additional states in routers, including suspension flags and LEB values in MFT entries and a bandwidth list, three timers, and two thresholds at each outgoing interface. They are the major burden RALM introduced to the network. Among these, the suspension flags and LEB values only consist of a very small fraction of MFT. However, sorting MFT based on the LEB value may require an excessively long time when the multicast group number is large. The bandwidth list is a small cache with several entries, and each entry includes only a group ID and an LEB value. Therefore, storage and manipulation of the list do not incur too much burden.

RALM also shows simplicity in the following aspects: Besides information in the bandwidth list, a RALM-aware router does not need to maintain other priority information associated with each group and does not need to know which groups consist of a multicast session. RALM uses a single FIFO queue at each outgoing interface and can still share bandwidth fairly with TCP sessions.

As stated above, RALF works in the same framework as RALM and requires no additional complexity in the network. RALM-II assumes RED as the queue management scheme at RALM-aware routers, whereas RALM assumes Drop Tail.

We conclude that, although the proposed protocols require some extra complexity in the network, the complexity is not excessive and is justified by the advantages of running these protocols, as shown in the simulation results in Section 6. Finding ways to further reduce this complexity is a major direction of our future work.

## 8 Conclusions

In this paper, we first review related work on layered multicast congestion control and error control for real-time applications, focusing on network-supported schemes. Then, we introduce our proposed protocols—RALM and RALF—and extended RALM to RALM-II, which is compatible with TCP traffic. Our proposed protocols enable

delivering, through layered multicast, a large volume of streaming media to a large number of receivers over the Internet efficiently.

The proposed protocols employ network mechanisms for enhanced performance. We believe that this approach is promising in the design of Internet multicast protocols. Today's Internet is a worldwide public information infrastructure, and complete end user cooperation cannot be guaranteed. Many difficulties with Internet multicast deployment, such as those associated with transport control, counting, and billing, stem from the fact that the Internet operates in an "open" and "best effort" mode. With the fast development of router technologies, network-supported multicast protocols will be more widely available and allow multicast to be fully deployed in the Internet in the near future.

However, network mechanisms should only be used when necessary. The added complexity in the network should be compensated for by enhanced performance. Standard ways of evaluating the complexity and performance gains are still lacking. This is an open problem for future research.

Implementation issues should also be considered when designing a network-supported protocol. For example, priority dropping needs complex queue management schemes, which require expensive hardware in implementation and degrade packet delivery speed. Therefore, although this scheme enjoys attractive performance gains, it is not practical. The ability to support incremental deployment is also desired.

## ACKNOWLEDGMENTS

This research is supported in part by the Research Grants Council, Hong Kong Special Administrative Region, China, Grant HKU 7044/02E, and by the Natural Science Foundation of Jiangsu Province, Jiangsu, China, Grant BK2005409. Preliminary versions of parts of this paper have been published in the *Proceedings of the IEEE International Conference on Communications 2002* and the *Proceedings of the IEEE Global Telecommunications Conference 2002*.

