Stability and Fairness Issues in Layered Multicast

R. Gopalakrishnan, James Griffioen, Gisli Hjálmtýsson, and Cormac J. Sreenan

Networking and Distributed Systems Research
AT&T Labs - Research
Florham Park, NJ 07932

Abstract

Layered multicast is a promising technique for broadcasting adaptive-quality video to heterogeneous receivers. Past evaluation of layered multicast protocols has focused on the effectiveness of determining the maximal number of layers that can be delivered to each receiver. Unfortunately, this is only a partial metric and does not capture all aspects of the "viewing experience".

This paper extends past analysis by investigating *stability* of layered multicast schemes and the related issues of *fairness* and *scalability*. Our study is motivated by the desire to transmit small-scale TV broadcasts (hundreds or thousands of viewers) over IP-based networks with heterogeneous receivers. Although users are aware of their physical bandwidth limitations, they will still expect TV-like characteristics of their video such as consistent quality (i.e., stability).

Our results shows that Receiver-Driven Layered Multicast (RLM) exhibits significant and persistent instability. Previous work also raises fairness as a potential problem with RLM. We show that with CBR traffic RLM can be arbitrarily unfair, but with VBR traffic, RLM provided better fairness than we anticipated. Our analysis of stability in RLM prompted us to question RLM's ability to select the optimum number layers for a non-shared access link. We show that RLM is very conservative in its choice resulting in low link utilization.

1 Introduction

The Internet is rapidly becoming the next global network infrastructure, supplanting special-purpose telephony and TV networks. Although originally designed for data transport, the Internet is increasingly being used to deliver multimedia services. This introduces new challenges, since media streams require higher levels of service quality and service stability than traditional data transport. Unlike telephone and TV networks, the Internet is a heterogeneous network, where receivers differ greatly in capabilities, link capacities, and network connectivity. Consequently, no single fixed bandwidth media stream will be optimal for all receivers. In addition, network load and traffic conditions (losses) can change dramatically, and rapidly. The ability of the Internet and its applications to cope with heterogeneity and adapt to changing network conditions has been a key factor in its success.

To accommodate heterogeneous receivers and adapt to congestion, it has been proposed to encode the media stream onto multiple layers and transmit each layer on its own multicast group [MJV96]. Receivers subscribe to as many layers as network conditions and receiver capabilities allow. Several schemes for layered multicast have been proposed [MJV96, WSS97, LPA98, BCZ98, VRC98, TPB97]. This paper focuses on the Receiver-Driven Layered Multicast (RLM) protocol [MJV96], but we believe our discussion and analysis are generally applicable to layered multicast protocols (section 6).

Past evaluation of layered multicast protocols have focused on their ability to identify the maximal number of layers that can be delivered to each receiver in the face of heterogeneity and sustained congestion [MJV96]. In [BBS98], a "utility model" was developed to compare RLM with schemes that use uniform and priority dropping at routers. Although these studies concluded that RLM is well suited for congestion and heterogeneity adaptation, they are just the first step toward evaluating the receiver's overall "viewing experience". Additional metrics are needed.

This paper extends past analysis by investigating the *stability* of layered multicast in RLM, and the related issues of *fairness* and *scalability*. Our study is motivated by the desire to transmit small-scale TV broadcast (hundreds or thousands of viewers) over IP-based networks to heterogeneous receivers. The model is that video is streamed from a broadcast source (or a small number of sources) to receivers. We are primarily interested in broadcast events that are large enough to benefit significantly from multicasting, but are still too small or too geographically sparse to be cost-effectively distributed via satellite or cable distribution networks currently used for TV broadcasts. Example broadcast events include distance learning and training sessions (*e.g.*, televised university classes), special events (*e.g.*, concerts, lectures or sporting events such as a European soccer match), and focused-community events (*e.g.*, large conferences or meetings). Although users are aware of their physical bandwidth limitations, they will still expect TV-like characteristics from their video such as consistent quality (i.e., stability). Frequent (observable) quality changes quickly become annoying. Furthermore, we expect multiple IP-based broadcast sessions will occur simultaneously and will compete for the network bandwidth, much like current TV networks broadcast multiple channels at the same time. As a result, it is important that bandwidth be allocated fairly among the broadcast sessions, where each session consists of all layers in a given media stream.

In this paper we report on our evaluation of how multiple RLM sessions interact and interfere with one another in an IP environment. Our initial objective was to investigate "multimedia-only" networks consisting solely of competing RLM sessions. Although the Internet carries other traffic types (namely TCP/UDP), we wanted to distinguish the interactions of RLM with itself from the interactions of RLM with arbitrary protocols. Moreover, early deployment of IP-based TV broadcasts will likely employ virtual private networks built on top of the Internet that will isolate video traffic from other traffic sources. In fact, the MBONE, which carries a variety of multicast sessions, consists of virtual links each with limited virtual bandwidth (typically 512 Kb/s or less). Even conventional cable TV networks may decide to support or convert to digital IP-based TV distribution over reserved channels as well as supporting other channels reserved for general IP traffic. The compression ratios of digital video versus analog video make this an enticing idea.

The following sections describe the results of our study. We simulated the performance of multiple RLM sessions competing for resources across a single bottleneck using the NS simulator from Lawrence Berkeley Labs. We then considered four measures of protocol quality: i) stability of the layered assignment and the protocol, ii) fairness of bandwidth allocation to sessions, iii) the impact of scalability on stability and fairness, and iv) utilization of access links as a metric of RLM's ability to adapt to heterogeneity. Stability is the most important of these as frequent changes in quality are annoying at best and intolerable at worst. Our results show that RLM exhibits significant and persistent instability even with favorable conditions. Lack of fairness is a known issue with RLM [MJV96, BBS98, MVJ99], although prior work does not quantify the extent of unfairness. Our results show that RLM exhibits arbitrary unfairness, and shows strong dependency between the bandwidth acquired by a session and the relative arrival order of sessions for CBR traffic. Utilization is of primary importance on access links where there is limited or no sharing. We use this as a measure of RLM's ability to adapt to heterogeneity. Our studies show that RLM utilizes such links very poorly.

2 RLM

The Receiver-driven Layered Multicast protocol (RLM) [MJV96] describes an approach for multicast delivery of layered video to heterogeneous receivers. The source encodes the video signal onto multiple discrete layers, where each layer incrementally refines the layer below it and is transmitted on a separate IP multicast group. A receiver selectively subscribes to as many layers as its access bandwidth and network conditions will permit. Thus RLM can cope with bandwidth heterogeneity and can adapt to changing congestion conditions. RLM is designed to exploit existing IP multicast capabilities, and does not require any new mechanisms within the network. Moreover the RLM protocol is run only at receivers and is transparent to senders.

