SARLM: Sender-adaptive & Receiver-driven Layered Multicasting for Scalable Video

Qiang Ni, Qian Zhang, and Wenwu Zhu*

Microsoft Research, China
5F, Bejing Sigma Center, No. 49, Zhichun Road
Haidian District, Beijing 100080, P. R. China

ABSTRACT

This paper introduces a new Sender-Adaptive & Receiver-driven Layered Multicasting (SARLM) for layered or scalable video. The sender first codes the video sequence into multiple data streams by using scalable source codec with FEC and then sends each stream as a separate multicast group. In the meantime, the receivers dynamically estimate the available bandwidth using packet pair approach followed by join or leave groups independently according to their network condition. Moreover, the receivers send sparse feedbacks containing statistical information about the receivers' network status. According to the analysis of feedback messages, the sender adjusts sending parameters. Simulation results show that higher network throughput and better video quality can be achieved using SARLM.

I. INTRODUCTION

Layered multicast is a promising technique for saving resource when delivering scalable or layered video over network. One of the main challenges for layered multicast is how to support heterogeneous receivers. To cope with network heterogeneity, receiver-driven, sender-driven, and sender-adaptive & receiver-driven layered multicast protocols are emerged recently.

Receiver-driven Layered Multicast is first proposed by McCanne et al. [1]. Legout introduces a packet pair receiver-driven layered multicast protocol (PLM) [2], which uses packet pair to infer the available bandwidth at the bottleneck for receivers to decide which layer to subscribe. Both of them had not addressed how to adapt sending rate according to the varying network condition. Therein fixed sending rate is provided for each layer, which leads to low utilization when background traffic varies. Vickers et al. introduced a source adaptive multi-layered multicast algorithm (SAMM) where the source uses congestion feedback to adjust the number of generated layers and the bit rate of each layer [3]. However, feedback mergers are needed in the network, which is not easy to be implemented in current Internet, especially from application's capability. Cheung et al. proposed destination set grouping (DSG) algorithm, which combines both receiver-driven and sender-driven approaches [4]. However, by using independent streams rather than layered streams, it leads to an inefficient usage of the bandwidth.

For sender-adaptive & receiver-driven layered multicast, the main challenges on the receiver side are how to estimate the varying network status, e.g., available bandwidth, how to manage the join/leave behaviors on the receiver side, and how to send feedback to avoid so-called feedback implosion.

There are three types of bandwidth measurement techniques, which are TCP throughput model, pathchar, and packet pair, respectively [5]. The former two are highly depending on network RTT (Round Trip Time) estimation, which cannot be computed as easily as loss rate on the receiver side in multicast environment [6]. M. Handley and Sally Floyd suggested using an external clock source such as GPS or NTP to achieve clock synchronization for calculating RTT [7]. However, that is rather hard to implement in current Internet. As for packet pair approach, since RTT calculation is not needed in this mechanism, it seems to be a promising method to estimate bandwidth in multicast environment.

To avoid feedback implosion, several solutions had been proposed based on hierarchy [3], parameterized [7], and randomly delayed timers [8]. The hierarchy scheme needs router's special support, which can't be implemented by application in the current Internet. The parameterized scheme is easy to implement but can't represent the global property of all receivers. Nonnenmacher et al. suggested a truncated exponentially distributed timer to achieve efficient feedback suppression. However, a multicast feedback channel or multicast-emulated unicast feedback channel is needed for every receiver, which leads to additional overhead for the network.

For efficient scalable video transmission, we proposed a sender-adaptive & receiver-driven layered multicast scheme called SARLM [9]. This paper mainly focuses on the behaviors on the receiver side. The available bandwidth of each receiver is estimated by using Receiver Based Packet Pair (RBPP) algorithm. Meanwhile, the receiver manages the join/leave behaviors according to the estimated network status. In addition, a Gamma-distributed random timer is introduced for generating scalable feedback that is sent through a unicast feedback channel from the receiver to the sender.

II. THE ARCHITECTURE OF SARLM FOR VIDEO MULTICASTING

Our proposed SARLM is composed of an adaptive layered video sender, layered or scalable video receivers, and multicast-capable routers. There are no further requirements for the network infrastructure.

