Achieving Inter-Session Fairness for Layered Video Multicast
Wei Kuang Lai and Chieh Ying Pan
Department of Computer Science and Engineering, National Sun Yat-Sen University,
Kaohsiung, No. 70, Lien Hai Rd., Taiwan

Abstract—The Internet is increasingly used to deliver multimedia
services. Since there are heterogeneous receivers and changing
network conditions, it has been proposed to use adaptive rate
control techniques such as layered video multicast to adjust the
video traffic according to the available Internet resources. A
problem of layered video multicast is unable to provide fair
bandwidth sharing between competing video sessions. In this
paper, we propose two schemes, layered video multicast with
congestion sensitivity and adaptive join-timer (LVMCA) and
layered video multicast with priority dropping (LVMPD), to
achieve inter-session fairness for layered video multicast. RLM,
layer-based congestion sensitivity, LVMCA, and LVMPD are
simulated and compared. Results show both schemes, especially
LVMPD, are fairer and have shorter convergence time than
other two schemes.

Key Words: Multicast, Video, Layer, Fairness, Session

I. INTRODUCTION

To handle heterogeneous network conditions [1] and
congestion while sending video streams, it has been proposed
to use adaptive rate control techniques to adjust the video
traffic according to the available Internet resources. There are
different approaches to the multicast transmissions of digital
videos, and the most scalable one is layered video multicast.
So in this paper, we focus on the layered video multicast.

A problem with proposed layered multicast algorithms,
such as RLM and LVMR which we will introduce later, is that
they do not provide fair bandwidth sharing between
competing video sessions [2]. If there are more than two
video sessions in progress, each receiver expects to receive
the videos with the quality that it is capable of receiving. In
reality, no matter whether the video session shares bandwidth
with other sessions or not, the receiver should get the quality
of videos commensurate with its share. This is called
inter-session fairness, and is what we will study in this paper.

Authors of both RLM [3] and LVMR [4][5] all point out
that their protocols do not provide inter-session fairness.
Some schemes are proposed to solve the fairness problem of
layered video multicasting, such as layer-based congestion
sensitivity, aggressive add-layer, top level random drop, and
drop to base layer. However, there are still different problems
that cause unfairness in those schemes as we will point out in
the next section. Our goal is to find better solutions for
inter-session fairness of layered video multicast.

The rest of the paper is organized as follows. Related works
of layered video multicast schemes are introduced in
Section 2. We first introduce RLM and LVMR. Both of them
can not achieve inter-session fairness. Then we introduce
recent schemes, which include layer-based congestion
sensitivity, aggressive add-layer, top level random drop, and
drop to base layer to improve inter-session fairness. In
Section 3, two schemes, layered video multicast with
layer-based congestion sensitivity and adaptive join-timer
(LVMCA) and layered video multicast with priority dropping
(LVMPD), are proposed to achieve inter-session fairness in
layered video multicast. Then four schemes, which include
RLM, Layer-based congestion sensitivity, and our two
proposed two schemes are simulated and compared in Section
4. Section 5 is conclusion.

II. LAYERED VIDEO MULTICAST SCHEMES

Layered video multicast schemes are receiver driven
protocols. That is, the source does nothing but transmitting
each layer of the video stream on a separate multicast group.
The protocol is actually run at each receiver, to add or drop a
layer by joining or leaving a multicast group. Each receiver
decides whether to add or drop a layer by monitoring network
congestion status and running the following simple control
loop [3]:

- When the network is congested, drop a layer.
- When there is spare capacity, add a layer.

With those schemes, the receiver would reach the maximal
level of subscription. The receiver adds layers until
congestion occurs, and backs off to a point below the
congestion.

A. RLM and LVMR [3][4][5]

McCanne, Jacobson, and Vetterli proposed RLM to
support rate adaptation for layered video transmissions over
IP multicast. In RLM, the receiver uses packet loss rate to
detect congestion. When there are sustained losses, it drops a
layer to reduce the congestion. When there has been no loss
for a period of time (join-timer \( T_J \)), it adds a layer to utilize
newly available bandwidth. This is called "join-experiment".
If the join-experiment results in congestion, it drops the new
layer and doubles the join-timer \( T_J \) for next join-experiment.
This is called the back-off action of join-timer. \( T_J \) would be
decreased if the receiver detects no packet loss for a period of
time (detection time \( T_D \) ) because there may be spare capacity.