When RLM detects sustained losses it drops a layer in an attempt to reduce the congestion. To learn of (newly) available bandwidth, RLM periodically conducts a *join-experiment* that probes the network by adding the next layer. If the join-experiment produces congestion, RLM concludes the bandwidth is not available, drops the new layer, and doubles the time before the next join-experiment. Each receiver also maintains a *detection-timer*, T_D, that estimates the amount of time between joining a layer and the subsequent onset of congestion. T_D is used to detect congestion or the lack of congestion (*i.e.*, successful join-experiments).

RLM includes optimizations to speed up convergence and enhance stability within a session. Receivers announce their join-experiments to other receivers in the session. The join announcement prevents other receivers from joining the same or higher layers during the experiment. However, a receiver that wishes to join a lower layer is allowed to do so. By announcing the experiments, all receivers learn of failed experiments. When they know a join is in progress and begin to observe losses, they conclude the join-experiment failed and exponentially back-off their join timer for the failed layer.

If a receiver has been stable for some time, it becomes more conservative about dropping a layer. When congestion arises, it enters a "hysteresis" state for one detection-time period (T_D) and then transitions to a "measurement" state. If the congestion abates by the end of the measurement period, the receiver does not drop a layer but instead returns to the stable state. If the congestion persists and the measured loss rate exceeds a threshold (25% for the RLM implementation in NS) it drops its highest layer. The delays in the various states are meant to dampen state transitions and prevent thrashing behavior.

2.1 Evaluation Environment

The original RLM paper [MJV96] investigated basic protocol properties such as latency, scalability and convergence rate within a single session. Bajaj *et. al.* [BBS98] considered multiple sessions, but focused on RLM's congestion adaptation mechanism compared to network-based mechanisms such as priority and uniform dropping rather than focusing on the interaction between sessions. Our work builds on and extends the simulation results presented in the above papers by analyzing properties of RLM when multiple RLM sessions share a bottleneck link. Before we present our results, we describe the simulation model we used to perform our experiments.

Our study is based on the simulation model used in [BBS98]. The simulation topology is depicted in Figure 1 and is designed to examine a shared bottleneck link. In each experiment there are n concurrent sessions, each having a single source and a single receiver. Each source has a dedicated 10 Mb/s link to the router at the head of the bottleneck link. The routers use FIFO scheduling with a tail drop policy. The bottleneck capacity is $n * 500$ Kb/s. All the receivers are downstream of the bottleneck link. Each source transmits a layered video session consisting of 6 layers. The base layer sends at a rate of 32 Kb/s, with the rate doubling with each subsequent layer. Thus the total bandwidth of 4 layers is 480 Kb/s, and hence the bottleneck link can accommodate 4 layers per source. Our results were obtained from simulations using *NS version 2* and the RLM protocol code used by the authors of the original RLM papers.

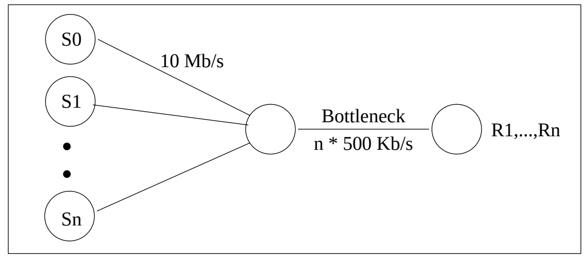

Figure 1: Simulation Topology

We used the source model described in [BBS98], with both CBR and VBR traffic. For the base layer, traffic is generated over 1 second intervals. In each interval n packets are transmitted, where n is chosen independently from the following random distribution: $n = 1$ with probability $1 - 1/P$, and $n = PA + 1 - P$ with probability $1/P$. A is the average number of packets per interval and is chosen to be four 1000-byte packets in our experiments. P is used to regulate the burstiness of the traffic and represents its peak to mean ratio. For $P = 1$ the above model produces CBR traffic. As P increases, traffic becomes more bursty. We use $P = 3$ and $P = 5$ for VBR experiments. Others [RT99] have commonly observed peak to mean ratios in the range of 2 to 10. The n packets are transmitted in a single burst, starting at a random time (uniformly distributed) within the interval. For each higher layer l the interval is broken into 2^l subintervals, and n packets are sent in one burst at a random time in each of these subintervals correlated across layers.

In several of the graphs presented here, we show a single example run to illustrate RLM's behavior and service quality as viewed by a receiver over time. Although behavior varies from run to run, we performed many simulations to verify that the examples shown here are representative. A more complete analysis can be found in [GGHS99].

To ground our work we started by reproducing the results in [BBS98] and [MJV96]. Using the parameters in the two papers we were able to reproduce and verify their results. The following sections build on the previous analysis, evaluating the issues of stability, scalability, fairness, and heterogeneity.

3 Stability

An important metric for evaluating video distribution protocols and end-user experience is the *stability* of the service quality. Ideally the user will not observe any noticable changes in the video quality for the duration of the media stream (*i.e.*, the video quality remains stable). The control mechanism in layered multicast schemes that manages service quality is the layer adaptation algorithm. Unfortunately, adding or dropping layers to react to changes in network load can result in sudden changes in video quality. The specific change in perceived quality depends on the encoding scheme being used. For example, if a spatial model such as MPEG-2's [Com93] *spatial scalability mode* is used to encode the layers, adding/removing a layer changes the size/resolution of the image. Most people would agree this is unacceptable. Temporal encoding schemes such as those used in [Com93, LPA98, BCZ98] typically result in substantial changes in frame rate that dictate how smooth or jerky the video appears. In short, the observed change in video quality depends on the encoding scheme, but changes in subscription level typically results in a noticeable fluctuation in video quality [1]. In some cases, the receiver application may be able to dampen the effects of level changes, making them less annoying. Therefore, the degree to which level fluctuations negatively affect quality depends on the user, the session and the application displaying the video.

Note that minor quality changes such as sporadic packet losses or short-term level changes are unavoidable. This can be caused by random transient congestion in the best-effort network. Alternatively, it can be caused by RLM's join-experiments that probe the network to learn if additional bandwidth is available. In a best-effort network, subscribing to a layer does not guarantee that the receiver will get every packet from that layer. Even if the subscription level is not changing, transient network congestion may cause short-term loses and delays that can affect the quality of the video. These transient losses or packet delays may be masked by the application via frame interpolation techniques and buffering respectively. Similarly, short-term changes in subscription level such as those caused by failed join-experiments may be masked at the application level. These are unavoidable and part of the protocol and thus do not constitute protocol instability (assuming we ignore the separate but related issues of join/leave latencies and overheads). The primary concern of our study was longer-term level changes that result in prolonged quality degradation or improvement that cannot be hidden or masked by the application. If these unmaskable, and thus perceivable, changes in subscription level occur frequently, the viewer will quickly become annoyed.

[1] One possible exception is the use of *ThinStreams* as proposed in [WSS97], but even that is dependent on the size of the layers and the encoding scheme.

