APPLICATION LEVEL SELECTIVE DROP FOR LAYERED VIDEO OVER MULTICAST NETWORKS

Qiang Liu, Jenq-Neng Hwang

Dept. of Electrical Engineering, Box #352500
University of Washington, Seattle, WA 98195
{liuq, hwang}@ee.washington.edu

ABSTRACT

This paper presents an approach of router management that is easy to deploy and can improve the performance of existing layered video schemes. The router is configured to selectively drop a packet of the same application instance from its queue when the network is congested, which may be caused by either network dynamic or failed join experiment. Compared with uniform drop, this application level selective drop (ALSD) can increase the received video quality and provide a more stable subscription level for the narrow-bandwidth receivers competing the same bottleneck link with high-bandwidth receivers. We evaluate the promising performance of the proposed ALSD algorithm with multiple layered video schemes through network simulations.

1. INTRODUCTION

Cumulative, receiver-driven layered multicast video schemes [1,2] have been proposed to address the heterogeneity and scalability problems of the Internet. In such a layered video system, the video source uses a layered scalable compression algorithm with a layered transmission scheme and the receiver tries to adapt to the dynamic network condition by joining/leaving a layer (i.e., joining and leaving a multicast group).

RLM [1] is the first receiver-driven cumulative layered multicast protocol. The behavior of RLM is determined by a state machine where transitions among the states are triggered by the expiration of timers (the join timer and the detection timer) or the detection of losses. RLM uses “*shared learning*” to scale with the number of receivers. RLC [2] is a TCP-friendly version of RLM. RLC is based on the source-generated periodic bursts for bandwidth inference and on synchronization points to scale with the number of receivers.

On the other hand, priority drop mechanism at the router is considered to be an alternative or a complement to provide a graceful degradation in the presence of packet loss. When congestion occurs, routers selectively discard less important information (i.e., low-priority packets) before more important information (i.e., high-priority packets). Bajaj, Breslau, and Shenker in [4] analyze the merits of uniform versus priority drop for transmission of layered video and concluded that the performance benefit of priority drop is smaller than expected (the maximum performance gain is about 36% as shown in [4]). Their paper specifically considers uniform drop and priority drop as alternatives to RLM [1] and compares the performances of these algorithms.

The authors in [1] argued against priority drop in two ways. Firstly, priority drop rewards poorly behaved users since the video quality doesn’t decrease when the requested rate exceeds the bottleneck. Secondly, under priority drop policy, the receivers may not take the benefit of "*shared learning*" and impair the scalability of the algorithm.

In this paper, application level selective drop (ALSD) is proposed to be used together with a layered multicast video scheme. ALSD is one kind of priority drop that the priority preference is only considered within a single application instance. More specifically, when a router decides to drop a packet, the router searches in the queue for a lower priority packet that belongs to the same application instance. If such a lower priority packet is found, it is dropped while the original target packet is saved; otherwise the original target packet is dropped. We believe that ALSD is easier to deploy than a universal priority drop and it can be adopted with most existing layered multicast video schemes to improve the received video quality.

The remainder of this paper is structured as follows. Section 2 describes the ALSD. Section 3 contends that ALSD can help, rather than make worse, a layered multicast video scheme. Section 4 shows the comparison simulation results of ALSD applied to various layered video schemes and Section 5 gives the conclusion of this paper.

2. APPLICATION LEVEL SELECTIVE DROP (ALSD)

It is difficult, if not impossible, to apply a single priority preference structure to all applications in a network. There are many kinds of applications and it is not clear how to assign priority to the packets belonging to different applications. An application will try to always assign higher priority to its own packets to protect them against other applications. This defeats the original intention to apply priority to the system.

However, it is feasible to assign priorities within a single application instance. The application is well aware that which packet is more important so it can assign proper priorities to different packets without the concern that the assignment may be adversely affected by other applications. At the same time, the router also needs to distinguish different applications when it starts to drop packets. There are some mechanisms for the router to distinguish whether two packets belong to the same application, such as using source IP and destination IP pair, or using source IP/port and destination IP/port all together. We use source IP and source port to distinguish an application when the destination IP is a multicast IP. By this way, all the video packets of different layers can be considered as from a single application and can be treated using the same priority preference structure.

By using such an ALSD, each application can define its own priority preference structure. We don't set the maximum priority level for the system. An application can use as many levels as necessary. To speed up the search process of the router, it is desirable to use a fixed number as the indicator for the lowest priority level. In this paper, we use "0" as the lowest priority level and "1" as the highest priority level and "2", "3" as the second, third highest priority level, and so on. There may be arbitrary number of priority levels in the system. When the router decides to drop a packet with priority "0", the packet is dropped directly without any search. All packets with other priority levels require the router to search for a lower priority packet with the lowest priority level in the queue.