The \( T_D \) is maintained by the receiver also. It is used to
detect successful join-experiments. If a join-experiment lasts
longer than the detection time \( T_D \) without congestion

The work is supported in part by NSC under contract number NSC-88-2213-E-110-0021.
occurring, we consider the join-experiment successful. \(T_D\) is also used to delay the layer drop after a packet loss is detected. If congestion lasts longer than \(T_D\), we can be sure that the network is in congestion status. If there are only few packets dropped for a very short time, it is not necessary to drop a layer.

LVMR proposed by Li, Paul, Pancha and Ammar is another protocol for distributing video using layered coding over the Internet. The behavior of receivers in LVMR is similar to that in RLM. The main contributions of LVMR are its error recovery and hierarchical rate control scheme [4].

We can find that the original rate control schemes (RLM and LVMR) in layered video multicast would not handle inter-session fairness when there are multiple sessions of videos competing for limited shared bandwidth. To achieve inter-session fairness, the high-level session should drop a layer and let low-level session join a layer. But it is hard to decide whether to drop a layer when packet losses occur. On one hand, we want to avoid video quality degradation caused by unnecessary layer drop. On the other hand, dropping a layer would allow other sessions to get reasonable amount of bandwidth, and avoid continuous congestion.

B. Rate Control Schemes for Supporting Inter-Session Fairness

In this subsection, we briefly discuss some schemes that can be applied to original receiver-driven rate control schemes to improve inter-session fairness. We first describe a scheme called layer-based congestion sensitivity [6]. As we discussed in the previous section, if there is no extra bandwidth, the high-level session must drop a layer to allow low-level session to join a layer. To ensure layer fairness in this scenario, we can let different video layers have different sensitivity to congestion. Two parameters were used in the original protocol:

- \(r\): the loss rate threshold for congestion – if the packet loss rate is above the threshold, then the receiver considers that the network is congested and prepares to drop a layer.
- \(t\): the congestion resilience time – this is similar to the parameter \(T_D\) in RLM. If the receiver detects packet loss, it has to wait for time \(t\). If the packet loss rate continues to be above the threshold \(r\) after time \(t\), the receiver drops a layer.

In this rate control scheme, the main idea is to adjust \(r\) and \(t\) to make higher layer drop easier. Sensitive functions \(f_t()\) and \(f_r()\) are defined as follows.

\[ r_i = f_t(i), 0 < f_t(i) < 1, \text{ where } i \text{ is the layer number,} \]
\[ t_i = f_r(i), 1 < f_r(i). \]

In this layer-based congestion sensitivity scheme, the functions are defined as follows.

\[ f_t(i) = R/i, \]
\[ f_r(i) = T/i, \]

where \(R\) and \(T\) are constants, and \(0 < R < 1, T \geq 1\).

There are several other schemes proposed. The aggressive add-layer scheme makes the add-layer experiment more aggressive to sustain congestion longer [6]. It would probably make other sessions to drop their top layers to allow the add-layer experiment to succeed. There is another less aggressive scheme called top-level random drop [6]. When a network is congested, the highest video layer is randomly dropped to allow the other video sessions to add new layers. The probability of random dropping should be related to the current video level to make higher layer drop more easily. In drop to base layer scheme, when a new session starts and the congestion is detected, the existing sessions drops all but the base layer so that new session and existing sessions can compete for shared bandwidth together from the base layer.

Apparently, there are problems in those schemes. Aggressive add-layer would make the situation worse if a high-level video session is aggressively adding a layer. Top level random drop won’t assure the fairness because of random dropping. There are probabilities that top layers are not dropped and low layers are dropped. And drop to base layer would obviously slows down the progress of converging to fairness since once a new sessions starts, all the other session would start from base layer again. It would also cause serious video quality degradation since all the receivers drop to the base layer no matter what video level it is receiving. In addition, for all schemes degradation of video quality is another problem. When congestion occurs, all layers might lose packets in those schemes.

In [6], the performance of top level random drop, aggressive add layer and drop to based layer schemes are all simulated and compared with layer-based sensitivity scheme. Overall, layer-based congestion sensitivity has the best performance in inter-session fairness [6]. So we simulate RLM (the basic scheme) and layer-based congestion sensitivity (the representative scheme for achieving inter-session fairness) for comparisons with our proposed schemes in later simulations.