3.1 Stability Experiments

There are several factors that have the potential to affect the stability of sessions through a bottleneck link. Factors such as the number of sessions, the burstiness of the traffic, the start time of new sessions, the number of receivers per session, the encoding/layering scheme, and the termination time of existing sessions can all play a role in stability. The following explores the first three factors and their affects on stability.

We measure stability as the frequency at which the subscription level at a receiver changes. Because the rate at which RLM adds or drops layers is a function of the congestion on the network, we began our analysis of stability in RLM by examining its performance using three different traffic sources; namely constant bit rate (CBR) sources, variable bit rate sources with P=3 (VBR-3), and variable bit rate sources with P=5 (VBR-5). These sources are defined in [BBS98] and summarized in section 2.1. Our intuition was that the more bursty the traffic source, the greater the fluctuation in network load would be, thereby increasing the probability that RLM will add/drop layers frequently. Because many encoding schemes produce VBR video sources, this could be a significant problem. On the other hand, as the number of sessions increases, the variation in network load should diminish because of statistical multiplexing, resulting in better protocol stability. We explore both of these hypothesis in the following sections (sections 3.1.1 and 3.1.2 respectively).

Before presenting our simulation results, recall that the RLM protocol causes receivers who have not recently added a layer to ignore congestion for one detection-timeout period before taking any action. Following the detection-timeout period, the receivers enter a measurement state where they monitor the loss-rate for another detection period. Only if the loss rate exceeds a predefined threshold will the receiver drop a layer. In other words, receivers that have been at the same level for some period of time will not react quickly to observed congestion. In theory, this slowness to react should enhance the protocol's stability.

3.1.1 Stability of Competing Sessions

To evaluate the stability of RLM for varying traffic types we study the interaction of two sessions competing for the bandwidth of a shared bottleneck link. We performed three experiments; one for each traffic class: CBR, VBR-3 and VBR-5. We then recorded the subscription level of each session over time and computed the number of level changes that occurred. To avoid “minor” changes, each point on the subscription level graphs reflects the average subscription level over a five second interval. Figures 2 - 4 show the results from the three tests. Each figure plots (a) the subscription level changes over time, and (b) the number of changes that occur during each 5-second interval.

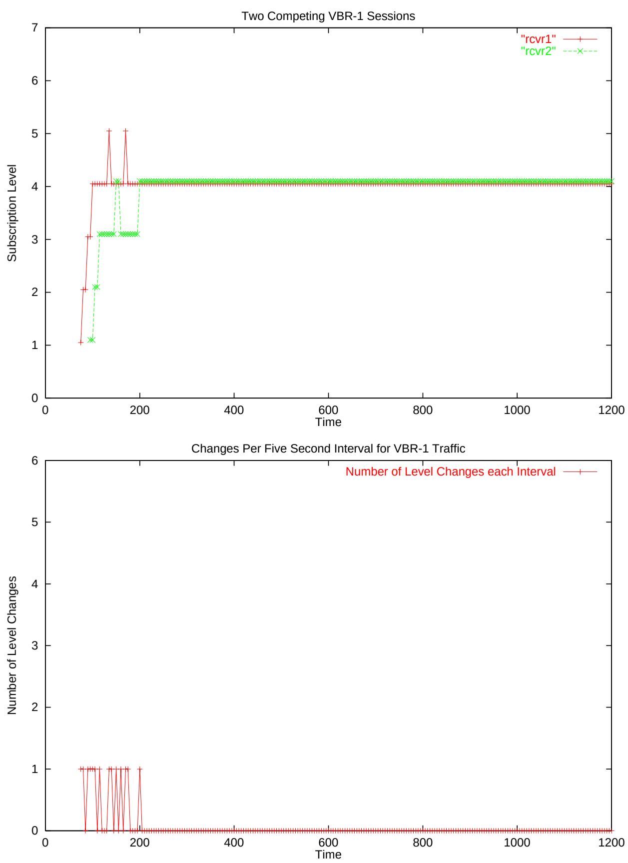

Figure 2: Subscription level stability for CBR traffic. Figure (a) on the left shows how subscription level changes over time. Figure (b) on the right shows the number of changes that occurred during each five second interval.

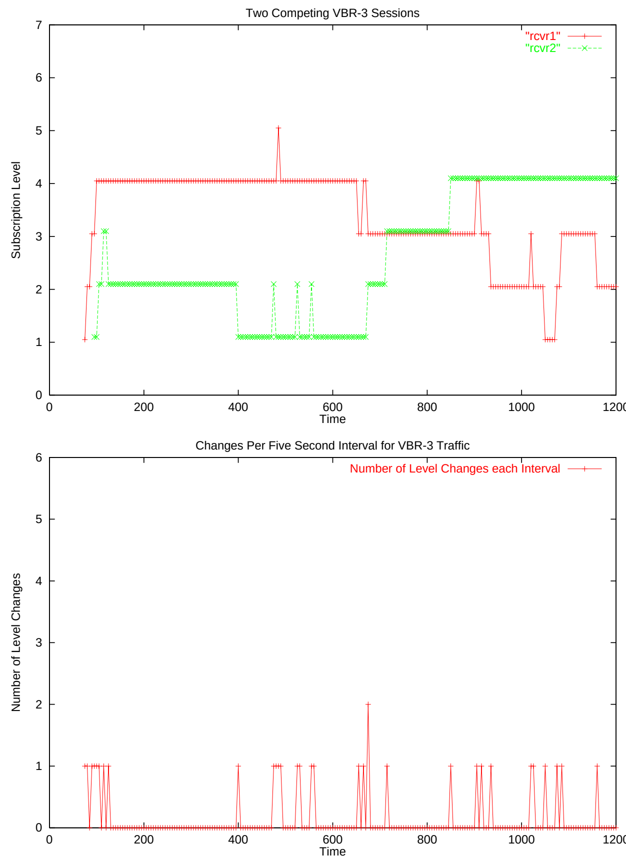

Figure 3: How subscription level changes over time for VBR-3 traffic. Figure (a) on the left shows how subscription level changes over time. Figure (b) on the right shows the number of changes that occurred during each five second interval.