3. APPLYING ALSD FOR LAYERED VIDEO

An important feature of ALSD is that the highest priority packet may still be dropped due to a failure to find a lower priority packet in the queue. When the highest priority packet happened to be the first application packet entering a full queue or when all the lower priority packets are already dropped due to heavy congestion, such a search failure may occur. To decrease the probability of dropping the highest priority packet, the application can interleave the packet sequence so that higher priority packets are located among lower priority packets [5]. However, drop of the highest priority packet due to very heavy congestion or small queue size cannot be avoided easily.

For a layered video scheme, there are multiple ways to assign priorities to different layers. A simple way is to assign "1" to the base layer and assign lower priorities to the higher layers (“2” for layer 2, “3” for layer 3, and so on), finally assign "0" to the highest layers. Of course, the application can apply more sophisticated priority assignment to the packets regardless to the layers. For simplicity, we use the simple way (1,2,3…,0) above to assign priorities for all layered video schemes in this paper.

We assume one application uses the same source IP and port for all different layers (i.e., all different multicast groups), so that all the packets of different layers can be considered as from a single application instance. We use ALSD *together* with a layered video scheme and compare the performance difference with and without ALSD. This is a different case as in [4], where priority drop and uniform drop are compared as alternatives with RLM. Because the nature of the Internet traffic is bursty and the network dynamic is unpredictable, a layered video scheme will benefit from the ALSD mechanism to better adapt to the changes of network condition and keep a better video quality during congestion period.

The ALSD is executed exactly within a single application instance, so it cannot improve the overall loss rate for an application. However, the ALSD can improve the distribution of packet loss among different layers. Since lower layers are assigned with higher priorities, the probability of dropping a lower layer packet is lower than dropping a higher layer packet. Under the same overall loss rate, such a loss distribution results in better video quality, especially during the join experiment and other congestion periods.

When the video encoding algorithm allow correlations among different layers, i.e., the higher layers data require the lower layer data to decode, the system will benefit more from ALSD since the lower layer is protected by ALSD mechanism.

Regarding to the concern of the authors of [1], when the requested rate is beyond the bottleneck capacity, packet loss will always be detected even by using priority drop. The packet loss can be used as an incentive to decrease the requested rate. ALSD may also fail to benefit from "*shared learning*". However, because the lower layers are protected during the join experiments (this is the same reason for ALSD’s failure to benefit from "*shared learning*"), the excessive join experiments cannot do much harm to the performance (video quality) of the overall system. We admit that applying ALSD with RLM may worsen the scalability by increasing the aggregate congestion time caused by excessive join experiments. However, ALSD can improve RLM on other issues such as interference among multiple join experiments and scalability can be improved by other mechanisms such as synchronization points used in RLC [2].

The ALSD can also help the narrow-bandwidth receivers competing the same bottleneck bandwidth with high-bandwidth receivers. With uniform drop, when the high-bandwidth receivers do join experiments at high layers and cause congestion to the common bottleneck link, the narrow-bandwidth receivers may also encounter packet loss and may take it as an indication to drop a layer or stop going up. This adversely impairs the received video quality of the narrow-bandwidth receivers and slows down their convergence to the optimal subscription levels. With ALSD, the join experiments at high layers have much lower probability to cause packet loss of lower layer packets. So the narrow-bandwidth receivers can have better video quality and have more stable subscription level during network dynamics.

4. SIMULATION RESULTS

This paper uses ns2 [3] network simulator to conduct the comparative network simulations. The loss distribution and performance comparison are mainly used to show the effectiveness of the ALSD. We assume each layer contributes equal value for the total performance and the performance function is given by

$$\sum_{l} \frac{r(l)}{b(l)} , \qquad (1)$$

where $r(l)$ is the received data of layer l and $b(l)$ is all the data sent in layer l .

This performance function describes what is the received video quality derived from the set of received packets. Because video quality is a subjective measure that depends in large part on human perception and on the underlying coding algorithm, the range of applicability of any particular performance function is limited. Since we are focusing on the performance comparison instead of an exact performance calculation, this simple performance function doesn’t affect our conclusion too much.

Due to limitation of space, this paper only shows part of our comparative results. More results will be presented in the conference.

4.1 RLM comparison simulations

We use the example topology as in ns implementation (Figure 1) to run the RLM simulations. Packet size is set to 1000 bytes and rates in each layer are (32, 64, 128, 256) Kbps. All the 3 receivers *R1*, *R2* and *R3* try to adapt to their corresponding network bottlenecks. The simulations run for 600 seconds and the first 200-second warm-up period is not included in the performance representation.