III. LAYERED VIDEO MULTICAST WITH CONGESTION SENSITIVITY AND ADAPTIVE JOIN-TIMER (LVMDA) AND LAYERED VIDEO MULTICAST WITH PRIORITY DROPPING (LVMPD)

In the Section, we first propose one scheme, LVMDA, to reduce the convergence time (the time needed for all sessions to come to their steady reception levels) and the variations of video transmission rates after convergence. Then we propose another scheme, LVMPD, which drops a high-level video session instead of a low-level video session when congestion occurs, to further decrease the convergence time and the variations of transmission rates after convergence.

A. Layered Video Multicast with Congestion Sensitivity and Adaptive Join-Timer (LVMDA)

We modify the join-timer of layer-based congestion sensitivity. Main idea of the LVMDA scheme is to allow low-level video session to add a layer easier. In layer-based congestion sensitivity, a high-level session drops a layer faster than a low-level video session in congestion. In our scheme, we also allow low-layer session to be more active in adding a
Drop the top layer $i$ if
1. the packets loss rate $> r$ (loss rate threshold), and
2. the duration of congestion $> T_D$ (detection timer of level $i$)

Adding the layer $i+1$ when
the time the receiver remains uncontested (no packet loss) $> T_S$ (join-timer for layer $i$)

Fig. 1. The algorithm of layer-based congestion sensitivity with adaptive join-timer.

new layer. We adapt $T_J$ to video levels. We assign larger $T_J$ to high-level session and small $T_J$ to low-level session. This scheme would decrease the time to converge to fairness. It would make adding layer in high-level sessions harder, decreasing unnecessary congestion caused by the join-experiment of high-level sessions while there is no spare bandwidth for high-level sessions to add a layer. While there is spare capacity, the lower-level video session would add a layer easier and faster than the higher-level video session. Fig. 1 is the pseudo code of the layer-based congestion sensitivity with adaptive join-timer for a receiver at level $i$.

We define a function $T_J = f_J()$, which adjust the join timer with the video level, as follows. $f_J() = J + i$, where $J$ is a constant, and $i$ is the layer number. Similar to layer-based congestion sensitivity, the other two parameters are defined as follows:

$r_J = f_J() = R / i$, $T_D = f_J() = T / i$, where $R$ and $T$ are constants, and $0 < R < 1, T \geq 1$.

B. Layered Video Multicast with Priority Dropping (LVMPD)

In [7], RLM is compared with two schemes without layering:

- Priority Dropping: Different priorities are assigned to the I, B, and P frames in the MPEG video stream. More important frames such as I frames are assigned the higher priority and vice versa. In congestion, the frames with the lowest priority are dropped first.
- Uniform Dropping: In congestion, packets are dropped randomly without layering or priority dropping.

The studies only focus on overall performance of RLM, uniform dropping and priority dropping. The simulation results showed that “priority dropping achieved performance improvements of 50% to 100% over RLM in many settings” [7] while in [3], McCanne, Jacobson, and Vetterli claim that in dropping schemes (both priority and uniform dropping), “a greedy or naive user would likely request a rate far above the bottleneck rate, driving the network into a persistently congested state.” Therefore, it would waste unnecessary bandwidth before packets pass the bottleneck link and cause unnecessary congestion.

From these studies, we can see both advantages and disadvantages of priority-dropping scheme and layered video multicast schemes such as RLM and LVMR. The priority-dropping scheme has better performance, and

\begin{verbatim}
In sender
  packet.priority = packet.layer
In switch
  P: the total number of priority levels
  i: priority number
  packetcount(i): the number of packets with priority i in the queue
  when a packet is received:
    i = packet.priority
    packetcount(i)++
    sum += packetcount(i)
    for(j = P to 1, j--)
      if(packetcount(j) != 0) break
    packetcount(j) --
    delete first packet with priority j from queue
\end{verbatim}

Fig. 2. The algorithm of layered video multicast with priority dropping.

obviously it does not have the problem of fairness since the only packets dropped are packets with lower priority, no matter what sessions those packets belong to. However, it would waste bandwidth and cause congestion with improper requests from users. The layered video multicast schemes could solve the problem of the priority-dropping scheme, but it has the problem of fairness as we discussed in previous sections.