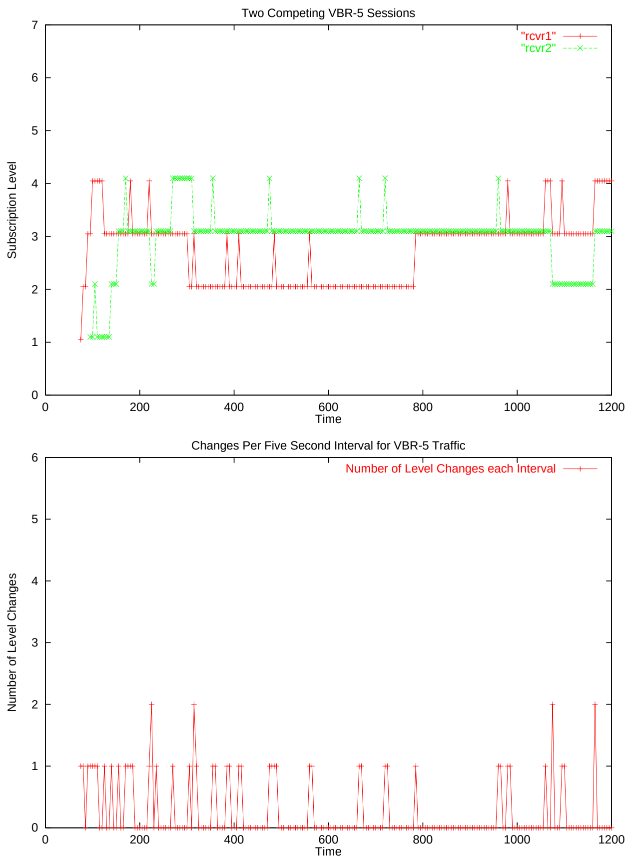

Figure 4: How subscription level changes over time for VBR-5 traffic. Figure (a) on the left shows how subscription level changes over time. Figure (b) on the right shows the number of changes that occurred during each five second interval.

As expected, the two CBR sessions converge rapidly to a fair allocation - four levels each - and then remain stable (see figure 2a). The same does not hold for the VBR sources. Both VBR-3 and VBR-5 experience many level changes throughout the simulation causing receivers to observe a wide range of video quality. Note that the VBR-3 test shows a period in which the video quality becomes four times better in less than a four minute window. Although the VBR-5 test in figure 4a does not show it, we have seen similar quality swings with VBR-5.

Another interesting phenomena is how the receiver's quality "flip-flops" over time. This occurs in both the VBR-3 and VBR-5 tests. In the VBR-3 test, receiver 1 initially receives four layers while receiver 2 only receives two layers (e.g. time 400-650). As time progresses the two trade places with receiver 2 receiving four layers and receiver 1 only getting two layers (e.g. time 920-1050). In between (at time 800) the system is running at the "fair" allocation of 3 layers for each receiver. From a stability standpoint, this is a significant protocol deficiency.

In a system that is becoming more stable over time, one expects an initial series of layer changes followed by longer and longer durations without any layer changes. Only the CBR exhibits this type of stabilization (see figure 2b). As evident from Figure 3b and Figure 4b, level changes continue to occur with VBR traffic for the duration of the simulation. In other words, there is no indication that the system will converge to a stable state. We have run much longer simulations and have not seen any signs of convergence. Also recall that we do not count join-experiments as instability, so each point represents a significant level shift lasting for at least five seconds.

Breakdown of Layer Changes							
Traffic Type	Total Up/Dn	Receiver 1			Receiver 2		
		Tot	Up	Dn	Tot	Up	Dn
CBR	12	7	5	2	5	4	1
VBR3	30	17	9	8	13	8	5
VBR5	49	27	15	12	22	12	10

Table 1: Number of subscription level changes seen by each receiver

Table 1 shows the breakdown of level changes per receiver for each of the traffic classes. From the users standpoint, there are 49 level changes that occur during the 1200 second VBR-5 test, with 27 occurring in receiver 1 and 22 occurring in receiver 2. This implies that, on average, receiver 1 and receiver 2 see a substantive change in video quality every 44 and 54 seconds respectively. For most users this would be considered unacceptable. If we assume that level increases can be integrated gradually by the application, then only level decreases introduce noticeable changes. Even then, receiver 1 and receiver 2, on average, see level drops every 100 and 120 seconds respectively.

Figure 5 compares the stability of CBR, VBR-3, and VBR-5 by plotting the cumulative number of changes per receiver that occurred over time. The slope of the curve indicates the rate of change, or the instability. From the figure we clearly see a marked increase in the rate of change as variability increases. Except for the transient start-up period the three curves appear in strict order of variability, CBR, VBR-3 and VBR-5.

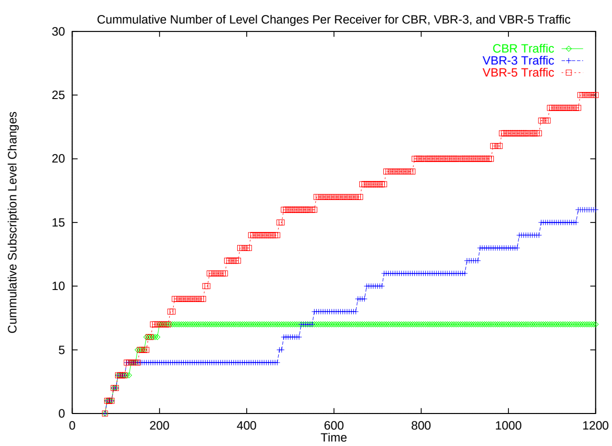

Figure 5: Stability Comparison: the cumulative number of level changes per receiver over time for traffic types CBR, VBR-3, and VBR-5.

3.1.2 Scalability

The previous section showed that RLM does not provide stability for VBR traffic, despite the fact that it is designed to ignore transient congestion. However, the principles of multiplexing have the potential to mitigate the instability affects caused by VBR traffic. Large-scale systems may not be as susceptible to VBR-induced instability because of the increased level of resource sharing between sessions.

To evaluate whether stability improves as system size increases, we ran simulations with two, four, eight, and sixteen simultaneous VBR sessions. In all cases we use VBR-5 type sources. To maintain system ratios, the network bandwidth was increased in relation to the number of sessions.

Figures 6 and 7 shows the subscription level over time for tests involving 2, 4, 8, and 16 independent VBR-5 sessions. Again, to remove the affect of short-term level changes, the subscription level is reported as an average over five second intervals. To make the graphs more readable, the subscription level for each session is offset by 0.05 times the session level. For example, session 3 subscribed to level 2 would be plotted at 2.15 rather than 2. Each graph illustrates the same stability problems found in the small-system case (2 sessions); namely that subscription levels are unstable, experience large quality swings, and flip-flop.

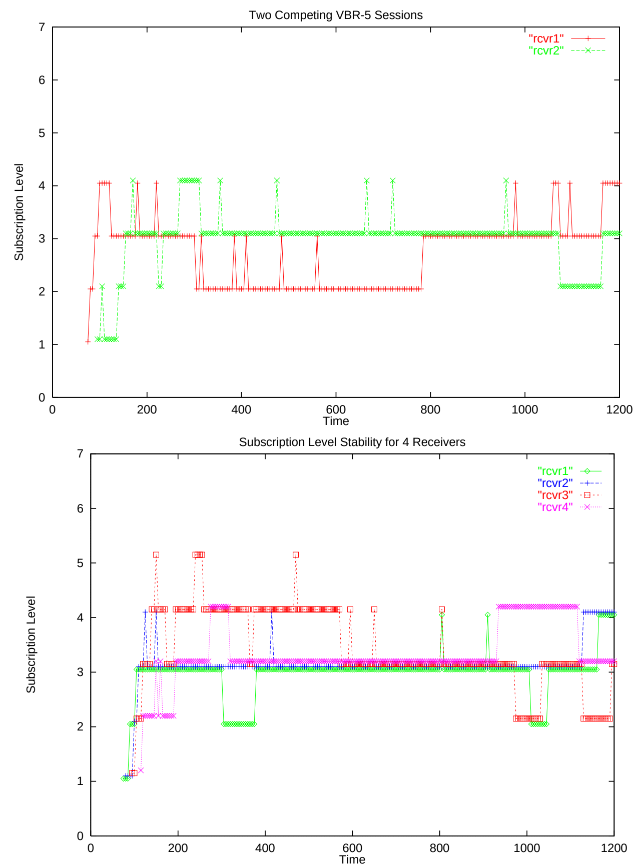

Figure 6: Subscription level stability for VBR-5 traffic for 2 and 4 simultaneous sessions.

