

Layered Multicast with Forward Error Correction (FEC) for Internet Video

ZaiChen Zhang and Victor O. K. Li

The University of Hong Kong

Hong Kong, China

Abstract—In this paper, we propose RALF, a new FEC-based error control protocol for layered multicast video. RALF embodies two design principles: decoupling transport layer error control from upper layer mechanisms and decoupling error control and congestion control at the transport layer. RALF works with our previously proposed protocol RALM — a layered multicast congestion control protocol with router assistance. RALF provides tunable error control services for upper layers. It requires no additional complexities in the network beyond those for RALM. Its performance is evaluated through simulations in NS2.

I. INTRODUCTION

The Internet is evolving from a pure data network to a multimedia network. Disseminating real-time video to many receivers over the Internet is important for many applications, such as Video on Demand (VoD) and tele-medicine. Multicast [1] is an efficient approach for one-to-many delivery. However, today's Internet multicast is best effort. Proper congestion control mechanisms are necessary for multicasting real-time video over the Internet. Error control also helps provide performance enhancements.

The Internet multicast congestion control problem has three major challenges: scaling to large number of receivers, dealing with heterogeneity in the network and among the receivers, and being compatible with other traffics, such as TCP.

Layered multicast [2], [3] is proposed to solve the heterogeneity problem. In a basic layered multicast scheme, the sender encodes the original video stream into several layers, and sends each layer to a separate multicast group. The layers are cumulative. There is a basic layer and several higher layers. The basic layer can be independently decoded. Higher layers, which provide performance enhancements, can only be decoded with some or all of the previous layers. A receiver makes join/leave decisions on the layers based on observed network conditions. It tries to join as many layers as it can handle and adapts to network conditions dynamically. This receiver-driven approach also alleviates the "feedback implosion" problem [4], in which feedback messages from many receivers to the sender congest the network and overwhelm the sender.

Receiver adaptation in layered multicast congestion control can be achieved through probing or equation-based approaches. Receiver-driven Layered Multicast (RLM) [3] proposed a probing mechanism, called "join-experiment," which makes join/leave decisions based on observed packet losses. Basically, a receiver joins 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. In an equation-based 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 subscribes to a proper number of layers directly.

Although many real-time applications are loss-tolerant, excessive packet losses may lead to performance degradation. There are basically two error control mechanisms, Automatic Repeat reQuest (ARQ) [5] and Forward Error Correction (FEC) [6]. FEC and hybrid ARQ/FEC are widely used in the multicast environment.

Packet-level FEC, which deals with erasures instead of bit errors, is used in multicast error control. At the packet level, an erasure is a lost packet with known location in a data stream. A commonly used code is the Reed-Solomon Erasure (RSE) [7] code. It is designed based on the well-known Reed-Solomon (RS) code [6], but works at the packet level. The RSE code encodes a block of packets into an n -packet codeword, with $h = n - k$ redundant (parity) packets. The code is often systematic which means that the original k packets are included in the codeword in a clear form. Receiving any k out of the n packets in the codeword is enough for decoding the original k packets.

Incorporating FEC in layered multicast can be achieved by layered FEC [8], [9]¹. In layered FEC, parity packets are organized into separate layers (the FEC layers) and sent to different multicast groups. Receivers join one or more FEC layers for error correction as needed. In [8], data layers are sent together and FEC layers are delayed by different times. As a result, receivers can make tradeoffs between FEC protection level and latency, individually. A pseudo-ARQ scheme is proposed in [9], where receivers ask for parity packets from delayed FEC layers when data packets are lost. Delaying FEC layers also helps alleviate the effect of burstiness in packet losses.

Proposed layered FEC schemes try to achieve optimal bandwidth allocation between data and FEC layers, so that maximized user utility can be obtained with given available bandwidth and packet loss ratio. However, optimal bandwidth allocation is determined by source coding and channel coding schemes, and also related to a utility function which maps 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 there are variable time delays in joining/leaving multicast groups in the Internet environment, whether an optimal solution can achieve its desired performance in a real implementation is quite questionable.

In this paper, we propose an alternative approach — decoupling multicast transport congestion control and error control from upper layers. We do not attempt to achieve the overall optimal performance which involves source coding and user utility issues. Instead, we propose a thin transport layer error

¹In this paper, the term "layered FEC" is not restricted to the layered FEC protocol proposed in [8].

control service which can be tailored for different upper layer schemes by adjusting a set of parameters.