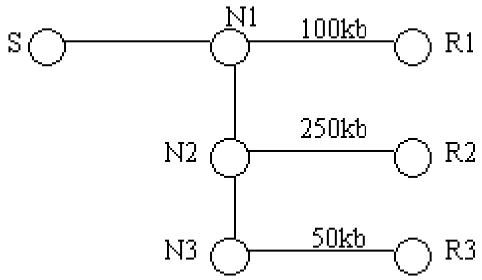

Figure 1. RLM simulation topology. All unmarked links are 10Mbps.

In the first set of simulations, we set the router queue size to 100. After applying ALSD, the subscription level process is not affected much since ALSD cannot help the total loss rate. However the loss distribution has considerable change as shown in Table 1.

	L 1	L 2	L 3	L 4	Total
R1	17%	17%	46%	N/A	25%
R1,ALSD	0%	0%	94%	87%	38%
R3	38%	51%	71%	N/A	54%
R3,ALSD	0%	6%	94%	N/A	42%

Table 1. Loss distribution for RLM simulation. Queue size = 100.

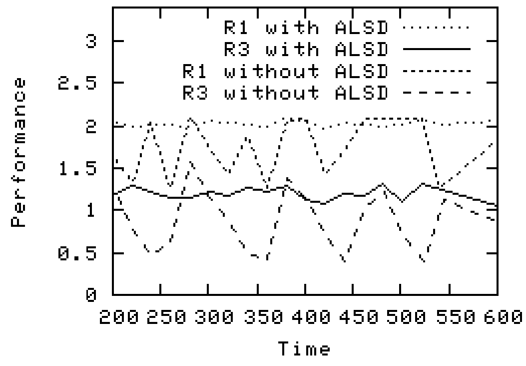

Figure 2. Performance for RLM simulation. Queue size = 100.

As shown in Table 1, ALSD can protect the more important packets while the total loss is not changed much. From the received packets, we can calculate the performance using Formula 1. We use 20 seconds as the interval to calculate the performance curve. The result is shown in Figure 2, which shows that ALSD can provide the receivers more stable and better video qualities.

The router queue size affects both RLM's performance and ALSD's performance. When the queue size is small, a failed join experiment will cause the queue overflow much faster. Thus the receiver can learn the network condition faster and more easily converge to the optimal subscription level. On the other hand, ALSD can only have limited effect when the router queue size is small because the probability to find a lower priority packet in a small queue is also small. We set the router queue size to 25 and repeat the simulations. The loss distribution is shown in Table 2. The total loss rate is much lower in this set of simulations because RLM adapts to the bottleneck faster and doesn't try the failed join experiments at higher layers too much.

	L 1	L 2	L 3	L 4	Total
R1	2%	2%	28%	N/A	4%
R1,ALSD	0%	0%	66%	N/A	3%
R3	2%	21%	N/A	N/A	5%
R3,ALSD	0%	34%	N/A	N/A	9%

Table 2. Loss distribution for RLM simulation. Queue size = 25.

As shown in Figure 3, ALSD doesn't improve the performance much when the queue size is 25 since RLM is quite optimal by itself in this case. However, because RLM still suffers from base layer packet loss as shown in Table 2 while ALSD doesn't lose base layer packets, some applications still can benefit from ALSD, especially when the coding algorithm relies on dependencies between layers (the simple performance function used in this paper doesn't capture the dependencies between layers).

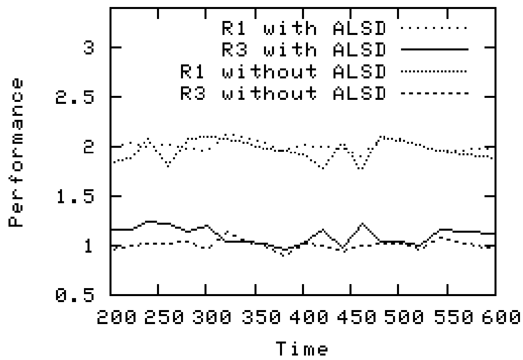

Figure 3. Performance for RLM simulation. Queue size = 25.

4.2 RLC comparison simulations

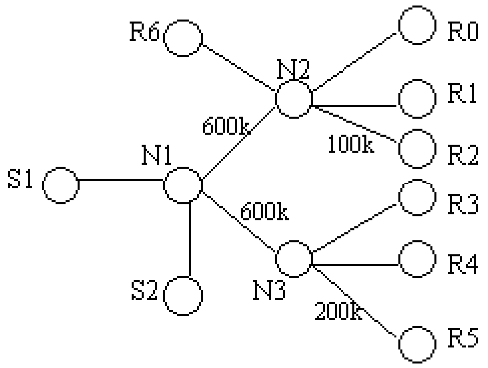

Figure 4. RLC simulation topology. All unmarked links are 10Mbps.