Figure 8 shows the cumulative number of level changes for the 4, 8, and 16 session tests. Each is normalized to show the cumulative number of level changes per receiver. As the system size increase, the rate at which sessions change levels does not improve much,

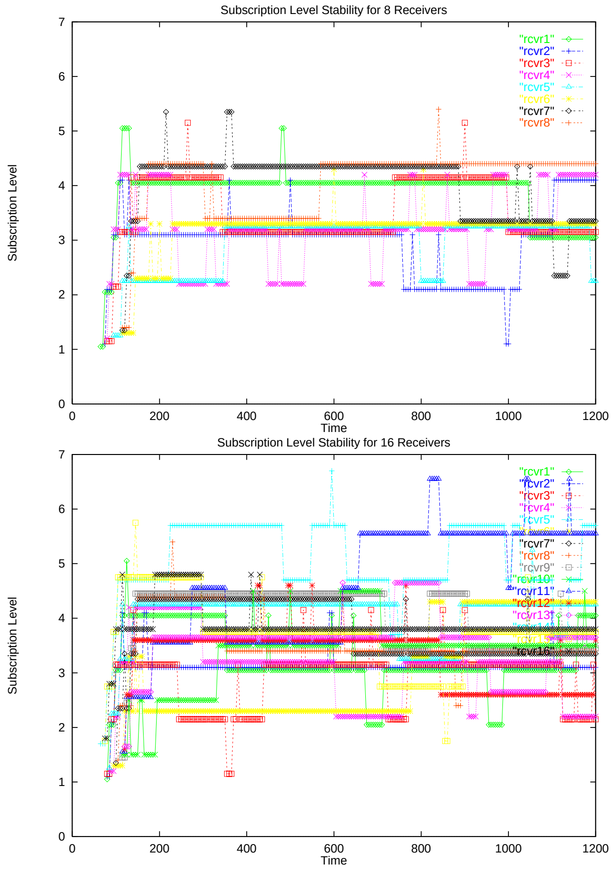

Figure 7: Subscription level stability for VBR-5 traffic for 8 and 16 simultaneous sessions.

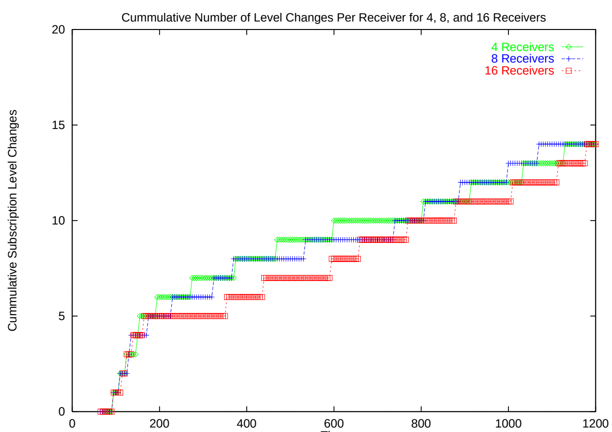

Figure 8: Stability Comparison: the average number of layer changes per receiver (cumulative) over time for different numbers of sessions.

implying that larger-scale multiplexing is unable to mitigate the effects of the VBR traffic.

4 Fairness

Past work on RLM [MJV96, BBS98] did not analyze interaction between sessions, but noted that RLM does not ensure fairness. This does not necessarily imply that RLM is grossly unfair. We set out to quantify fairness in RLM, analyzing bandwidth allocation among competing sessions and exploring how the protocol mechanisms impact fairness.

We define *fair* allocation as equal allocation to all sessions sharing a bottleneck link. Clearly other fairness policies are possible. Given our fairness policy the goal is to avoid a situation where one session is receiving up to the highest layer (e.g., HDTV quality service) while another session is only receiving the base-layer (e.g., fuzzy black/white service). From a fairness standpoint, both sessions should receive the same number of layers (e.g., standard VHS quality service). Clearly some receivers will receive better or worse performance depending on the capacity of their bottleneck link. However, all sessions that share the same (primary) bottleneck should receive an equal slice of the bottleneck link's bandwidth.

Ideally the protocol (RLM) will provide both fair *and* stable video quality. However, these properties are independent, and thus it is possible for RLM to fail at one or both. For example, receivers of two different sessions may both experience constant quality (stability) but one session's receivers get up to the highest layer while the other session's receivers only get the base layer (unfairness). On the other hand, receivers of two different sessions may all see continuously changing quality (instability), but over time both sessions receive the same average bandwidth (fairness). Finally, one session may receive constantly changing high-quality video, while another session receives constantly changing low-quality video (unstable and unfair).

4.1 Fairness among competing sessions

In RLM and other receiver-driven layered multicast schemes[VRC98, TPB97, WSS97], receivers in different sessions independently decide how many layers they should subscribe to. To analyze how these decisions are made, we define a receiver's *hold priority* for layer L, denoted h_L, as the importance (priority) of holding (not dropping) layer L. Similarly, we define a receiver's *request priority* for layer L, denoted r_L, as the importance (priority) of adding layer L. Both h_L and r_L

are typically defined in terms of the minimum amount of time that congestion must persist before dropping a layer. Stability is enhanced by setting the hold priorities higher than the request priorities. However to achieve fairness in layer allocation, receivers requesting lower layers must have precedence over those requesting or holding resources for higher layers. This implies that for a layered scheme to be fair and stable, the holding priority of each layer must exceed the request priority for that layer, and the request priority of a lower layer must exceed the holding priority of a higher layer. More formally, we must have that $h_L > r_L$, and $r_L > h_{L+k}, \forall k > 0$, where r_L and h_L are the request and hold priorities for layer L, respectively. Clearly this implies that $r_L > h_{L+k} > r_{L+k}$ and $h_L > h_{L+k}$ for $k > 0$. In other words, the hold and request priorities must decrease for higher layers.