We also propose decoupling error control from congestion control in layered multicast. Most existing protocols conduct them together, usually through the equation-based approach. A receiver measures packet loss ratios, estimates available bandwidth in the bottleneck link using the measured loss ratios, 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 purpose. This average loss ratio is necessary in congestion control for achieving protocol stability. Furthermore, since receivers downstream to the same bottleneck link will observe similar average loss ratios, fair share of bandwidth can be achieved using the equation-based approach. However, packet losses actually occur in bursts in networks, so the smoothed long term average loss ratio is not suitable for determining proper FEC protection levels.

We solve this problem by using two protocols for congestion control and error control, separately, but under the same framework. We have proposed Router-Assisted Layered Multicast (RALM) [10], which is a layered multicast congestion control protocol for real-time applications. As in RLM, RALM adapts to network status using join-experiments — a probing approach. Different from RLM, RALM relies on additional network mechanisms to achieve fair share among different sessions and avoid unnecessary congestion caused by failed join-experiments. In this paper, we introduce a new error control protocol, Router-Assisted Layered FEC (RALF), which works with RALM. RALF uses instantaneous observed packet losses for error control.

The rest of this paper is organized as follows. In Section II, we introduce RALM briefly. The detailed protocol can be found in [10]. The RALF protocol is described in Section III. We have implemented RALF in the Network Simulator (version 2) (NS2) [11]. Simulation results are given in Section IV. Finally, Section V concludes this paper.

II. ROUTER-ASSISTED LAYERED MULTICAST

In the section, we introduce the RALM protocol briefly. RALM is a receiver-driven layered multicast protocol with router assistance. It can be incrementally deployed. If all the routers are unaware of RALM, the protocol defaults to RLM. It outperforms RLM and the additional state and processing required in RALM-aware routers are not excessive. It is easy to implement, and compatible with current multicast protocols.

In RALM, the sender encodes the original video 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 $k - 1$ is B_d^k , which we call the Lower End Bandwidth (LEB) of layer k . The value of B_d^k should be communicated to receivers joining the group carrying layer k . For the basic layer, B_d^1 is 0.

A basic idea of RALM is router-initiated suspension/retry. A RALM-aware router monitors the buffer status at each of its outgoing links. If congestion occurs at an outgoing link, the router will immediately suspend some of the current transmitting groups, i.e. stop sending packets of the groups to that out-

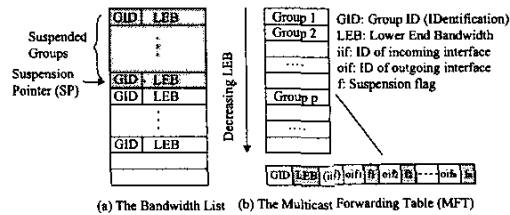


Fig. 1. The bandwidth list and MFT.

going 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.

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 smaller LEB has higher priority. A RALM-aware router maintains a bandwidth list, as shown in Fig. 1(a), at each outgoing link. The list caches groups that are likely to be suspended or retried. A RALM receiver puts a group’s LEB value in an 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 Multicast Forwarding Table (MFT) at the router, there is a suspension flag associated with each outgoing interface in each group’s entry, as shown in Fig. 1(b). 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 subcasting² a suspend, retry, or drop message to all receivers in the group downstream to the link. RALM receivers perform all RLM operations for supporting incremental deployment. They also react to control (suspend, retry, and drop) messages from RALM-aware routers. Specifically, there is a “Suspended” state in the state machine of the RALM receiver protocol. A receiver will change to this state when it knows from the control messages that at least one of its subscribed layers is currently suspended by a RALM-aware router. In this state, it will not join or leave layers.

