Fast-response Receiver-driven Layered Multicast

Hon Sun Chiu and Kwan L. Yeung
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong, PRC.
Tel: (852) 2857-8493 Fax: (852) 2559-8738
E-mail: {hschiu,kyeung}@eee.hku.hk

Abstract - *In this paper, a new layered multicast protocol, called Fast-response Receiver-driven Layered Multicast (FRLM), is proposed. The differences between our FRLM and the original RLM are only at the receivers. Our design allows the receivers to track the available network bandwidth faster; this enables the receivers to converge to their optimal number of subscribed layers quicker, and to respond to the network congestion prompter. An early trigger mechanism for shortening IGMP leave latency is also designed. We show that FRLM can avoid several potential problems with the original RLM, which have been overlooked previously. Last but not the least, FRLM is a practical scheme that can be readily implemented in today's best-effort Internet.*

1. Introduction

Multimedia applications become more and more popular in the Internet. For real-time multimedia, multicast distribution is the most efficient way for transmitting multimedia data from the same source to a group of receivers at the same time.

Receiver-driven Layered Multicast (RLM) was first introduced by Steven McCanne *et al.* [1] for multimedia transmission over the Internet. The source encodes the video signal into cumulative layers, and transmits each layer on a separate IP multicast group. Each receiver makes decision on adding or dropping a layer according to its own experience on congestion. Thus RLM can cope with bandwidth heterogeneity and can adapt to changing network conditions. There are, however, several drawbacks of this protocol. First, RLM cannot distinguish whether a packet loss is due to join-experiment or due to network congestion. Second, the response time to network congestion is long. Third, a receiver may subscribe to layers higher than its bottleneck link allows, i.e. over-subscription may occur.

A TCP-friendly version of RLM, called RLC, was introduced by Lorenzo Vicisano *et al.* [2], in which synchronization points (SPs) are used to synchronize the time for the receivers to add a layer. With the use of exponentially distributed layering rate structure, the resulting scheme becomes more TCP-friendly. However, the large layer granularity at higher layers may lead to under-utilization of bandwidth. Adding SPs to the multimedia data streams is also troublesome. There are also some other drawbacks as described in [3].

Many protocols have been proposed to overcome various weaknesses of layered multicast. PLM [4] uses packet pair (PP) to discover the available bandwidth, with the aim of converging faster. CALM [5] locates the congested link with the aim of reacting to congestion faster (drop a layer within one RTT). FLID-DL [6] introduces dynamic layering (DL) to ease the problem of long IGMP leave latencies [7, 8]. Fine-Grained Layered Multicast [9] uses non-cumulative layering, and STAIR [10] employs both dynamic and static layering, and cumulative and non-cumulative layering, to emulate the TCP additive increase and multiplicative decrease (AIMD) behavior. More recently, RALM [11] maintains state in network router to improve its congestion control scheme. All of the mentioned protocols either add extra workload to both sender and receivers, or increase the number of add/drop procedures, or require router assistance which cannot be provided in the current best-effort Internet.

In this paper, a new layered multicast scheme based on the original RLM [1] is proposed. We call it Fast-response Receiver-driven Layered Multicast (FRLM). Unlike existing approaches, FRLM does not add extra workload to both sender and receivers, requires no modification of the existing best-effort network routers, and thus faces no development hurdles. In short, FRLM is a practical scheme (like RLM) that can be readily implemented in today's Internet.

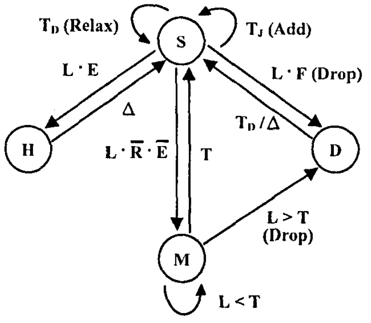
2. Fast-response RLM

Fast-response RLM (FRLM) is designed to allow the receivers to track the available network bandwidth faster. It can avoid several potential problems with the original RLM. The differences between our FRLM and RLM are only at the receiver. Detailed explanation of RLM can be found in [1].