In LVMPD, we adapt the idea of sending video packets over various layers and combine it with priority dropping to improve video quality and achieve better inter-section fairness. We assign higher priority to lower layers and lower priority to high layers of video sessions. In a switch, there is a priority queue for each layer, when a new packet arrives, if the total number of packets has exceeded the queue limit of the switch, the nonempty priority queue with the lowest priority is chosen and the first packet from the selected queue is deleted. Fig. 2 is the pseudo code of priority dropping in senders and switches. The works performed in receivers are similar to other layer-based schemes and are not shown in Fig. 2.

IV. SIMULATION RESULTS AND COMPARISONS OF FOUR SCHEMES

We simulate the original RLM, layer-based congestion sensitivity, LVMRCA and LVMPD. The simulations are implemented with network simulator ns-2 [8]. We first performed two simulations with multiple video multicast sessions to show RLM can not achieve inter-session fairness.
We have two layered video multicast sessions and one shared link (Fig. 3). There are two receivers, $C_1$ and $C_2$, and two senders, $S_1$ and $S_2$. $S_1$ sends a video stream to $C_1$ (session 1) and $S_2$ sends a video stream to $C_2$ (session 2). Each video session can at most subscribe four layers which add up to 1.6 Mbps. All the simulation parameters are listed in Table 1.

In the following simulations, we set the bandwidth of the link $E$ to 1.7 Mbps. Hence, there is not enough for the base layer of session 2 to join at 50 second. Session 1 starts first, and $C_1$ joins all four layers soon as in the first experiment. At time 50 sec, session 2 starts, which causes congestion since there is not enough bandwidth for the base layer of session 2. Session 1 drops layer 4 (the highest layer) since packet losses occur. The behavior of changes in join-timer is similar for both simulations (Fig. 4). Both simulation results show that the RLM does not achieve inter-session fairness while there are multiple video sessions competing for limited bandwidth.

Fig. 5 shows the simulation result of layer-based congestion sensitivity. The parameters used for layer-based congestion sensitivity are listed in Table 1. In Fig. 5, we can see the improvement of inter-session fairness of the layer-based congestion sensitive scheme. The convergence time is 53.9 seconds. Here the "convergence time" means the time between a new session starts and the video sessions come to their steady reception levels. Because of the back-off action of $T_s$, session 1 rarely perform add-layer action, while session 2 kept performing adding-layer action.

The simulation result of LVMCA is shown in Fig. 6. We modify the join-time $T_j$ with video level. The parameters of LVMCA are also listed in Table 1. The convergence time is reduced to 39 seconds, which is better than layer-based congestion sensitivity. We can say that the improvements by layer-based congestion sensitivity with adapting $T_j$ are:

- Less convergence time: The convergence time is decreased because in this scheme, low-quality video sessions will add layers more often which would lead to the network congestion and force high-quality video sessions to drop layers. The result is a faster convergence time compared to layer-based congestion sensitivity.

- Steadiness: It is known that the congestion would affect the video quality. After converging to fairness status, longer joining frequency would avoid unnecessary and useless congestion. As we can see, after the video sessions reach to fairness, the joining frequency decreases. It would cause less congestion and smooth out the video quality.

### Table 1. The Simulation Parameters

<table>
<thead>
<tr>
<th>(in seconds)</th>
<th>RLM Layer-Based Congestion Sensitivity</th>
<th>LVMCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_D$</td>
<td>4 / (highest layer)</td>
<td>4 / (highest layer)</td>
</tr>
<tr>
<td>Congestion Threshold</td>
<td>0.06 / (highest Layer)</td>
<td>0.06 / (highest Layer)</td>
</tr>
<tr>
<td>$T_J$</td>
<td>5 / (highest Layer)</td>
<td>2.5 / (highest Layer)</td>
</tr>
</tbody>
</table>

Fig. 4. Simulation results of RLM (two video sessions competing for 2.1 Mbps of bandwidth.).

Fig. 5. Layer-based congestion sensitivity.

Fig. 6. Layered video multicast with congestion sensitivity and adaptive join-timer.

Fig. 7. Layered video multicast with priority dropping.

Fig. 7 shows the simulation result of LVMPD. With priority dropping scheme, the improvements on the convergence time and steadiness are obvious. At 68 sec, the second video session adds layer 2 successfully. Layer 3 of the video session 1 is dropped right after video session 2 adds layer 2. The convergence time is further reduced to 17.1 seconds. Priority dropping could also improve steadiness. After converged to fairness, both video sessions are sending packets up to layer 2 and both are very stable. If one of the sessions performs join-experiment by adding layer 3, packets of layer 2 in both video sessions will not be dropped since the packets of layer 2 have higher priority than those of layer 3.

