Fast-response Receiver-driven Layered Multicast with Multiple Servers*

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
Email: {hschiu,kyeung}@eee.hku.hk

Abstract - **Almost all the proposed layered multicast algorithms support a single server, i.e. a receiver can only subscribe to at most one server. A common restriction to single server approach is that the maximum number of subscribed layers, as well as the maximum achievable throughput is limited by the specific bottleneck link between a receiver and the server. In this paper, a new layered multicast protocol, called Fast-response Receiver-driven Layered Multicast with Multiple Servers (FRLM-MS) is proposed. Our design allows a receiver to subscribe to more than one servers. A FRLM-MS receiver can benefit from multiple paths to the multiple servers, resulting in a higher achievable bandwidth. It in turn allows the receiver to have a higher layer subscription, and thus a better playback performance.**

Keywords – **FRLM-MS, FRLM, RLM, congestion control, layered multicast**

I. INTRODUCTION

Multimedia applications grow rapidly in the Internet. If multiple users want to receive the same data at the same time, multicast transmission is the most efficient way. To handle network heterogeneity, cumulative layered coding is used: a signal is encoded into a number of layers, and each higher layer contains a refinement of the signal transmitted in the lower layers. Receiver subscribes to as many layers as the bottleneck bandwidth (between the receiver and the server) permits.

Receiver-driven Layered Multicast (RLM) was 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 network congestion. RLM can cope with bandwidth heterogeneity and can adapt to changing congestion conditions. However, there are still some weaknesses such as inducing packet loss, and responding slowly to network congestion.

Many other protocols were proposed to overcome various weaknesses of RLM. FLID-DL [2] and CALM [3] enhance the protocol with a faster response to network congestion. HALM [4] and RLM using Active Networks [5] address the issue of TCP-friendliness, inter-session fairness and intra-session fairness. However, all of the mentioned protocols either add extra workload to both sender and receivers, or increase number of the add-drop procedures, or require router assistance which cannot be provided in the current network.

In our previous work, Fast-response Receiver-driven Layered Multicast (FRLM) [6] was proposed. It enhances the RLM protocol by modifying its receiver's state machine, and introducing an adaptive loss threshold. FRLM allows the receivers to have a shorter convergence time, and to respond to the network congestion faster. FRLM also solves the over-subscription problem, which has been overlooked previously.

In this paper, we observe that almost all the proposed layered multicast algorithms [1-6] support a single server. A common restriction to single server approach is that the maximum number of subscribed layers, as well as the maximum achievable throughput, is limited by the bottleneck link capacity between a receiver and the server, which in turn limits the performance of the protocol. To address to this common limitation, we extend our FRLM protocol [6] to support multiple servers, we call it Fast-response Receiver-driven Layered Multicast with Multiple Servers (FRLM-MS). This is a very practical and important extension that benefits from multiple paths from a receiver to multiple servers, resulting in a higher bandwidth, compared with the traditional single-path approach. With this advantage, FRLM-MS allows a receiver to have a higher layer subscription, and thus a better playback performance.

The remainder of the paper is organized as follows. We first compare the performance of single-server approach with the multiple-server approach in the next section. In Section III, operations of FRLM-MS are explained. We then address the implementation issues of FRLM-MS in Section IV. In Section V, the performance of FRLM-MS is evaluated by simulations. Finally we conclude the paper in Section VI.

II. SINGLE-SERVER VS MULTIPLE-SERVER

Fig. 1 shows the overlay multicast tree from 3 servers to 3 receivers. We assume that S1 and R1 are located in an autonomous system (AS1), S2 and R2 are located in AS2, S3 and R3 are located in AS3. Each link can support a maximum throughput of 5 layers.

Consider the traditional single-server approach, e.g. RLM [1]. A receiver has to subscribe the base layer first from a nearby server, say R1 subscribes layer 1 from S1. Then R1 can add/subscribe to higher layers as long as the bottleneck link permits. In this situation, R1 can subscribe up to 5 layers from S1. Similarly, R2 and R3 also subscribe up to 5 layers from S2 and S3 respectively.