Fig. 1 shows the state machine of a FRLM receiver. Like RLM receiver, it also consists of 4 states, steady state (S), hysteresis state (H), measurement state (M),

0-7803-8623-X/04/$20.00 ©2004 IEEE

and drop state (D). But the state transition is a bit different from RLM. Several additional events are introduced. In particular, event E means currently there is at least one active join-experiment (by other receivers) at a layer higher than the receiver under consideration. We are only interested in higher layer join-experiments because any lower layer join-experiment will not cause loss to the receiver under consideration. Noted that event E is known by each receiver because every receiver notifies the entire group [1] before conducting a join-experiment, and its duration is also known.



E	: there is at least 1 active join-experiment at a layer higher than the receiver under consideration.
Δ	: remaining join-experiment time
T	: adaptive loss threshold
T_D / Δ	: if drop due to failed join-experiment, use Δ; otherwise, use T_D

Figure 1. State machine for FRLM receivers

If a receiver experiences a packet loss (L) in S state, there are three possible state transitions depending on:

(1) Event F: the receiver has an active join-experiment, and the layer being added is the highest layer among other concurrent join-experiments.

(2) Event $\overline{R} \cdot \overline{E}$: the receiver does not have an active join-experiment, and does not detect any active join-experiments with layer higher than itself.

(3) Event E: there is at least one active join-experiment (by other receivers) at a layer higher than the receiver under consideration.

In case (1), the join-experiment failed and the state transition is from S to D. The highest subscribed layer is dropped. The receiver will then stay in D state for the remaining time[1] (Δ) of the failed join-experiment. When Δ expires, the receiver returns to S state.

In case (2), the receiver enters M state and measures the degree of network congestion, and determines whether a drop action is necessary.

In case (3), the receiver enters H state in order to filter out the packet loss effect caused by other active join-experiments conducting in higher layers. The amount of time it should stay in H state is the *remaining* join-experiment time (Δ) of the most recently initiated join-experiment. When the timer expires, the receiver returns to S state.

In the following section, our receiver design is justified in contrast to the original RLM receiver design in [1]. Note that solid knowledge on RLM's receiver design is needed for in-depth understanding of the next section.

3. Protocol Details

3.1. Actions on packet loss

There are two potential causes for packet loss: *join-experiment* and *network congestion.* We argue that when congestion is detected, the receiver should promptly drop a layer to ease the congestion. This would also help the receiver to maintain a higher level of receiving quality[2].

In FRLM, if a receiver is not engaging in an active join-experiment, and no other active join-experiments are being conducted by other receivers at any higher layers, a packet loss must be caused by network congestion. So in Fig. 1, this triggers a direct state transition from S to M, which bypasses H state (as required by RLM) and thus shortens the congestion detection time. Besides, all loss statistics can now be properly counted in M state for effective congestion detection.

3.2. Transition from H state

If the loss is caused by other receivers' active join-experiments, entering H state has already filtered out its effect and therefore there is *no need to enter M state* afterwards, as in the original RLM [1]. Because of this, the state transition of H=>M is modified to H=>S in FRLM. Since a receiver *can* distinguish the cause of packet loss, entering H state is purely based on the fact that there is at least one active join-experiment by another receiver at a higher layer. When the timer expires, all other active join-experiments should be finished. So there is no need to enter M state for measurement (as required by RLM), and should transit back to S state directly.

3.3. Adaptive threshold value

[1] Let t_i denote the initiation time of the failed join-experiment and t_D denote the time of entering D state. Then $\Delta = T_D - (t_D - t_i)$.

[2] If the packet loss rate is high, the receiving quality can be improved by dropping some higher layers so as to obtain a smaller packet loss rate.

Let T denote the threshold in M state. In RLM, T is a fixed value of 25% [3]. We believe the intention of using such a large value is to prevent an unnecessary layer drop from happening due to *short-term/transient* network congestion. But if the congestion is *persistent* and the loss rate is just less than 25%, no layer will be dropped. The video quality becomes unacceptable as almost one out of four packets will be lost.

A network congestion episode is transient or persistent can be derived from the time gap between the current transition from S to M and the immediate last transition from M to S. Without loss of generality, let t denotes this time gap. If $t < T_D$, the current visit[3] to M state and the previous visit to M state are said to be *correlated*, i.e. very likely they are caused by the same *persistent* congestion episode. This likelihood grows as t decreases. So when t is small, the threshold T should be reduced for more accurate and faster detection. If $t > T_D$, we simply assume that the two immediate visits to M state are caused by two different congestion episodes. So there is no need to adjust T.