III. ROUTER-ASSISTED LAYERED FEC

RALF assumes that the original video stream is compressed by a scalable source coding scheme [12], [13]. In a typical scalable coding, video frames are blocked as Group of Pictures (GOP). Frames in a GOP are compressed together and the resulting bit stream is embedded. An embedded bit stream can be

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

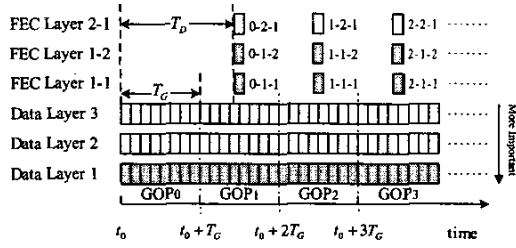


Fig. 2. Data organization in RALF.

divided into several sub-streams that allow decoding at multiple rates. In layered multicast, each sub-stream is packetized into several packets, and packets in the same sub-stream but from different GOPs are organized as one data layer and sent to a multicast group. Usually, some layers are more important than others. They may contain more information or be necessary for decoding other layers. The concepts are illustrated in Fig. 2. In a typical layered FEC scheme, error protection is provided by encoding data packets from each GOP using a channel coding scheme. RSE code is commonly used for this purpose. The generated parity packets (FEC packets) are organized into several FEC layers and sent to different multicast groups. Usually, the RSE coding is performed on each data layer, and the resulting FEC layers are delayed with respect to corresponding data layers. Different from other layered FEC schemes, where FEC layers are delayed with different time values, RALF sends all FEC layers together. We believe this is preferred since it simplifies the operation and avoids the possibility of receivers missing their deadlines due to excessive FEC layer delays. The data organization of RALF is shown in Fig. 2.

In this figure, there are three data layers. Each data layer has eight packets for each GOP. Data layer 1 is the most important one and protected by two FEC layers. Data layer 2 is protected by one FEC layer. There is no FEC protection for data layer 3. In this example, there is one packet in each FEC layer for each GOP, but more packets are also possible. In Fig. 2, index $i-j-k$ means that the FEC 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. Its value is chosen as following:

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

where $T_{D,min} = T_G + t_{d,max} + t_{rtt,max} + t_{0,max}$. $t_{d,max}$ is the maximal value of a detection time — the time used 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.

Fig. 3 illustrates the case where a receiver asks for FEC packets from the sender. In this figure, the first packet of a GOP is sent at time 0 and the last packet of this GOP is lost. After detecting the loss, the receiver requests an FEC packet for this GOP by subscribing to an FEC layer. The join message arrives at the sender after t_j and the FEC packet is received at time t_1

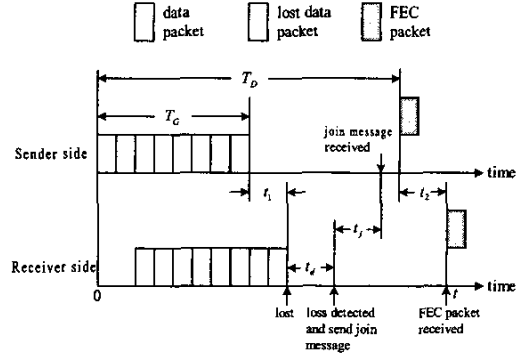


Fig. 3. Setting T_D in RALF.

and t_2 are the transmission times of the lost packet and the FEC packet, respectively. Since an FEC packet should be available when the join message arrives, we have:

$$\begin{aligned} T_D &\geq T_G + t_1 + t_d + t_j = T_G + t_d + t_j + t_2 + t_1 - t_2 \\ &= T_G + t_d + t_{rtt} + t_0, \end{aligned} \quad (2)$$

where $t_0 = t_1 - t_2$ is the delay jitter between the lost packet and the FEC packet on the path from the sender to the receiver. t_d and t_{rtt} are the detection time and RTT in this example, respectively. If the receiver joins the FEC layer at an on-tree router, we prove that (2) still holds except that t_{rtt} is now the RTT between the on-tree router and the receiver [14]. Replacing t_d , t_{rtt} , and t_0 with their maximal values and considering the upper delay bound T_R proposed by source coding, we get (1).