Next, consider our proposed FRLM-MS (to be described in the next section in details). When a receiver starts, similar to the single-server approach, it has to subscribe the base layer from a nearby server first. Then the receiver can add layers

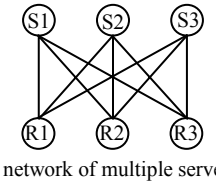


Fig. 1. Overlay network of multiple servers and receivers

* This work is supported by Hong Kong Research Grant Council Earmarked Grant HKU 7150/04E.

according to the bottleneck link's bandwidth. Hence, R1 can subscribe up to 5 layers from S1. R1 tries to add layer 6 from S1 but it fails. At this moment, R1 sends the same add request to the second nearest server S2, and adds layer 6 successfully. Similarly, R1 can subscribe layers 6-10 from S2, and layers 11-15 from S3.

In this simple scenario, we show that multiple-server approach can benefit from having more than 1 servers.

III. FRLM-MS PROTOCOL

In FRLM-MS, the state machine used by a receiver is exactly the same as FRLM [6] as shown in Fig. 2. It consists of 4 states, steady state (S), hysteresis state (H), measurement state (M), and drop state (D). The state transition is triggered by events, e.g. event T_J denotes the join-timer expires. The actions taken on state transition, if any, are shown inside the parenthesis. In particular, there are three main events trigger the state transition from the steady state:

- $L \cdot F$ - Packet lost due to failed join-experiment, a single packet lost can trigger this transition. The receiver enters D state and drops the offending layer;
- $L \cdot E$ - The receiver experiences packet lost, but there is a join-experiment carrying out by other receivers. It enters H state to filter out this effect;
- $L \cdot \overline{R} \cdot \overline{E}$ - Packet lost due to network congestion. The receiver enters M state to measure the loss probability, and triggers a drop action according to the threshold.

More detailed explanations of the FRLM protocol and the state machine are given in [6].

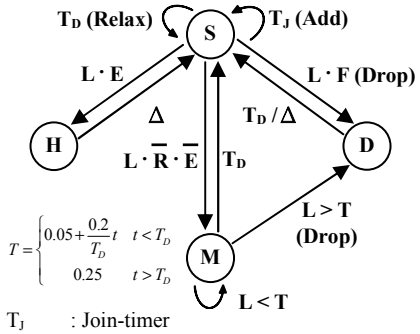
Since a receiver can subscribe layers from more than one server, each receiver maintains a state machine for each server connection. For the operation of FRLM-MS, the protocol consists of 3 phases: start-up, cycle, and recovery.

A. Start-up phase

When a receiver joins a multicast session, it contacts the DNS server. The DNS server returns a list of IP addresses of the available multicast servers, in the order of "nearest server first". This is because in general, the shorter the server-receiver distance, the better the network performance.

Then the receiver subscribes to the first server in the list. The procedure is exactly the same as FRLM (as well as RLM). The receiver adds the base layer first. When the join-timer T_J expires, an add request is sent for adding the next higher layer.

When a layer is dropped, due to failed join-experiment or network congestion, the receiver subscribes to the next server on the list. For example, if layer 10 is to be dropped from server 1, the receiver sends a drop-10 request to server 1. At the same time, the join-10 request is sent to server 2. The receiver will not conduct further add action (join-experiment) in server 1, i.e. stops the corresponding join-timer, with the belief that the bottleneck link is fully utilized. All subsequent add procedures are switched to server 2. The next layer to be added is layer 11 in this case. Until a layer is dropped from server 2, the receiver subscribes to server 3. This procedure continues until all the servers in the list are subscribed. Then the receiver enters the cycle phase.