The challenge to achieving a deterministic and stable state is to dynamically determine settings for h_L and r_L across all sessions (and their receivers) to meet the requirements above. We want h_L to be large enough to withstand transient congestion, but small enough to detect real load shifts. Unfortunately, it is difficult for a receiver to distinguish transient congestion from persistent shifts in network load. Transient congestion may be protocol induced, as caused by join experiments from other sessions, or random congestion as caused by new sessions, or by variable bit rate traffic from this or other existing sessions. The duration of the join-experiment, r_L depends on the time it takes a joiner to detect its own congestion. This means that the hold times must be set dynamically based on the request times in order for a receiver to persist through the congestion caused by a join-experiment. Given this dependency, coordination is required between receivers in order to set the h and r values in a way that ensures fairness and stability. We are currently investigating methods to approximate the above optimal priority assignment without communication.

In RLM, receivers do not even know about other RLM sessions, which seems a reasonable design. However, it means that receivers from different sessions do not communicate to set the values of h_L and r_L. Instead, h_L is based on RLM's detection-timer (which is derived from network measurements of the time it takes a joiner to generate congestion) and varies from receiver to receiver. The request time is designed to be much smaller than the hold time, $r_L << h_L$, so that join-experiments will not cause receivers to give up the layers they currently hold. This works well within a session because the receivers that are getting higher layers do not steal bandwidth from receivers with lower bandwidth. However, this does not work well for multiple competing sessions. RLM's algorithm tries to "ride-out" transient congestion rather than give up a layer to a more deserving session. Only if the congestion persists for at least two detection timeout periods will a receiver believe the congestion is a persistent change in the offered load and drop a layer.

4.1.1 Fairness Experiments

To quantify the potential unfairness of RLM, we created multiple RLM sessions sharing a single bottleneck using the topology show in Figure 1 and described in section 2.1. We then recorded (and graphed) the subscription level of each session during every second of the simulation. The subscription level provides an indication of the quality currently observed by each session.

We began by testing the simplest case: two competing RLM sessions sending CBR traffic. Our thinking was that if RLM is unable to achieve fairness in this situation, then it is unlikely that RLM will achieve fairness under any conditions. Figure 9a shows the subscription levels of the two CBR sessions over time when the two sessions are started at roughly the same time; within 10 seconds of each other. In this case, the receivers ramp-up at approximately the same rate until the link becomes congested. Because they ramp-up at similar rates, they grab equal amounts of the bottleneck bandwidth and achieve fairness, demonstrating that in an ideal situation RLM will provide both fairness and stability. (Interestingly, starting the sessions at exactly the same time increases the time it takes for them to converge to the fair level because the join-experiments overlap).

Knowing that RLM is engineered for stability and thus hesitates to give up a layer, we wanted to see if a new session could steal bandwidth away from an existing session and achieve a fair allocation of the bandwidth. To test this, we reran the previous experiment but started the sessions at different times, namely times 60 and 260 in the simulation. Figure 9b shows the subscription levels for the two sessions. Initially the first session is able to subscribe to five layers, consuming all the bandwidth on the link. The second session initially subscribes to and gets the base layer without exceeding the maximum loss rate threshold. However, as soon as it attempts to join layer two it causes substantial congestion causing it to backdown. On the other hand, the first session goes into a measurement state to see if the congestion was short-term congestion or long-term congestion. Because the second session gives up quickly, the

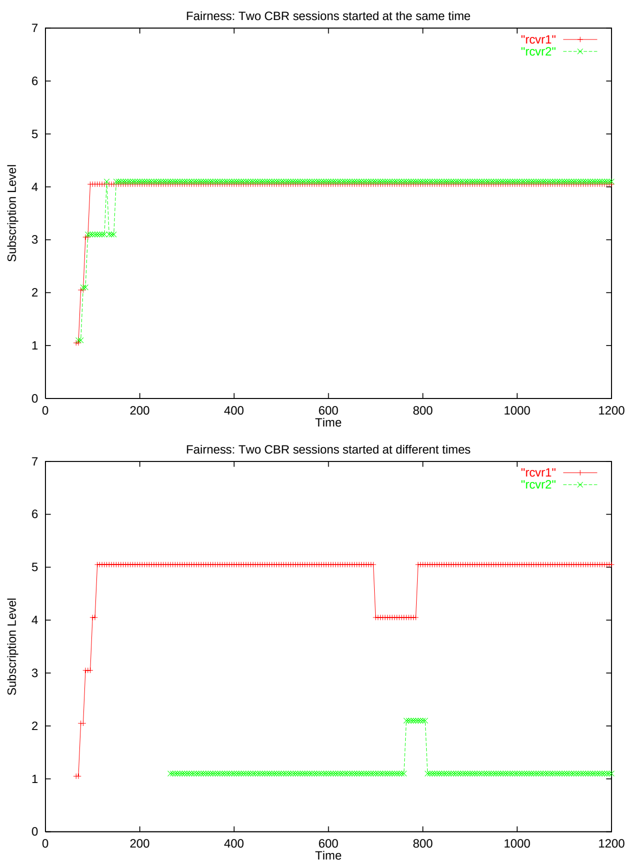

Figure 9: Fairness between two competing CBR sessions. Figure (a) on the left shows the sessions started at approximately the same time while Figure (b) on the right shows the sessions started at different times.

first session assumes it was short-term congestion and does not give up its layers. In short, RLM with CBR traffic has the potential to be extremely unfair. Bandwidth is allocated among sessions on a first-come-first-served basis which is unacceptable for the IP-based TV distribution mechanism we envision. For example, the person that wants to tune into their televised university class may not be able to obtain more than the base layer because another user started watching the NASA space shuttle launch first. By virtue of being first, the NASA watcher receives HDTV quality video and refuses to give up quality. Ironically, RLM is suppose to be designed to provide incentives for users to cut back their bandwidth usage in the face of congestion [BBS98].

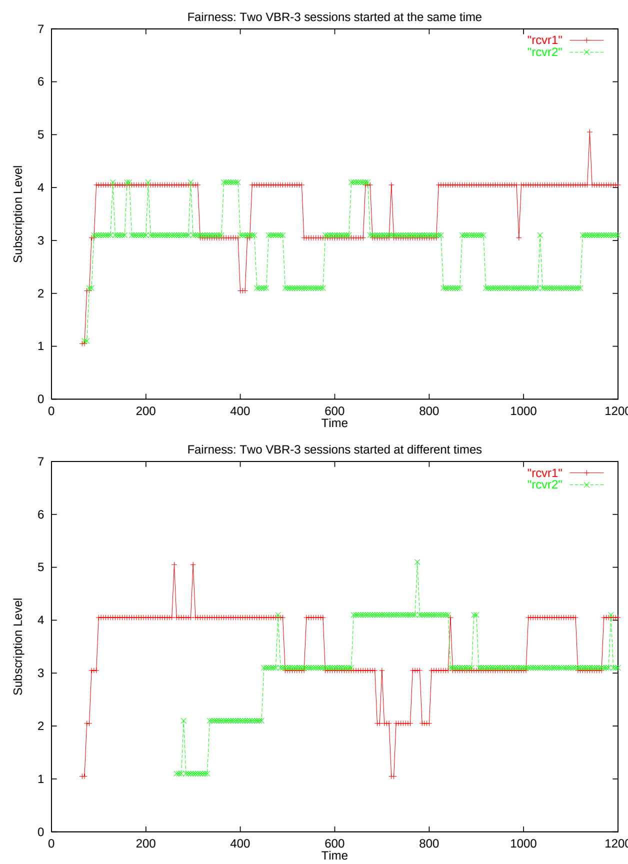

Figure 10: Fairness between two competing VBR-3 sessions. Figure (a) on the left shows the sessions started at approximately the same time while Figure (b) on the right shows the sessions started at different times.