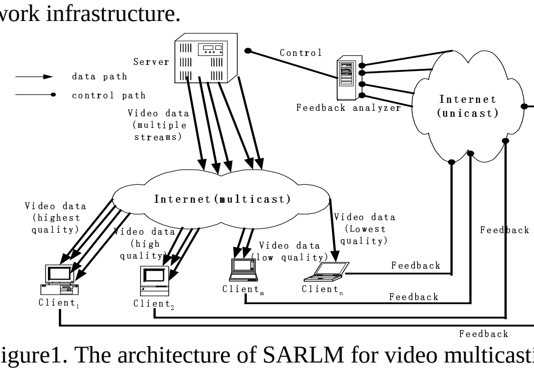


Figure1. The architecture of SARLM for video multicasting.

* This work is performed while he is a visiting student at Microsoft Research China. He is from Department of Electronics & Information Engineering, Huazhong University of Science and Technology, P. R. China.

Figure 1 depicts the end-to-end architecture of our scheme. The sender first codes the video sequence into multiple data streams and then sends each stream as a separate multicast group. In the meanwhile, the receivers join or leave groups independently according to their network condition. More specifically, SARLM works as follows:

(1) The sender codes the video sequence into multiple data streams by using scalable source codec and forward error correction (FEC), and then sends each stream as a separate multicast group over Internet. The stream of each multicast group includes either source data in different video layers or corresponding parity check data.

(2) Each receiver dynamically monitors the varying network condition, estimates available bandwidth, and decides whether to join a higher layer, stay at or leave the current layer.

(3) The sender (or feedback analyzer) periodically multicasts a feedback request to all receivers.

(4) Once receiving the sender's feedback request, each receiver generates a Gamma distributed random timer, and selectively sends feedback in a short period.

(5) Based on all feedback messages, the sender explicitly classifies the receivers into different groups.

(6) Sender adjusts the sending rate and parity check level for each multicast group based on Rate-Distortion relation of source video.

The sender is in charge of forming multiple multicast groups from the video sequence and determining the parameters for each multicast group. This part of work had been discussed in [9]. Besides that, the other key issues of SARLM include estimating the varying network condition, managing join/leave behaviors, and sending scalable feedback to the sender. In the following sections, we will discuss those issues in detail.

III. AVAILABLE BANDWIDTH ESTIMATION AND JOIN/LEAVE BEHAVIOR MANAGEMENT

As mentioned above, packet pair is a promising approach to estimate bandwidth in multicast environment. Packet pair relies on the fact that if two packets are queued next to each other at the bottleneck link, one is t second apart from the other. t is calculated as

$$t = {PacketSize} / {B_{bottleneck}} \text{ ,} \tag{1}$$

where *PacketSize* is the size of the second packet, and $B_{\text{bottleneck}}$ is the bottleneck bandwidth as illustrated in Figure 2.

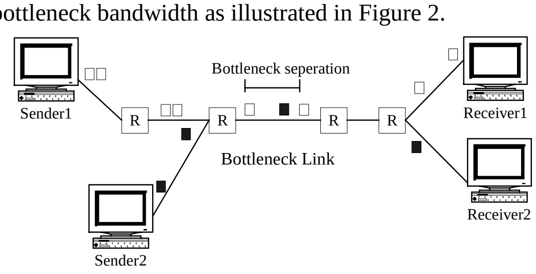


Figure 2. Packet pair probing scheme.

Receiver Based Packet Pair (RBPP) and Sender Based Packet Pair (SBPP) are two typical types of packet pair algorithms, which measure the bottleneck bandwidth by arrival times of packets and acks, respectively. In general, the results of SBPP may be highly inaccurate during congestion. Considering the multicast application, we use RBPP algorithm in this work to measure the bottleneck bandwidth for each receiver.

The main problem with the packet pair algorithm is how to filter out the noise caused by time compressed and extended packets. Many algorithms such as Measured Bandwidth Filtering (MBF), Packet Bunch Mode (PBM) and Potential Bandwidth Filtering (PBF) have been proposed for filtering noise [5, 10]. By finding the point of the greatest density in the distribution of bandwidth estimation, we can get the correct estimation from the samples.

In our approach, we send three packet-pair probing packets (back-to-back, which is denoted as a burst) for all the multicast groups every second. Taking the channel efficiency into account, we do not use the particular control packets for probing as in [10], instead, only data packets are used in our work. All the multicast groups of the same session are considered as a single flow. We set a flag in the header of the first packet of a burst so as to make receiver easy to identify the beginning of a packet-pair burst. Each receiver can estimate the available bandwidth by

$$B_{available} = \frac{PacketSize_2 + PacketSize_3}{t_{a3} - t_{a1}} \text{ ,} \tag{2}$$

where $PacketSize_2$ *and* $PacketSize_3$ are the sizes of the second packet and the third packet, respectively, and t_{a1}, t_{a3} are the arrival times of the first and the third packets, respectively.