T_J : Join-timer
T_D : Detection-timer
Add : Add a layer
Drop : Drop a layer
Relax : Decrease T_J multiplicatively
L : packet loss
F : our layer is highest of recently added layers
R : our layer was recently added
E : at least 1 active join-experiment at layer higher than ours
L > T : loss rate exceeds threshold
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

Fig. 2. State machine of FRLM receivers

B. Cycle phase

When all the servers on the list have been tried, the receiver enters the cycle phase. At this time, the join-timers of all the state machines (one for each server) are restarted. Since the values of the T_J's are selected randomly, they will expire in different times. Whenever a T_J expires, a join-experiment is carried out in the corresponding server, and this server is said to be active. All T_J's of the passive servers are stopped. This is because the join- experiment is still in progress, the layer is not added successfully, so there should not be another active join-experiment.

The state machines corresponding to the passive servers monitor the packet loss for determining a drop decision. For the active server, further join-experiments are conducted. When a join-experiment fails, we treat the link to that server as fully occupied and the offending layer is dropped. All the T_J's are rescheduled again by choosing a random value. The above cycle procedures repeat again.

Whenever a join-experiment fails, the corresponding T_J is backed off, same as RLM [1] and FRLM [6], and the state machine of that server enters D state. Hence, when all the T_J's restart, the likely available servers are having smaller values of T_J. This property helps FRLM-MS to have a shorter convergence time, by not choosing a server randomly to add a layer. Also, this property reduces the number of unnecessary join-experiments if all the bottleneck links are fully occupied.

C. Recovery phase

A receiver enters recovery phase only when an intermediate layer is dropped, i.e. not the highest subscribed layer. Due to the use of cumulative layers, if the data of an intermediate layer is missing, all the higher layers become useless. Hence, the layer has to be recovered as soon as possible.

We choose the server providing the highest subscribed layer to recover the dropped intermediate layer. Recall that the next

server is subscribed when the previous server is not available to provide a higher layer. So it is believed that the link of the server providing the highest subscribed layer should have enough resources for adding an extra layer.

When an intermediate layer is to be dropped, the drop request is sent. And at the same time, the add request is also sent to the server providing the highest subscribed layer. If bandwidth is not enough to provide an extra layer, the join-experiment fails. Here, the highest subscribed layer is dropped, instead of the intermediate layer. Since it is the highest layer, dropping such layer does not make other lower layers become useless. This recovers the dropped intermediate layer.

IV. Implementation of FRLM-MS

To implement FRLM-MS protocol, there are three main components to be considered: Server Location, Server Selection, and Path Diversity. They are indeed similar to that of a Content Delivery Network (CDN) [7, 8].

A. Server Location

There are four server placement algorithms presented in [7]:

Tree-based Algorithm – It assumes that the underlying topologies are trees, and models it as a dynamic programming problem. They divide a tree T into several small trees T_i, and show that the best way of placing $t > 1$ proxies in the tree T is to place t_i proxies the best way in each small tree T_i, where $\sum_i t_i = t$. However, this algorithm is shown to be not as good as the other algorithms.

Greedy Algorithm – In choosing M replicas among N potential sites, it chooses 1 replica at a time. In the first iteration, it chooses 1 site which yields the lowest cost among others. In the second iteration, it chooses the second site, with the site already picked, yields the lowest cost. The iteration loops until M replicas are chosen.

Random – As its name, it chooses M replicas among N potential sites randomly.

Hot Spot – This algorithm attempts to place replicas near the clients with the greatest load. It sorts the N potential sites according to the amount of traffic generated, and then selects the first M sites in the sorted list.

In FRLM-MS, since the receivers may join and leave the multicast group at any time, we can not predict the receivers' location. Therefore, we employ random server placement algorithm for locating the multicast servers for our simulations in the next section.

B. Server Selection

In FRLM-MS, a receiver has to obtain a list of servers, in the order of "nearest server first". The technique is similar to that used in DNS-based client redirection [9]. The client requests the DNS to translate a hostname into an IP address. Then the DNS contacts the servers on its own to test for the response time from the servers, and reply to the client with the IP address of the fastest server.