In FRLM, the adaptive behavior of threshold T is governed by the following equation:

$$T = \begin{cases} 0.05 + \dfrac{0.2}{T_D}t & t < T_D \\ 0.25 & t > T_D \end{cases} \qquad (1)$$

where t is the time between the two immediate transitions and T_D is the detection-time period. We can see that T varies linearly with t from 0.05 to 0.25. The minimum threshold is set to 5%. Here we assumed that 5% or more persistent packet loss cannot be tolerated by the associated applications. If $t > T_D$, the maximum threshold of 25% is used for filtering out the possible transient congestion, as in the original RLM.

3.4. Time spent in H state and D state

For both RLM and FRLM, if a receiver enters H state or D state, it has to stay there for some time. The reason behind is to absorb the side-effect caused by other receivers' join-experiments, and/or to prevent the receiver from over-reacting to losses caused by the remaining effect of its own failed join-experiment.

In FRLM, we use a timer Δ. Its value is less than or equal to T_D. If the receiver is in H state, Δ denotes the remaining time of the most recently initiated join-experiment in a layer higher than that of the current receiver. Let t_i denote the initiation time of the most recently initiated join-experiment. Let t_H denote the time of entering H state. Then $\Delta = T_D - (t_H - t_i)$.

Similarly, if the receiver is in D state resulting from an S=>D transition, Δ denotes the remaining time of the failed join-experiment. Let t_i denote the (failed) join-experiment start time and t_D denote the time entering D state. Then $\Delta = T_D - (t_D - t_i)$.

It should be noted that the values of t_i, t_H and t_D are available at the receiver and the associated processing overhead is minimal.

3.5. Concurrent join-experiments

RLM favors the newly joined receivers by giving priority to join-experiments at lower layers. This enables the late joiners to converge faster. However, it also produces two major side-effects: extra delay in M state when the higher layer join-experiment causes packet loss, and over-subscription problem (Refer to Figs 5-7).

When a receiver is conducting a join-experiment, there are either other higher layer join-experiments in progress, or itself is the highest. In FRLM, if packet loss is experienced in the second case, the receiver simply drops the offending layer and enters D state, just the same as RLM. If it is the first case, all other higher layer join-experiments must have started earlier than its own. The receiver enters H state to filter out their effect by waiting for the remaining time (Δ) of the latest higher layer join-experiment. Then it transits back to S state to finish its own join-experiment. At this moment, it has the highest subscribed layer and thus a single packet loss can fail its join-experiment. If the join-experiment succeeds, it can immediately conduct another join-experiment as soon as T_J expires, without an extra delay of T_D in M state, as required in RLM.

3.6. Further improvement - *Early Trigger*

In IGMP [7, 8] protocol, when a host wants to leave a multicast group, it sends a leave message to the last hop multicast router. Then that router polls all downstream hosts 3 times to determine whether or not to unsubscribe the group. The whole process takes 3 to 9 seconds. This long latency is undesirable in reacting to network congestion.

In FRLM, an *early trigger* mechanism is designed. Each receiver maintains 2 packet loss thresholds, a smaller threshold for early trigger and a normal/larger threshold for detecting network congestion. When the smaller threshold is exceeded, a leave request is sent to the last hop router to trigger the polling process.

If the packet loss rate does not exceed its normal threshold at the end of the polling process, the receiver simply sends a standard IGMP reply message to the earlier polling. The router then ignores the previous leave group request. On the other hand, if the normal threshold is exceeded before the completion of the polling process, the receiver takes no action and the

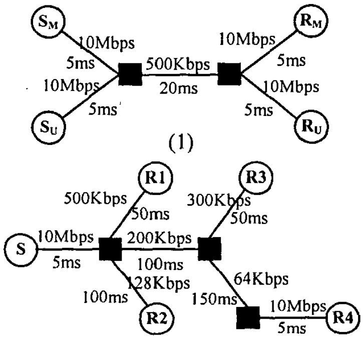
[3] Note that each visit to M state is caused by packet loss (L) event in S state.

group/layer will be dropped as soon as the router finishes its polling.