## REFERENCES

[1] J. Postel, *User Datagram Protocol,* IETF RFC 768, Aug. 1980.
[2] J. Postel, *Transmission Control Protocol,* IETF RFC 793, Sept. 1981.
[3] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM Special Interest Group Data Comm. (SIGCOMM '88),* pp. 158-173, Aug. 1988.
[4] S. Deering, "Internet Multicast Routing: State of the Art and Open Research Issues," *Proc. Multimedia Integrated Conf. for Europe (MICE '93) Seminar at the Swedish Inst. of Computer Science,* Oct. 1993.
[5] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. ACM Special Interest Group Data Comm. (SIGCOMM '96),* pp. 117-130, Aug. 1996.
[6] V.O.K. Li and Z. Zhang, "Internet Multicast Routing and Transport Control Protocols," *Proc. IEEE,* vol. 90, pp. 360-391, Mar. 2002.
[7] L. Vicisano and J. Crowcroft, "TCP-Like Congestion Control for Layered Multicast Data Transfer," *Proc. INFOCOM,* pp. 996-1003, Mar./Apr. 1998.
[8] H. Chiu and K.L. Yeung, "Fast-Response Receiver-Driven Layered Multicast," *Proc. IEEE Symp. Computer and Comm. (ISCC '04),* pp. 1032-1037, June/July 2004.
[9] A. Legout and E.W. Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes," *Proc. ACM SIGMETRICS,* pp. 13-22, June 2000.
[10] S. Bajaj, L. Breslau, and S. Shenker, "Uniform Versus Priority Dropping for Layered Video," *Proc. ACM SIGCOMM '98,* pp. 131-143, Aug./Sept. 1998.
[11] R. Gopalakrishna, J. Griffioen, G. Hjalmtysson, C.J. Sreenan, and W. Su, "A Simple Loss Differentiation Approach to Layered Multicast," *Proc. INFOCOM,* pp. 461-469, Mar. 2000.
[12] R.F. Sari, "Performance Evaluation of Active Network-Based Layered Multicast Congestion Control Protocol," *Proc. Sixth Int'l Conf. Advanced Comm. Technology,* pp. 555-560, 2004.
[13] L. Cheng and M.R. Ito, "Receiver-Driven Layered Multicast Using Active Networks," *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME '03),* pp. 501-504, July 2003.
[14] Z. Zhang and V.O.K. Li, "Router-Assisted Layered Multicast," *Proc. IEEE Int'l Conf. Comm. (ICC '02),* pp. 1657-1661, Apr./May 2002.
[15] Z. Zhang and V.O.K. Li, "Layered Multicast with Forward Error Correction (FEC) for Internet Video," *Proc. IEEE Global Telecomm. Conf.,* pp. 1465-1469, Nov. 2002.
[16] The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns, 2005.
[17] S. Lin and D.J. Costello, *Error Control Coding: Fundamentals and Applications.* Prentice Hall, 1983.
[18] A.J. McAuley, "Reliable Broadband Communication Using a Burst Erasure Correcting Code," *Proc. ACM SIGCOMM '90,* pp. 297-306, Sept. 1990.
[19] J.W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *Proc. ACM SIGCOMM '98,* pp. 56-67, 1998.
[20] A.S. Tanenbaum, *Computer Network,* third ed. Prentice Hall, 1996.
[21] L. Vicisano and J. Crowcroft, "One to Many Reliable Bulk-Data Transfer in the Mbone," *Proc. Third Int'l Workshop High Performance Protocol Architectures (HIPPARCH '97),* June 1997.
[22] L. Vicisano, "Notes on a Cumulative Layered Organisation of Data Packets Across Multiple Streams with Different Rates," Computer Science Research Note RN/98/25, Univ. College London, 1998.
[23] X. Zheng, S.-H. Chan, Z. Qian, W. Zhu, and Y. Zhang, "Feedback-Free Packet Loss Recovery for Video Multicast," *Proc. Int'l Conf. Comm. (ICC '03),* pp. 870-874, May 2003.
[24] W.-T. Tan and A. Zakhor, "Video Multicast Using Layered FEC and Scalable Compression," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 11, no. 3, pp. 373-386, Mar. 2001.
[25] P.A. Chou, "FEC and Pseudo-ARQ for Receiver-Driven Layered Multicast of Audio and Video," Technical Report MSR-TR-99-86, Microsoft Research, Nov. 1999.
[26] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission," *Proc. ACM SIGCOMM '97,* pp. 289-300, Sept. 1997.
[27] J. Liu, B. Li, and Y. Zhang, "An End-to-End Adaptation Protocol for Layered Video Multicast Using Optimal Rate Allocation," *IEEE Trans. Multimedia,* vol. 6, no. 1, pp. 87-102, Feb. 2004.
[28] L. Nahm, Q. Li, and C.-C.J. Kuo, "Equation-Based Layered Video Multicast with Explicit Congestion Notification," *Proc. IEEE Global Telecomm. Conf.,* pp. 3580-3584, Dec. 2003.
[29] Z. Zhang, "Network-Supported Internet Multicast Congestion and Error Control," PhD dissertation, Dept. of EEE, Univ. of Hong Kong, Aug. 2002.
[30] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking,* vol. 1, no. 4, pp. 397-413, Aug. 1993.
[31] D.M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithm for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems,* vol. 17, pp. 1-14, 1989.

**Zaichen Zhang** received the BS and MS degrees in electrical and information engineering from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the PhD degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 2002. He joined the Southeast University as a postdoctoral researcher in 2002 and became an associate professor there in 2004. His current research interests include wireless communications, Ultra-Wideband (UWB) technology, and Internet multicasting. He is a member of the IEEE.

**Victor O.K. Li** received the SB, SM, EE, and ScD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, Massachusetts, in 1977, 1979, 1980, and 1981, respectively. He joined the University of Southern California (USC), Los Angeles, in February 1981 and became professor of electrical engineering and director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong, Hong Kong, where he is the chair professor of information engineering in the Department of Electrical and Electronic Engineering. He is a principal investigator of the Area of Excellence in Information Technology funded by the Hong Kong government. His research interests include information technology, including all-optical networks, wireless networks, and Internet technologies and applications. He has also served as managing director of Versitech (http://www.versitech.com.hk/), the technology transfer and commercial arm of the University, and on various corporate boards. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. He chaired the Computer Communications Technical Committee of the IEEE Communications Society (1987-1989) and the Los Angeles chapter of the IEEE Information Theory Group (1983-1985). He cofounded the International Conference on Computer Communications and Networks (IC3N) and chaired its steering committee (1992-1997). He also chaired various international workshops and conferences, including, most recently, IEEE INFOCOM 2004 and IEEE HPSR 2005. He has served as an editor of *IEEE Network*, the *IEEE Journal on Selected Areas in Communications* wireless communications series, and *Telecommunication Systems*. He also guest edited special issues of the *IEEE Journal on Selected Areas in Communications*, *Computer Networks and ISDN Systems*, and the *KICS/IEEE Journal of Communications and Networking*. He is now serving as an editor of *ACM/Springer Wireless Networks* and *IEEE Communications Surveys and Tutorials*. He has been appointed to the Hong Kong Information Infrastructure Advisory Committee by the chief executive of the Hong Kong Special Administrative Region (HKSAR). He is a part-time member of the Central Policy Unit of the Hong Kong Government. He also serves on the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Programme Committee, the Engineering Panel of the Research Grants Council, and the Task Force for the Hong Kong Academic and Research Network (HARNET) Development Fund of the University Grants Committee. He was a distinguished lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute. He has also delivered keynote speeches at many international conferences. He has received numerous awards, including, most recently, the Ministry of Education Changjiang Chair Professorship at Tsinghua University, Beijing, the UK Royal Academy of Engineering Senior Visiting Fellowship in Communications, the Outstanding Researcher Award of the University of Hong Kong, the Croucher Foundation Senior Research Fellowship, and the Order of the Bronze Bauhinia Star, Government of HKSAR, China. He was elected an IEEE fellow in 1992. He is also a fellow of the HKIE and the IAE. For further information, see http://www.eee.hku.hk/staff_personal/vli.htm.