To see if VBR traffic affects RLM's ability to achieve fairness, we reran the previous test using VBR-3 traffic instead of CBR traffic. Figure 10 shows the results which show the system is highly unstable, even when the sessions are started at different times. In fact, the system is so unstable that it is difficult to conclude that the system is fair or unfair. Although the two sessions are rarely subscribed to the same level, there does not appear to be any inherent bias in the algorithm that gives one session preferential treatment over another. In fact, over the long-term, figure 10a shows the average subscription level for session one and two to be 3.61 2.77 respectively, while figure 10b shows the average subscription level for sessions one and two to be 3.42 and 2.98 respectively. Certainly, the service received by the sessions differs but the difference it not as much as we anticipated. Similar observations can be made about the graphs in Figure 8. Part of the reason for the improved average fairness is the fact that the burstiness of the traffic prevents the algorithm from subscribing to the highest possible layer (5) (even when there is only one session in figure 10b from time 60 to 260), thereby leaving bandwidth available for new sessions.

On the other hand, each graph exhibits short periods of stability. Only a few of these stable periods show the sessions at the same layer. During most periods, the subscriptions levels differ by one and often two levels. In other words, on shorter-time scales, the protocol seems to be highly unfair, regardless of the session start time.

5 Heterogeneity

The primary aim of RLM is to support video distribution in a heterogeneous bandwidth environment, i.e. where receivers can have different bottleneck bandwidths, typically determined by their access link. The bottleneck capacity determines the number of layers to which a receiver can subscribe. RLM is said to "search for the optimal level of subscription" by adding layers until congestion occurs. When congestion occurs, it drops the layer, but continues to probe at progressively larger intervals to see if the bottleneck bandwidth has increased. In this section we present results for a set of experiments that aim to evaluate the fundamental claim that RLM can find and operate at the "optimal" level. In particular, we question the definition of "optimal". We examine the performance of a single receiver whose bottleneck link is the last-hop link (*e.g.*, the connection from an ISP to the home). In our experiments, the bottleneck link bandwidth is capable of sustaining four layers. This amounts to 480 Kb/s and is certainly in the range of bandwidths offered by digital subscriber loop (DSL) technologies. Subscription level plots are generated with an averaging interval of 5 seconds; the utility and received packet plots use a 50 second interval.

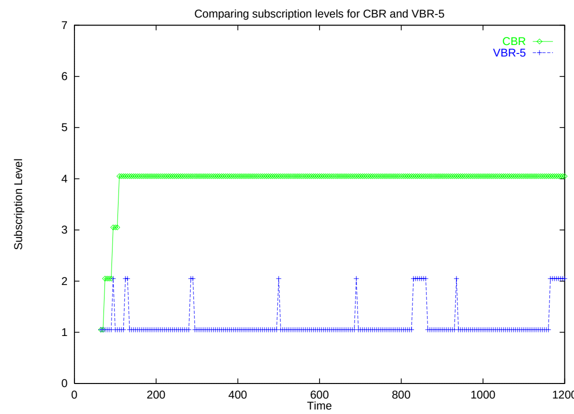

Figure 11: Subscription levels for a single receiver.

The first experiment aims to reproduce the result presented in [McCanne96] for a CBR source. As shown in Figure 11 RLM does indeed find the bottleneck bandwidth for CBR traffic. Next we conducted experiments to evaluate the effectiveness of RLM for VBR traffic sources. The burstiness of VBR traffic can be expected to produce greater packet losses in comparison to CBR, and may cause RLM to backoff, resulting in a lower average subscription level. The result is shown in Figure 11 compared with CBR. The impact is quite dramatic, with the subscription level rarely exceeding more than just the base layer. This is extremely low utilization of the link, given that the base layer, on average, accounts for just 1/15th of the available bandwidth. Clearly VBR traffic significantly reduces the effectiveness of RLM in determining the highest possible link utilization and leads us to conclude that RLM is overly cautious in its operation. Depending on the utility function used to evaluate user happiness, our home user may be very displeased with average utilization of just 32 Kb/s from the 480 Kb/s link.

The low level of subscription for RLM with VBR traffic raises the question of whether one would be better off with uniform dropping. In uniform dropping, the receiver statically chooses its optimal subscription level and performs no adaptation in the face of congestion. The network drops packets uniformly across all layers so that each layer experiences the same loss rate. The next set of experiments compare uniform and RLM. We show the packets received per layer in Figure 12a. The source sends three layers but RLM never subscribes to more than two, so we compare with the case where uniform is subscribed to two layers in order to match that of RLM. The results shows that RLM obtains better performance for the base layer than uniform. With uniform, layers 1 and 2 have averages of 3.44 and 7.04, while for RLM layers 1 and 2 have averages of 4.12 and 0.79. On the other hand, uniform dropping can clearly deliver more packets from a greater number of layers, as seen in Figure 12b. To compare these results really requires information which is dependent on the layered coder, specifically regarding how much packet loss is tolerable on a layer, and also the dependence of higher layers on the performance of lower layers.

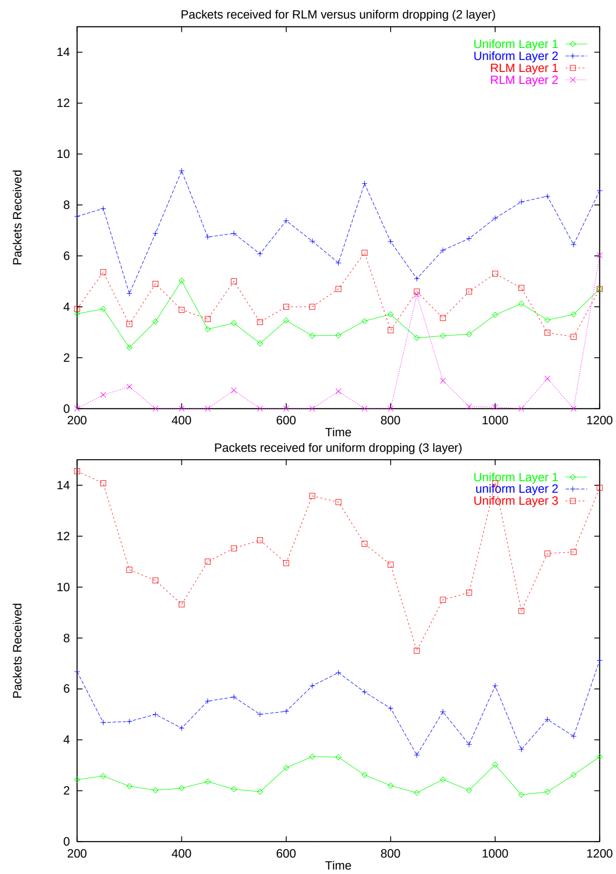

Figure 12: Received packets for a single receiver with VBR-5 traffic. Figure (a) depicts the case with uniform subscribed to 2 layers, while figure (b) is for uniform subscribed to 3 layers.