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 the estimated t_d , t_{rtt} , and t_0 values is smaller than T_G and T_R , T_D can be chosen in a relatively wide range in which RALF's performance is not sensitive to the T_D value. This condition holds for disseminating video streams in a typical Internet environment. RALF is not efficient to be used in an environment where t_d , t_{rtt} , and t_0 is 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. 2, each data layer is protected by zero, one, or more FEC layers. In a typical setting, an FEC layer contains one FEC packet for each GOP. In RALF, FEC layers for the same data layer have almost the same priority, which is lower than that of the 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 $B_{f,m}^k = B_d^k + \epsilon_m$, where B_d^k is the LEB of data layer k , ϵ_m is a very small positive number, and $\epsilon_{m1} > \epsilon_{m2}$ if $m1 > m2$. As a result, we have $B_d^k < B_{f,m}^k < B_d^{k+1}$. In RALM, the bandwidth list maintenance algorithm [10] swaps groups with the same LEB values, so that they can be fairly served. This is not necessary for FEC layers, since their bandwidths are much smaller than those of data layers. Therefore, we set slightly different ϵ_m for different FEC layers to avoid such swapping

and reduce the processing burden on the network. By default, RALF sets the priorities of FEC layers for data layer k to be higher than that of data layer $k+1$. Under RALM, packet losses are relatively rare, and joining one or a small number of FEC layers for a short time is usually enough for error protection and will not affect higher layers too much. However, applications using RALF can also choose to reduce FEC layers' priorities if needed, by setting higher LEB values for the FEC layers.

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. (Setting T_D to compensate for the maximum delay between lost data packets and the corresponding FEC 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 $T_H \geq T_D + t_{0,max} - t_{i,min}$, where $t_{i,min}$ 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 a time $T'_G = T_G + t_{0,max}$, it will be automatically decreased by n_0 . A receiver joins or leaves FEC layers for a data layer based on holding timers and the loss counter. Suppose each 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. Fig. 4 shows examples of the loss counter and joining/leaving FEC layers.

The above scheme is greedy. If n_t losses occur within one T'_G , 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.

IV. SIMULATION RESULTS

We simulated RALF in NS2. Fig. 5 shows the simulation topology. In this topology, there are ten RALF sessions and ten TCP sessions sharing a bottleneck link. Each session has

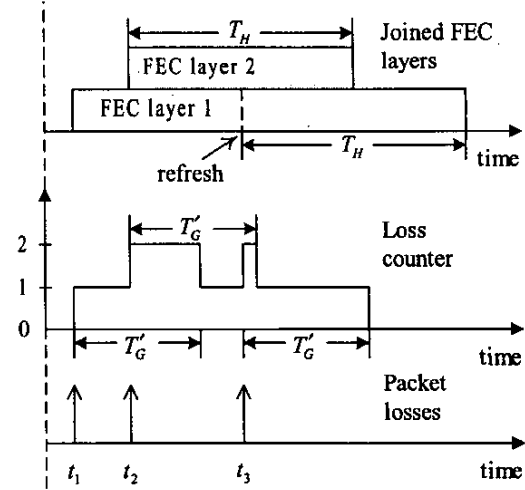


Fig. 4. Loss counter and joining/leaving FEC layers.

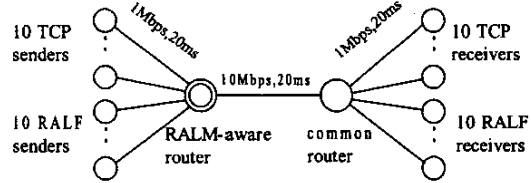


Fig. 5. Simulation topology.

one sender and one receiver. The bottleneck link bandwidth is 10 Mbps, corresponding to 500 Kbps fair share for each session. Other link bandwidths are all 1 Mbps. Propagation delay of each link is 20 ms. RALM is enabled at the router above the bottleneck link. Each simulation runs for 1000 simulated seconds.

Source data are encoded into ten data layers, with bandwidth of 80 Kbps for each layer. The GOP duration is $T_G = 1$ second, and the packet (payload) size is 1 KBytes. Therefore, a data layer has ten packets in each GOP. The ten data packets are encoded using an RSE code with parameters $k = 10$ and $h = 3$. As a result, three FEC 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, bandwidth of each FEC layer is 8 Kbps, and the maximum number of FEC layers for each data layer is three. Other RALF parameters are set as $T_D = 2$ seconds and $T_H = 2$ seconds.

Fig. 6 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 I.