In FRLM-MS, the mechanism of DNS-based redirection is slightly modified. Instead of replying the client with the fastest server's IP address, the DNS orders the IP addresses of the servers into a list with the fastest server's IP address first. Then it replies the client with this sorted list.

C. Path Diversity

In CDN, a technique of multiple description (MD) coding [8] is employed, which decodes the original signal into multiple bit-streams. Each bit-stream can be used to reproduce the original signal, and the quality of the decoded signal improves with the number of descriptions that are correctly received. Hence, MD is similar to cumulative layering except it contains redundant information for each description to reproduce the original signal.

To utilize the benefit of MD, different descriptions should be transmitted through different network paths. This is limited by the incoming degree of a receiver. In FRLM-MS, the performance of layered coding can be enhanced if the technique of path diversity is employed. However, since we aim at keeping the current best-effort IP network unchanged, there is no guarantee that the multiple paths from a receiver to a set of servers are disjoint. In order to simplify the FLRM-MS protocol, we just employ the shortest-path routing mechanism in the next section.

V. Performance evaluations

In this section, the performance of the FRLM-MS protocol is evaluated by simulations using the LBNL network simulator *ns* [10]. There are two sets of simulations. First, we investigate the performance of FRLM-MS, including layer subscription and congestion control behavior. Second, we compare the performance of FRLM-MS with the single-server approach, in which FRLM [6] is chosen for comparison.

A. Performance of FRLM-MS

Fig. 3 shows a simple topology for investigating the operation of FRLM-MS. The receiver is labeled by R. S1, S2, S3 and S4 are the four multicast servers, each consists of the same 30 layers of data, with 20kbps/layer. CBR_S is the congestion source, which transmits a constant-bit-rate (CBR) traffic of 40kbps to CBR_R. All the links are lossless, with bandwidth and delay specified, queue size is set to 20 packets and the simulation is run for 1000 seconds. All the parameters used for FRLM-MS are set to the same values of FRLM [6], as well as RLM [1], namely α=2, β=2/3, k_1=1, k_2=2, g_1=0.25, g_2=0.25, T_J^{min} =5sec and T_J^{max} =600sec.

We aim to investigate the performance of FRLM-MS with and without network congestion. Fig. 4 shows the layer subscription of receiver R without network congestion, i.e. CBR_S is shut off. When the receiver starts, it contacts the DNS

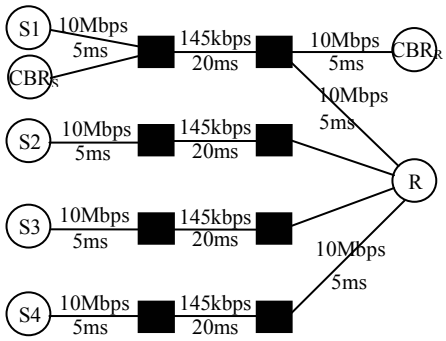
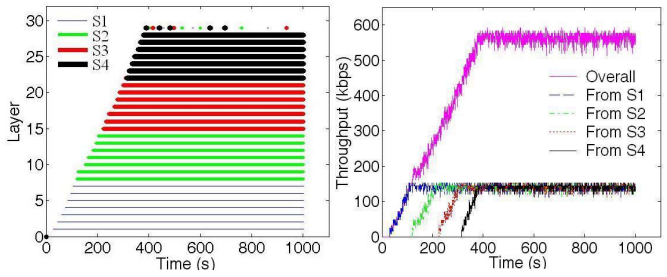


Fig. 3. Simulation topology



(a) Layer subscription (b) Throughput

Fig. 4. Layer subscription and throughput of R, no congestion

and receives a list "S1 S2 S3 S4". As expected in the "start-up phase", the receiver subscribes to layers 1-7 from S1, layers 8-14 from S2, layers 15-21 from S3, and layers 22-28 from S4.

The receiver is at its optimal layer subscription of layer 28. It then conducts a join-29 experiment. Obviously, it fails due to the fully utilized bottleneck link from R to S4. At this point, the receiver enters the cycle phase.