We also tried the ns implementation of RLC with the parameters as chosen by Vicisano in [2] (W = 8; P = 1). The rate of the base layer is 32Kbps and each enhancement layer has rate (32, 64, 128, 256, 512) Kbps. The packet size is 256 bytes and the queue size is 100. The simulation topology is as in Figure 4. Node *S1* is the source of layered multicast traffic while there are some competing TCP traffics from node *S2* to *R6*. Nodes *R0* to *R5* are multicast traffic receivers.

	L 1	L 2	L 3	L 4	Total
R1	9%	22%	N/A	N/A	11%
R1,ALSD	6%	27%	37%	N/A	8%
R2	9%	23%	N/A	N/A	11%
R2,ALSD	6%	28%	39%	N/A	8%
R5	3%	3%	4%	7%	4%
R5,ALSD	0%	0%	0%	16%	2%

Table 3. Loss distribution for RLC simulation. Queue size=100.

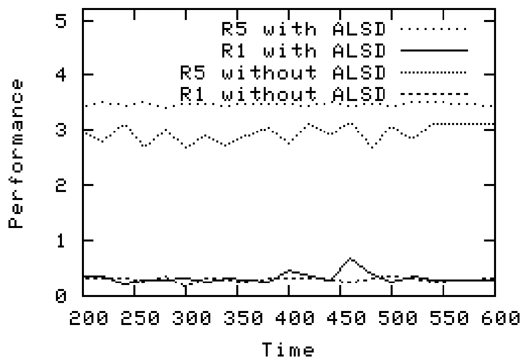

Figure 5. Performance for RLC simulation. Queue size = 100.

In all sets of simulations, each receiver can adapt to the network conditions and subscribe up to appropriate layers. The loss distributions of node *R1*, *R2*, *R5* are shown in Table 3 and the performance is shown in Figure 5. Because there are multiple TCP sessions from node *S2* to *R6*, the link *N1-N2* is heavily congested. This has two effects. Firstly, although *R1*'s last link is 10Mbps, it only subscribes to the same level as *R2* (because they share the same bottleneck *N1-N2*). Secondly, heavy congestion impairs the function of ALSD which has lower probability to successfully find a lower priority packet in the queue. So *R1* and *R2* encounter packet loss in layer 1 even with ALSD. And *R1*'s performance is not improved much by ALSD as shown in Figure 5. Instead, ALSD still helps the loss distribution and performance for R5 as achieved in RLM simulations.

4.3 Competition among heterogeneous receivers

As shown in Figure 4, narrow-bandwidth receiver *R5* is competing the bottleneck bandwidth (link *N1-N3*) with high-bandwidth receivers *R3* and *R4*. Our simulations show that ALSD can help the narrow-bandwidth receiver to have a more stable subscription level.

In a 600-second simulation, *R5* oscillates its subscription levels among layers 2, 3 and 4. However, when ALSD is applied, R5 seldom goes down to layer 2 and it mostly oscillates the subscription levels between layers 3 and 4. Table 4 shows the duration (in terms of seconds) that *R5* stays in each layer. As shown in the table, ALSD helps *R5* to have a more stable subscription level.

	L 1	L 2	L 3	L 4
R5	2s	141s	368s	89s
R5,ALSD	2s	2s	471s	125s

Table 4. Duration distribution for receiver R5.

5. CONCLUSION

This paper is devoted to evaluating the ALSD for layered video schemes. It is observed that ALSD can help a layered video scheme to have better performance by improving the loss distribution among the overall packet loss. The higher priority packets (i.e., the lower layer packets) are protected with a lower probability to be dropped by the routers. ALSD can also improve the performance of a narrow-bandwidth user competing the same bottleneck link with high-bandwidth users.

For the comparative studies, we used RLM [1] and RLC [2] algorithms as the examples of layered video schemes. The simulation results show that ALSD can improve the received video quality. We also notice that ALSD may still drop the higher priority packets, especially when the network is heavily congested or the router queue size is small.

6. REFERNCES

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast", In *Proceedings of ACM Sigcomm*, pp. 117-130, Palo Alto, California, August 1996.

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

[3] NS-2, The Network Simulator, version 2, http://www.isi.edu/nsnam/ns/.

[4] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus Priority Drop for Layered Video", In *SIGCOMM'98*, Vancouver, British Columbia, Canada, September 1998.

[5] H.F. Hsiao, Q. Liu, and J.N. Hwang, "Layered Video Over IP Networks By Using Selective Drop Routers", *invited special session talk in ISCAS*, Phoenix AZ, May 2002.