The top figure in Fig. 6 plots received packets in a RALM session, where RALF is not enabled. 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 I, 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

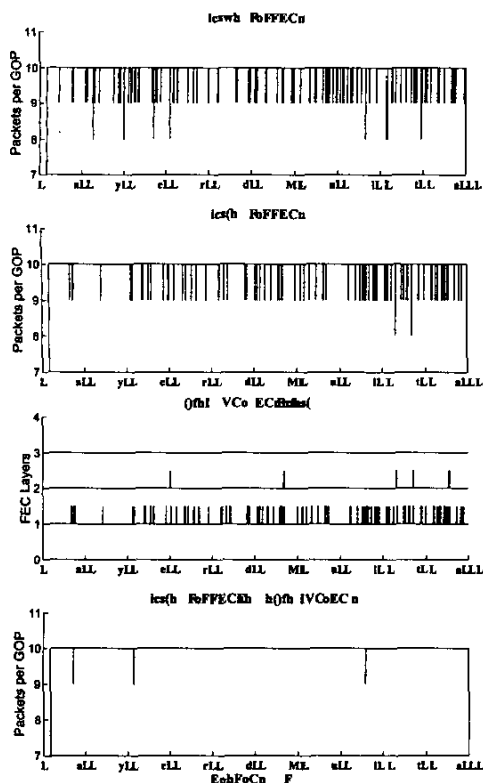


Fig. 6. FEC protection in RALF.

to congestion better than the data layers. The third figure in Fig. 6 plots received 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 RSE code. The result is near optimal: only three packets are lost during the 1000 second simulated time. For other active (subscribed and not suspended) data layers, the loss patterns and error protection effects are similar, as reflected in Table I.

Table I also records the total number of redundant FEC packets for each data layer. When the sum of received data packets and FEC packets for one GOP exceeds the total data packet number in a GOP (ten 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 I. However, since 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 I as an example, there are only 70 redundant FEC packets with the 9778 received data packets.

TABLE I
ERROR PROTECTION IN RALF.

	Layer	Data Rcvd	Lost Data	FEC Rcvd	Redundant FEC	Unrecoverable 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

V. CONCLUSIONS

In this paper, we proposed the RALF protocol. It works with RALM for error control. RALF embodies two new principles in designing a layered FEC protocol: decoupling transport layer control mechanisms from upper layers and decoupling error control and congestion control at the transport layer. As a result, RALF achieves good performance in error control and provides tunable error control services for different applications. Simulation results of RALF are also given in this paper.

ACKNOWLEDGMENT

This research is supported in part by the Research Grants Council, Hong Kong, Grant No. HKU 7044/02E.

REFERENCES

- [1] S. Deering, "Host extensions for IP multicasting," Network Working Group, RFC 1112, Aug. 1989.
- [2] S. Deering, "Internet multicast routing: State of the art and open research issues," Multimedia Integrated Conferencing for Europe (MICE) seminar at the Swedish institute of computer science, Stockholm, Oct. 1993.
- [3] S. McCanne and V. Jacobson, "Receiver-driven layered multicast," in Proc. ACM SIGCOMM'96, Palo Alto, CA, Aug. 1996, pp. 117-130.
- [4] R. Yavatkar and L. Manoj, "Optimistic strategies for large-scale dissemination of multimedia information," in Proc. ACM Multimedia'93, Anaheim, CA, Aug. 1993, pp. 13-20.
- [5] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [6] Shu Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [7] A. J. McAuley, "Reliable broadband communication using a burst erasure correcting code," in Proc. ACM SIGCOMM'90, Philadelphia, PA, Sept. 1990, pp. 297-306.
- [8] W.-T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, Mar. 2001, pp. 373-386.
- [9] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," Microsoft Research Technical report, MSR-TR-99-86, Nov. 1999.
- [10] ZaiChen Zhang and Victor O. K. Li, "Router-assisted layered multicast," to appear in *IEEE ICC 2002*, New York, Apr./May 2002.
- [11] The Network Simulator — ns-2, <http://www.isi.edu/nsnam/ns>.
- [12] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE E J. Select. Areas Commun.*, vol. 15, no. 6, Aug. 1997, pp. 983-1001.
- [13] W.-T. Tan and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. on Multimedia*, vol. 1, no. 2, June 1999, pp. 172-186.
- [14] ZaiChen Zhang, "Network-supported Internet multicast congestion control and error control," PhD Thesis, Dept. of EEE, The University of Hong Kong, Hong Kong, July 2002.