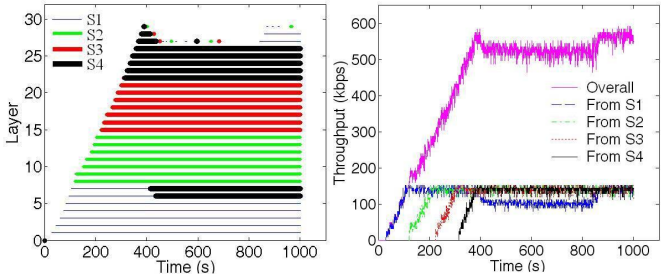
In the cycle phase, all T_J's are restarted at the same time. As shown in Fig. 4, S2's T_J expires first. T_J of S1, S3 and S4 are stopped. The receiver conducts a join-29 experiment in S2. By the same reason, it fails. Layer 29 is dropped, the T_J of S2 is backed off and the state machine of S2 enters D state. All T_J's are restarted again, and this time S3's T_J expires first. Again, the join-29 experiment fails in S3. S3's T_J is backed off and all T_J's are restarted. Due to the backoff of the join-timer, the frequency of join-29 experiment is reduced. And the receiver can stay at its optimal layer.

The result of the second simulation (i.e. with congestion) is shown in Fig. 5. The CBR_S starts at 400s and transmits at 40kbps, which causes network congestion in the bottleneck link from R to S1. Two layers, 6 and 7, must be dropped in order to solve the congestion. Since they are essential to the higher subscribed layers, the receiver enters recovery phase. At this moment, the receiver's highest subscribed layer, layer 28, is from S4, S4 is chosen to add the dropped layers (6 and 7). Unfortunately, adding these 2 layers causes congestion in the bottleneck link from R to S4. As a result, layers 27 and 28 are dropped instead.

In the presence of network congestion, we can see that the optimal number of subscribed layers is 26. The receiver drops to layer 26 within 30s, and then enters the cycle phase again. S3's T_J expires first. The receiver conducts a join-27 experiment at S3. Due to the limited bandwidth, it fails and S3's T_J is backed off again. We can see from Fig. 5a that the frequency of join-27 experiment is reduced due to the backoff of T_J.

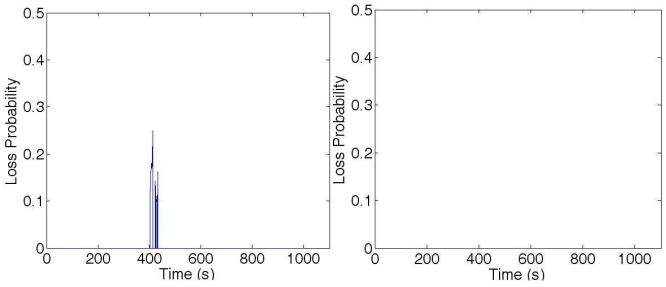
At t = 800s, CBR_S stops and the congestion disappears. When S1's T_J expires, the receiver can successfully add layer 27 from S1. It further adds layer 28 and conducts join-29 experiment at S1. Again, the frequency of join-29 experiment is reduced by the backoff of T_J.

Fig. 6 presents the packet loss probability in M state. We focus on the packet loss probability in M state but not the overall packet loss probability because a layer drop decision is made based on the former. This also provides insights on



(a) Layer subscription (b) Throughput

Fig. 5. Layer subscription and throughput of R, congestion



(a) S1 (b) S4

Fig. 6. Loss probability of R

persistent network congestion even when the packet loss probability is less than the threshold value used in M state.

When CBR_S starts, the packet loss probability of the path R-S1 ramps up immediately. Congestion is solved within a short period of time by dropping layers 6 and 7. These 2 layers are added from S4 again in the recovery phase. Layer 7 is dropped first, and hence is recovered first. Here, a single packet loss can cause the join-experiment to fail, and layer 28 (the highest subscribed layer) is dropped. This is the reason why there is no readings in Fig. 6b, as it does not enter M state. The same reason applies in recovering layer 6.