With priority dropping, the packet loss of higher layers (lower priority) would be prior to that of lower layers if congestion occurs. The advantages are:

- Less convergence time – If the higher-level video session drops a layer before the lower one, then the lower-level session would have the chance to add a layer. With priority dropping, the higher-level video sessions would certainly drop a layer before lower ones since higher layer has lower priority. Hence, we expect the priority-dropping scheme would converge to fairness quicker and have less variations in source transmission rates.

- Steadiness and improvements of video quality – If we use priority dropping, only higher layer video streams would loss packets.

To facilitate our comparisons, we use "bandwidth ratio" to
represent the fairness between two video sessions:

\[ \text{Bandwidth Ratio} = \frac{\text{Bandwidth of video session 1}}{\text{Bandwidth of video session 2}} \]

If the value of bandwidth ratio is near 1, then two video sessions send video streams at the same video level. If the video sessions are keeping sending video streams at the same level, we can say that the video sessions have converged to fairness. On the other hand, if the value of bandwidth ratio is far above or below 1, then video sessions do not share the bandwidth fairly. Fig. 8 is the bandwidth ratios of four schemes throughout the simulation time. In Fig. 8, we can see the convergence status of the four schemes. LVMPD takes the shortest time to convergence, LVMCA has the second shortest convergence time, RLM with layer-based congestion sensitivity scheme is with the third shortest convergence time, and the original RLM needs the longest convergence time. We also can see the steadiness of each scheme after convergence.

After coming to convergence for a period of time, the join-experiment of LVMCA is fewer than layer-based congestion sensitivity because of adjusting \( T_r \) by video level in LVMCA. As for the LVMPD scheme, the join-experiment is hardly discovered in the graph since the packets of the newly added layer are mostly dropped during the join-experiment. The comparisons of convergence time and steadiness are also shows in Table 2. In Table 2, we calculate the average value and standard deviation of the bandwidth ratio each second after two sessions have converged to fairness for every scheme. Since RLM has never converged to steady reception levels until the end of simulation, we calculate it from the time that two sessions do not change their levels of subscriptions. Table 2 shows that both LVMCA and LVMPD, especially LVMPD, have shorter convergence times, smaller standard deviations, and fewer differences between maximum bandwidth ratio and minimum bandwidth ratio.

**TABLE 2. COMPARISONS OF FOUR RATE CONTROL SCHEMES.**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Convergence time (sec)</th>
<th>Standard deviation</th>
<th>min</th>
<th>max</th>
<th>max-min</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLM</td>
<td>0.161961</td>
<td>3.347820</td>
<td>1.393258</td>
<td>1.954568</td>
<td></td>
</tr>
<tr>
<td>Layer-based congestion sensitivity</td>
<td>53.9</td>
<td>0.005649</td>
<td>1.478261</td>
<td>0.639655</td>
<td>0.839605</td>
</tr>
<tr>
<td>LVMCA</td>
<td>39.7</td>
<td>0.004338</td>
<td>1.115789</td>
<td>0.642276</td>
<td>0.473513</td>
</tr>
<tr>
<td>LVMPD</td>
<td>17.1</td>
<td>0.002271</td>
<td>1.03</td>
<td>0.895455</td>
<td>0.134545</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

Multicasting video stream on the Internet with the existence of heterogeneity is becoming an important issue. RLM and LVMR have the fairness problem. Some layered video multicast schemes are proposed to solve this problem. But there is still fairness problem in these schemes. The priority-dropping scheme can solve the fairness problem. However, if users request a rate which is far more than the bandwidth available, the network would be consistently congested. Under that condition, many packets would be dropped before arriving at destinations. Hence, two schemes, LVMCA and LVMPD, are proposed in this paper to solve the inter-session fairness problem.

We found out that LVMCA could shorten the convergence time to fairness and leads to steadiness after the fairness status is reached compared to RLM and layer-based congestion sensitivity. We also found that with the LVMPD scheme, the video session could converge to fairness with the time even shorter than the LVMCA, and the rates of video streams are more steady after convergence. Also with LVMPD, if a video session adds a layer which causes congestion, those packets in the highest layer are dropped first. It has the minimum influence to those sessions which only subscribe low layers.

REFERENCES