4. Performance Evaluations

In this section, the performance of our proposed FRLM algorithm is evaluated by simulations using the network simulator *ns* [12]. For performance comparison, original RLM and a trimmed down version of FRLM, called Intermediate Algorithm (IA) are also implemented. The IA algorithm adopts the modified state machine shown in Fig. 1, but with the original RLM's fixed threshold and sojourn time T_D. For simplicity, the early trigger mechanism designed in Section 3.6 is not simulated.

4.1. Simulation topologies



(1)

(2)

Figure 2. Simulation topologies

Simulation results presented in this paper are based on the two network topologies shown in Fig. 2. Topology 1 is a single-bottleneck topology consists of one multicast session (S_M, R_M), and one unicast session (S_U, R_U). S_U is a 300Kbps CBR source, starts at 400s. The purpose of this scenario is to study in the first part, i.e. normal behavior before starting the CBR source, and in the second part, i.e. their behavior in handling persistent network congestion.

Topology 2 consists of a multicast session with a single source and four receivers. Two simulations are carried out on this topology. First, all receivers start at the same time at 5s. Second, receivers start at different time: R1 starts at 5s, then new receivers join at every 200s interval in the order of R2, R3 and R4. The aim of this topology is to investigate the over-subscription problem.

4.2. Simulation environment

In both topologies, lossless links are employed. All the routers use FIFO scheduling and only provide best-effort services. The queue size is set to 15 packets. Without loss of generality, we set the packet size (FRLM, RLM and CBR) to 500 bytes.

For layered multicast transmission, all the parameters used for RLM and FRLM, are set to the same values as in [1], namely α=2, β=2/3, k_1=1, k_2=2, g_1=0.25, g_2=0.25, T_J^{min} =5s and T_J^{max} =600s. The layering rates used in our simulations are static 20Kbps layer granularity from 20, 40,..., to 600Kbps.

4.3. Simulation results and discussion

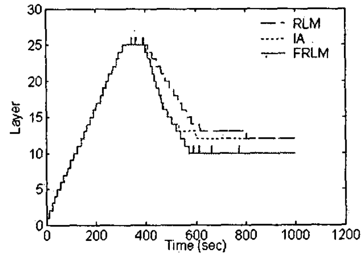
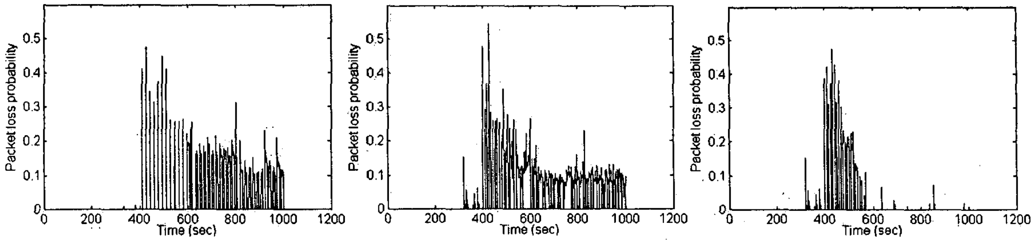


Figure 3. Layer subscription, topology 1

The simulation results of topology 1 are shown in Figs. 3 and 4. Fig. 3 presents the layer subscription over time while Fig. 4 presents the packet loss probability in M state. We focus on the M state because in which a layer drop decision is made, it also provides insights on persistent network congestion even when the packet loss probability is less than the threshold value used in M state.

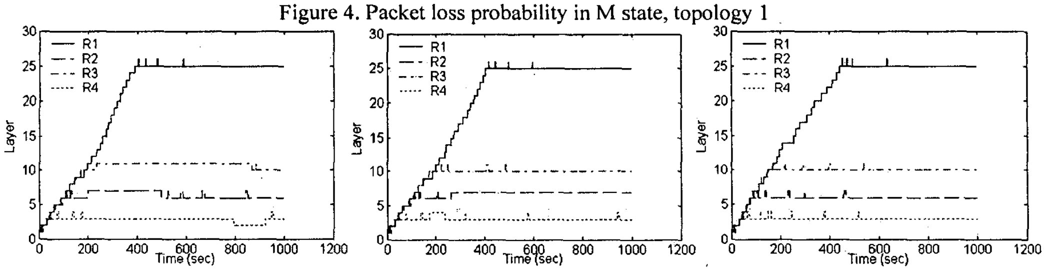
Before starting the CBR source, the three algorithms have the same behavior as it only involves the S state and the join-timer T_J. At 400s, the CBR source starts. The R_M has to drop to layer 10 in order to solve the congestion. The layer dropping rate of RLM is slow when compare with algorithms IA and FRLM. This is because RLM has an extra delay T_D in H state before entering M state. We can see that only FRLM can drop to layer 10. The reason is that FRLM employs adaptive threshold. A drop action is triggered by frequent re-visiting of M state. As a result, the packet loss probability at the converged state is significantly reduced (Fig. 4c).