The receiver controls the join/leave behaviors based on the above estimated bandwidth. Let $B_{\text{available}}$ be the available bandwidth estimated by the packet-pair method and B_{n} be the current bandwidth obtained with n cumulative layers. Then, we have

$$B_n = \sum_{i=1}^{n} Rate_i \text{ ,} \tag{3}$$

where $Rate_i$ is the sending rate of the i^{th} layer. The join and leave policy can be described as follows.

- *Leave Policy*

Each receiver drops a layer instantly when its estimated bandwidth is lower than the cumulative bandwidth of the current subscribed layer (i.e., $B_{\text{available}} < B_{\text{n}}$).

- *Join Policy*

The receiver tries to join a new group if the minimum estimated bandwidth is greater than the cumulative bandwidth of the current subscribed group for the last 1 second. (i.e., $\min(B_{\text{available}}) > B_{\text{n}}$).

As described in [11], synchronization among all the receivers in the same downstream bottleneck link is important for multicast pruning mechanism. Our scheme gets both join and leave synchronization in the following:

- each receiver behind the same bottleneck will receive the same probing packet pair at the same time (according to the distance between the receivers and the bottleneck) and will drop layers at the same time when estimated bandwidth is lower than the current subscribed group.
- if the receivers don' t join the session at the same time, late joiner will be resynchronized when the first drop occurs.

This resynchronization avoids the problems due to clock drift as well.

IV. SCALABLE FEEDBACK MECHANISM

As discussed in Section I, Nonnenmacher et al. introduced a truncated exponentially distributed timer to achieve feedback suppression. However, their scheme requires a multicast feedback channel or multicast-emulated unicast feedback channel for every receiver to suppress others' feedback.

In order to increase suppression ratio and lower feedback latency, we introduce a Gamma-distributed timer from 0 to T for each receiver, which outperforms other distributed timers when the number of users is very large. The density of our truncated Gamma distributed timer is defined as:

$$f_{Z_i}(z_i)=\begin{cases}\dfrac{1}{(e^{\lambda}-1)}\times\alpha\times\dfrac{\lambda}{T^{\alpha}}\times z_i^{\alpha-1}\times e^{\frac{\lambda}{T^{\alpha}}z_i^{\alpha}} & ,0\le z_i\le T\\ 0 & ,otherwise\end{cases},\quad (4)$$

where T is a fixed interval of our feedback, λ and α are factors related to the number of receivers. The sender updates the receivers' distribution after a certain interval, which is equal to T in this work.

Our proposed scalable feedback scheme can be divided into the following four steps:

(1) The sender multicasts a feedback request packet (I, λ, α, T) to the receivers, where I is the identification number of the feedback interval, λ and α are parameters of the gamma-distributed timer, and T is the interval size.

(2) Receiver i generates a gamma distributed random timer $z_i \sim [0, T]$ upon receiving the request (I, λ, α, T) (See Figure 3). To suppress the other receivers' feedbacks, only the receivers who get the timer between $(0, 0+c)$ can send feedback, where c is the delay between the receiver and the sender. When the timer z_i expires, receiver i sends feedback messages, *FBMs* (z_i, T_{λ}), to the sender, where T_{λ} is estimated available bandwidth obtained in Section 3.

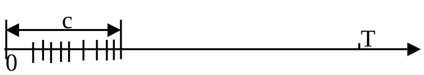


Figure 3. Gamma distributed timer setting.

(3) Upon receiving the FBMs, the sender estimates the number of receivers given by

$$\hat{R}=\frac{X}{F_Z(z)}, \quad (5)$$

where X is the number of received feedback messages, and z is obtained from the receivers' FBMs. To achieve a fast convergence and a reasonably smooth estimation, an exponential weighted moving average is used as follows:

$$\hat{R}_{I,\alpha}=\begin{cases}1 & I=1\\ (1-\alpha)\hat{R}_I+\alpha\hat{R}_{I-1,\alpha} & I>1\end{cases}, \quad (6)$$

where α is set to 0.8 in our work.

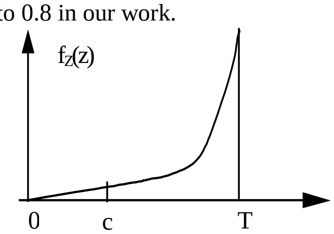


Figure 4. Estimation of the numbers of receivers.