B. FRLM-MS vs FRLM

To compare the performance of FRLM-MS and FRLM, another simulation is conducted using the randomly generated topologies by BRITE [11]. The generated topology is input to the ns simulator [10] for simulations. Without loss of generality, the following parameter settings are used in the BRITE generator:

- Intra-AS bandwidth: 100-300kbps randomly
- Inter-AS bandwidth: 1Mbps
- 15 nodes within each AS
- Inter-AS degree: $m = k$-1 for k AS case
- Min-degree in AS: $m = 5$

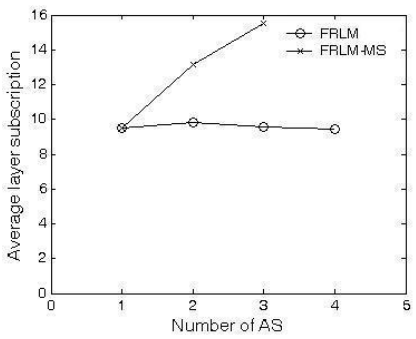


Fig. 7. Average layer subscription

Our idea is to create scenarios that the intra-AS bandwidth is limited, and the inter-AS bandwidth is abundant for taking advantages of multiple servers.

We select the node with highest degree in each AS to be the multicast server. The reason is that the higher is the degree of the server, the less is the chance the bottleneck link appears at the outputs of the server. We randomly choose 3 nodes from the remaining nodes in each AS to be the receivers.

Fig. 7 presents the result of the simulations., in which the average statistics from 100 random topologies for each number of AS are collected. It shows that when the number of ASes increases, i.e. the number of servers increases, the average number of layer subscription of FRLM-MS also increases. From Fig. 7, we can see that FRLM cannot benefit from the increased number of servers. The layer subscription of FRLM remains at a level of about 9.5 layers.

As a final remark, a FRLM-MS receiver subscribes different layers from different servers. Therefore, the servers should be synchronized to, say, within tens of milliseconds. This can be easily achieved via the Internet. Although the propagation delays from different servers vary, it can be tolerated by adjusting the playback buffer size at each receiver.

VI. Conclusion

In this paper, we proposed an efficient algorithm for layered multicast transmission with the use of multiple servers, called Fast-response Receiver-driven Layered Multicast with Multiple Servers (FRLM-MS). The performance of FRLM-MS was evaluated by simulations. We showed that FRLM-MS can perform well during congestions by subscribing to a larger number of layers from different servers.

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] 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.

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

[4] J. Liu, B. Li, and Y. Q. Zhang, "An End-to-End Adaptation Protocol for Layered Video Multicast Using Optimal Rate Allocation," *Multimedia, IEEE Transactions*, vol. 6, issue: 1, Feb. 2004, pp. 87-102.

[5] Lechang Cheng, and Ito, M.R., "Receiver-driven Layered Multicast Using Active Networks," *Proc. of ICME'03*, vol. 1, 6-9 July 2003, pp. 501-504.

[6] H. S. Chiu, and K. Yeung, "Fast-response Receiver- driven Layered Multicast," *Proc. of IEEE ISCC'04*, vol. 2, 28 June – 1 July 2004, pp. 1032-1037.

[7] I. Lazar, and W. Terrill, "Exploring Content Delivery Networking," *IT Pro*, July/August 2001, pp. 47-49.

[8] J. Apostolopoulos, T. Wong, W. T. Tan, and S. Wee, "On Multiple Description Streaming with CDN," *Proc. of IEEE INFOCOM 2002*, vol. 3, 23-27 June 2002, pp. 1736-1745.

[9] V. Cardellini, M. Colajanni, and P. S. Yu, "Dynamic Load Balancing on Web-server Systems," *IEEE Internet Computing*, vol. 3, issue: 3, May – June 1999, pp. 28-39.

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

[11] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," *2001 Proc., Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 15-18 Aug 2001, pp. 346- 353.