Fig. 5 shows the layer subscription behavior and Fig. 6 shows the packet loss probability of our second simulation on topology 2, with the four receivers start at the same time. The expected optimal layers of R1, R2, R3 and R4 are 25, 6, 10 and 3 respectively. From Fig. 5, we can see that only FRLM can converge to these optimal values.



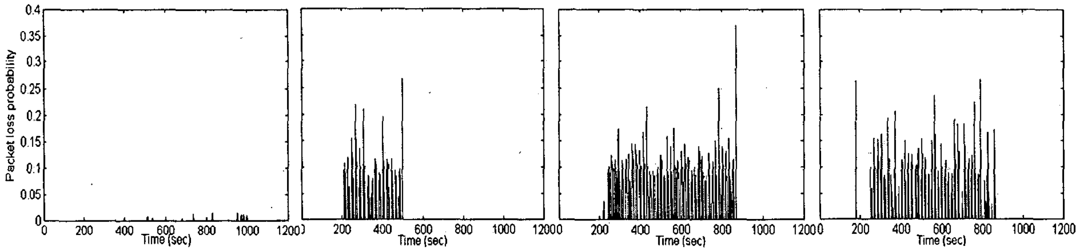
(a) RLM (b) Intermediate Algorithm (IA) (c) FRLM

Figure 4. Packet loss probability in M state, topology 1



(a) RLM (b) Intermediate Algorithm (IA) (c) FRLM

Figure 5. Layer subscription, topology 2 with receivers start at same time



R1 R2 R3 R4

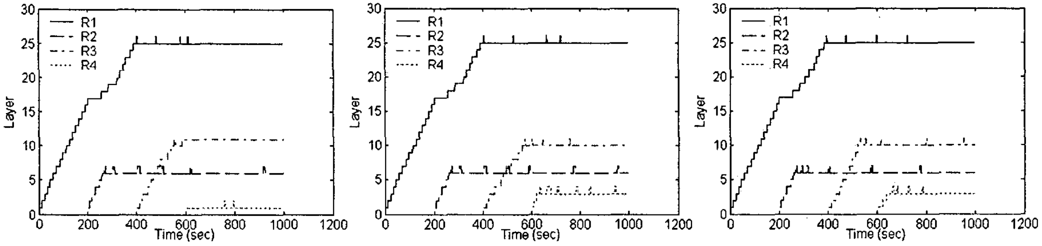
Figure 6. Packet loss probability of RLM in M state, topology 2 with receivers start at same time

In RLM, R2 and R3 (in Fig. 5a) both add layers higher than their optimal layer numbers. The over-subscriptions take place because R1 is conducting a join-experiment in a higher layer. R2 and R3 enter M state instead of failing their join-experiments. Such packet loss does not exceed the threshold of 25% (Fig. 6), so the offending layers are not dropped. Noted that R3's over-subscription causes R4's under-subscription as its bottleneck link is on R4's path.

Although a receiver transits from H state to S state but not M state in the IA algorithm, over-subscription still occurs. The reason is that the join-experiment finishes in H state within T_D. A further packet loss in S state makes R2 enter M state. Similar to RLM, the layer is not dropped, and over-subscription occurs.

As shown in Fig. 5c, all FRLM receivers converge to their optimal layers. This is because the time required to stay in H state is the remaining time of the latest join-experiment conducting in higher layer and adaptive threshold is employed.

The last simulation is done on topology 2 with the four receivers start at different times as stated earlier. Fig. 7 shows the layer subscription behavior of the three algorithms. Over-subscription occurs in RLM as shown in Fig. 7a, that R3 subscribes to layer 11 which is higher than its optimal layer 10. This happens before the starting of R4. When R4 starts, the link shared with R3 has already been congested. It observes a persistent packet loss and therefore cannot conduct join-experiment for adding a higher layer. R4 can only subscribe to the base layer throughout the simulation. There is no over-subscription found in IA and FRLM. This is because the join-timers are randomly chosen. However, according to the pervious simulation that all receivers start at the same time, over-subscription can happen if we only modify the transitions between states. Therefore, this justifies that the remaining time (Δ) of join-experiment and the adaptive threshold should be adopted.