(4) The sender calculates the new λ and α for the next feedback request based on $\hat{R}$, required feedback latency, and the expected number of FBMs. Mathematically,

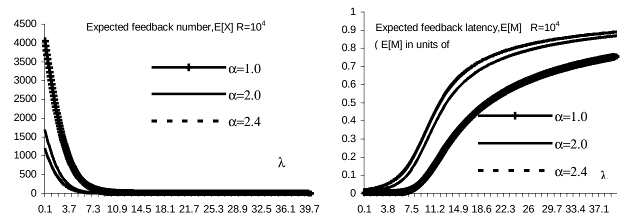
$$\lambda=1.1\cdot\ln\hat{R}+0.8, \quad (7)$$

and

$$\alpha=\frac{\ln[\frac{1}{\lambda}\ln[\frac{N(e^{\lambda}-1)+\hat{R}}{\hat{R}}]]}{\ln(c/T)}, \quad (8)$$

where N is a desired number of feedback messages. It can control the feedback bandwidth and suppress feedback implosion.

Figure 5(a) shows the suppression performance of gamma-distributed timer with *T=10c*. It can be seen from the Figure 5(a) that through dynamically tuning λ and α on the sender side, we can achieve a small number of expected feedback number, e.g., *E[X]<20*, for large scale of receivers, e.g., up to 10^4 receivers. In this way, the feedback implosion is avoided. With scalable feedback scheme, the feedback latency depends on λ and α, which are illustrated in Figure 5(b).



(a) (b)

Figure 5. Performance of scalable feedback scheme.
(a) Suppression performance of gamma-distributed timer.
(b) Performance of feedback latency.

V. SIMULATION RESUTLS

This simulation is to demonstrate effectiveness of SARLM. In this simulation we tested: (1) our proposed SARLM; (2) PLM that proposed by Legout.

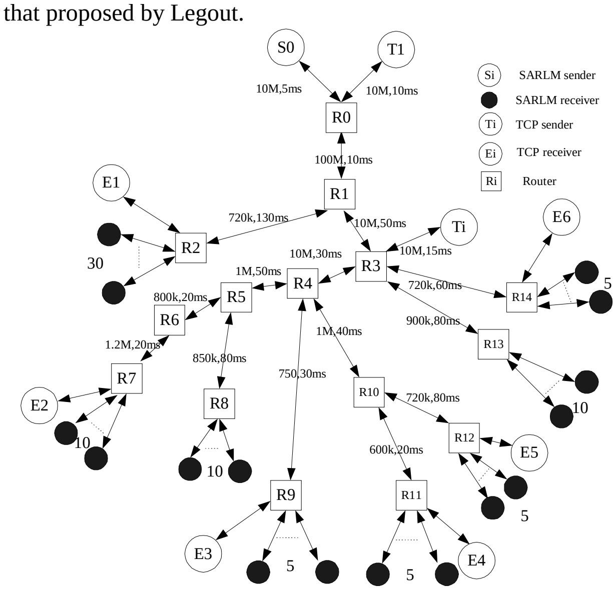


Figure 6. Simulation topology.

We use network simulator (NS-2) in this simulation [12]. The network topology used is shown in Figure 6. The multicast routing protocol is DVMRP. All the queues are fair queuing and each router has a queue size of 20 packets. We simulate a dynamic network condition by generating several FTP flows as background traffic. There are one sender, eighty receivers, and six varying background traffic in this topology. All local area access bandwidth is 10M. In this simulation, the number of expected feedback messages from the receivers is set to 35. The parameter *c* is set to 200ms and *T* is set to 10s. The testing video sequence *Foreman* is coded into three layers, with the corresponding bitrate as 192kbps, 384kbps and 1024kbps, respectively.

Figure 7 shows the throughput comparison using SARLM and PLM. Notice that here the throughput is the cumulative one of all the receivers. It can be seen from Figure 7 that SARLM achieves higher throughput than PLM. More specifically, the mean throughput of SARLM is 40% higher than that of PLM. This is because the varying network bandwidth can be estimated in SARLM, and the sending rate of each multicast group is adjusted according to the receivers' status.

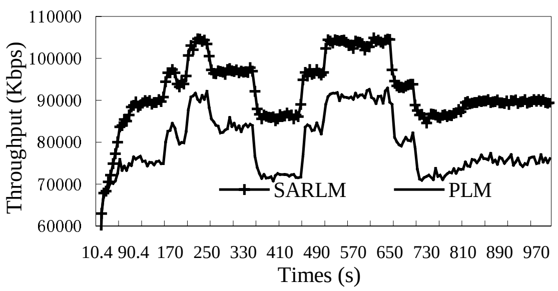


Figure 7. Throughput comparisons of all the receivers.