(a) RLM (b) IA (c) FRLM

Figure 7. Layer subscription, topology 2 with receivers start at different time

5. Conclusions

We have proposed an efficient congestion control algorithm for layered multicast: Fast-response Receiver-driven Layered Multicast (FRLM). We showed that FRLM can achieve a fast response in solving network congestion, only about half of the time required by RLM. We also showed that FRLM can prevent the receivers from over-subscribing. This ensures that each receiver converges to their optimal number of subscribed layers faster.

In this paper, FRLM was designed to enhance the basic operation of the receiver-driven cumulative layered multicast congestion control scheme. We have not taken the TCP-friendliness and inter-session fairness into account. This is partially because we do not think it is necessary/possible to make real-time multimedia traffic TCP-friendly. Having said that, algorithms of such aspects can be easily integrated with our FRLM.

Appendix

For the completeness of the paper, the calculations of T_J and T_D are quoted below from [1].

A. Backoff T_J

$$\hat{T}_J^k \leftarrow \min\left(\alpha \hat{T}_J^k, T_J^{\max}\right)$$

where $\alpha > 1$ is the backoff parameter/constant and $T_J^{\max}$ is the maximum join-timer interval.

B. Relax T_J

$$\hat{T}_J^k \leftarrow \max\left(\beta \hat{T}_J^k, T_J^{\min}\right)$$

where $\beta < 1$ is the relaxation constant and $T_J^{\min}$ is the minimum join-timer interval.

C. Detection-time estimation

$$\hat{\sigma}_D \leftarrow (1-g_2)\hat{\sigma}_D + g_2\left|D_i - \hat{T}_D\right|$$

$$\hat{T}_D \leftarrow (1-g_1)\hat{T}_D + g_1 D_i$$

where g_1 and g_2 are gains of first-order low-pass filters, D_i is the time since a join-experiment started to the detection of its failure.

D. Detection-timer

$$T_D = k_1\hat{T}_D + k_2\hat{\sigma}_D$$

where k_1 and k_2 are design constants.

References

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," *Proc. of ACM SIGCOOM*, Stanford, CA, August 1996, pp. 117-130.

[2] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," *Proc. of IEEE INFOCOM*, San Francisco, CA, USA, March 1998, vol. 3, pp. 996-1003.

[3] A. Legout and E. W. Biersack, "Pathological Behaviors for RLM and RCL," *Proc. of NOSSDAV'2000*, Chapel Hill, North Carolina, USA, June 2000, pp. 1-9.

[4] A. Legout and E. W. Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Scheme," *Proc. of ACM SIGMETRICS 2000*, Santa Clara, California, USA, June 2000, pp. 1-10.

[5] S. Wen, J. Griffioen, and K. L. Calvert, "CALM: Congestion-Aware Layered Multicast," *Proc. IEEE OPENARCH 2002*, 2002, pp. 179-190.

[6] J. W. Byers, G. Horn, M. Luby, M. Mitzenmacher, and W. Shaver, "FLID-DL: Congestion Control for Layered Multicast," *IEEE Journal on Selected Areas in Communications*, October 2002, vol. 20, no. 8, pp. 1558-1570.

[7] S. Deering, "Host Extensions for IP Multicasting," *IETF RFC 1112*, August 1989.

[8] W. Fenner, "Internet Group Management Protocol, Version 2," *IETF RFC 2236*, November 1997.

[9] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-Grained Layered Multicast," *Proc. of IEEE INFOCOM 2001*, April 2001, pp. 275-283.

[10] J. Byers and G. I. -Kwon, "STAIR: Practical AIMD Multirate Multicast Congestion Control," *Proc. of NGC'01*, 2001, Full version appears as BU-CS-TR-2001-018, Boston University, 2001.

[11] Z. C. Zhang and V. O. K. Li, "Router-Assisted Layered Multicast," *IEEE ICC 2002*, 2002, vol. 4, pp. 2657-2661.

[12] ns: UCB/LBNL/VINT Network Simulator – ns (version 2), http://www-mash.cs.berkeley.edu/ns/.