As for the individual receiver, Figure 8 depicts the throughput comparison between SARLM and PLM for a receiver of group 7. It can be seen that SARLM obtains higher throughput than PLM. Notice that there is no background FTP traffic during 140s to 200s and 850s to 1000s. This bandwidth variation can be measured by SARLM, and sender adjusts the parameters of multicast group accordingly. Thus, rather higher throughput can be obtained during those periods of time.

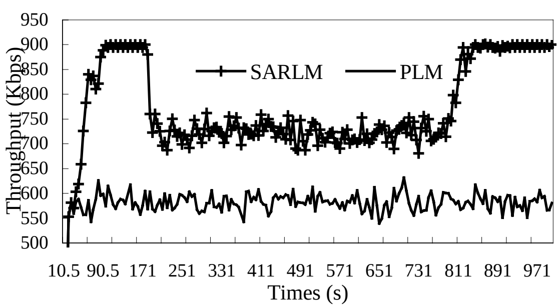


Figure 8. Throughput comparison of a receiver in group 7.

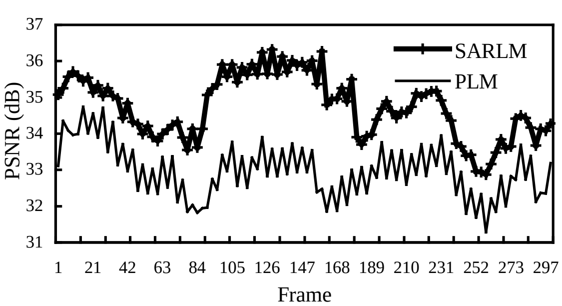


Figure 9. PSNR comparison of a receiver in group 7.

Figure 9 depicts the PSNR comparison between SARLM and PLM for a receiver of group 7. It can be seen that SARLM achieves higher video quality than PLM. More specifically, the average PSNR obtained by this receiver using SARLM is 33.58; while the average PSNR obtained by this receiver using PLM is 32.08.

For the other receivers in the other groups, similar comparison results can be obtained.

VI. Conclusions

This paper presents a novel architecture of Sender-Adaptive & Receiver-driven Layered Multicast (SARLM) Scheme for layered video. By receiver-based Packet Pair bandwidth estimation algorithm, the receivers can join or leave groups according to their network conditions. In the meanwhile, the receivers send scalable feedback information about the network conditions to the sender. By effective classification, the sender can adjust sending parameters according to the analysis of the feedback messages.

Currently we adjust the sending rate only through available bandwidth. The future work will focus on the source rate and parity-check rate determinations based on the Rate-Distortion relation of the video codec.

References

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast", *ACM SIGCOMM'96*, 1996.

[2] A. Legout, "PLM: Fast convergence for cumulative layered multicast transmission scheme". *Proceedings of ACM SIGMETRICS 2000*, 2000.

[3] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive Multicast of Multi-Layered Video: Rate-Based and Credit-Based Approaches," *Proc. of IEEE INFOCOM ' 98*, 1998.

[4] S. Y. Cheung, M. H. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution", *Proc. of IEEE INFOCOM*, 1996.

[5] K. Lai, M. Baker, "Measuring Bandwidth", *IEEE INFOCOM*, 1999.

[6] T. Turletti, S.F. Parisis, and J. Bolot, "Experiments with a layered transmission scheme over Internet", *INRIA Research Report* No 3296, Nov. 1997.

[7] M. Handley, S. Floyd, "*Strawman Specification for TCP Friendly (Reliable) Multicast Congestion Control (TFMCC)*", http://www.east.isi.edu/RMRG/newindex.html.

[8] J. Nonnenmacher, E. W.Biersack, "Scalable feedback for large groups", *IEEE/ACM transactions on Networking*, Vol. 3, June 1999.

[9] Q. Guo, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "A Sender-Adaptive & Receiver-Driven Layered Multicast Scheme for Video Over Internet", *IEEE ISCAS 2001*.

[10] S. Keshav, "Packet pair flow congestion control", *IEEE/ACM transactions on Networking*, Feb. 1995.

[11] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer", *IEEE INFOCOM'98*, 1998.

[12] "The Network Simulator (NS)," http://www-mash.cs.berkeley.edu/ns/ns.html.