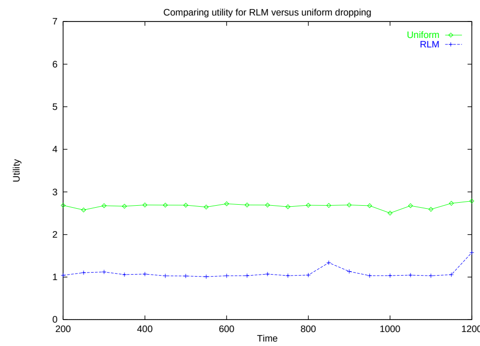

Figure 13: A comparison of utility for a single receiver with three layers of VBR-5 traffic.

To further compare, we examined the impact on utility, where utility is a rough estimate of the service quality observed by a receiver. The utility function maps the service received (usually based on the number of packets received) to a performance level. We used the utility function from [BBS98] in which the utility (value-per-bit) is exponentially decreasing with each layer, resulting in layers of equal utility. Figure 13 shows the utility results for RLM and uniform dropping, using three layers. The difference in favor of uniform dropping is substantial, showing well over 50% better utilization. An issue in comparing RLM and uniform dropping is that it depends on a specific choice of utility function. In short, the choice of parameters used in the RLM implementation defines the "optimal" subscription level, which may, or may not, match the user's definition of optimal utility.

6 Related Work

We have conducted a detailed analysis of RLM, investigating issues of fairness, stability and bandwidth utilization. After having reproduced and verified the results of

[MJV96, BBS98], we have extended their simulations, allowing us to scrutinize the assumptions and properties of the protocol. In particular we have studied the performance of multiple competing RLM sessions. Although multiple sessions were used in [BBS98], the paper focused on the very specific problem of priority versus uniform packet dropping policies, and did not address the interaction between RLM sessions.

ThinStreams[WSS97] addresses issues of fairness for multiple sessions consisting of "thin" equal-sized layers. The algorithm requires that receivers calculate join and leave thresholds based on their current level of subscription. The idea here is that receivers subscribing to a larger number of layers will surrender them more quickly than a receiver with a smaller subscription set. Stability issues are not addressed, and may be significant given the join/leave overheads of thin layers.

The issue of fairness between RLM and TCP traffic has also been studied [VRC98, TPB97]. The idea in [VRC98] is for receivers to use a join/leave strategy for congestion control which mimics the behavior of TCP. This relies on making appropriate choices of layer bandwidths and the time delay between trying to increase subscription. In [TPB97], each receiver tries to determine the share of bandwidth that an equivalent TCP connection would use, and then makes join/leave decisions in order to match that value for the multicast session. TCP's objectives differ significantly from the objectives one would design for a video transmission protocol. Consequently, making the layered multicast scheme behave like TCP can help it "get along", but may wreak havoc on the "visual experience". The focus in both studies is to provide fairness between all TCP and all RLM traffic, rather than between individual RLM sessions. Although these solutions interact better with TCP, they will experience the same saw-tooth instability that TCP flows. In fact, the use of multiplicative decrease will result in more sudden and drastic level changes than RLM. Also, because these protocols behave like TCP, they will offer "fairness" similar to that of TCP, which is known to produce arbitrary unfairness. Other enhancements over RLM such as synchronized joins between receivers in the same session may offer substantial improvements over RLM. However, since our study used a single reciever per session, these enhancements will not improve on the RLM stability and fairness results presented here.

Other work on layered multicast includes [LPA98, LPPA97], which addresses error recovery issues and [BCZ98] which uses router-based adaptation to provide more accurate reactions to network congestion.

7 Conclusions

In this paper we have reported on our studies of how RLM manages multiple competing sessions. Specifically we have studied the stability and fairness of the layered allocation across multiple sessions, and how these two metrics are affected by number of sessions and variability of the traffic offered by these sessions. Lastly we have studied the ability of RLM to adapt to heterogeneity and discover the bandwidth of an access bottleneck.

We conclude that RLM exhibits significant and persistent instability with VBR traffic. This instability causes frequent layer shifts - frequent enough to be intolerable by users, still too far apart for an application to be able to effectively mask it. In one example, using our most variable source we observed significant level shifts every 45 seconds. At times the instability is severe enough to cause the relative allocation of sessions to flip-flop. As observed in the original RLM paper, RLM is stable for CBR.

With CBR traffic RLM can be arbitrarily unfair. Once a session has achieved dominant allocation, another CBR source is starved. We showed an example where a session starts after another session has ramped up to link capacity. The second session only manages to add the base layer and is unable to acquire its fair allocation from the competing session. With VBR traffic the instability of RLM prevents us from reaching a strong conclusion. Competing VBR sessions seem to alternate between periods of relative stability and periods of absolute chaos. During the stable periods the system is arbitrarily unfair. However during unstable periods (when there is flip-flopping of sessions), we notice 20 (simulated) minute runs in which the average subscription level for the run is approximately the same across sessions.

We find that RLM successfully adapts to heterogeneous links. However we find that RLM aggressively (perhaps too aggressively) protects the lower layer of the video, to the point of not tolerating any losses at lower layers before utilizing the higher ones. With VBR traffic this results in very low utilization of the link bandwidth. Although this is consistent with the stated goals of the RLM design, this exposes a built in assumption of the utility function, namely that any fraction of a lower layer is more valuable than some fraction of a higher one. We question the validity of this assumption. Of course the incredibly low utility observed also questions the goodness of the encoding scheme used.

While our studies have identified some significant problems with RLM, significantly more work is warranted. We are still exploring (a) inherent limitations of RLM and (b) how layered multicast without the problems of RLM can be realized.

References

[BBS98] S. Bajaj, L. Breslau, and S. Shenker. Uniform versus priority dropping for layered video. In *Proceedings of the SIGCOMM '98 Conference*, pages 131–143, September 1998.

[BCZ98] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. Network support for multicast video distribution. Technical Report 98-16, Georgia Institute of Technology, College of Computing, 1998.

[Com93] MPEG-2 Systems Committee. MPEG-2 Systems Working Draft. ISO/IEC/JTC1/SC29/WG-11-N0501, July 1993.

[GGHS99] R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, and C. Sreenan. A Stability and Fairness study of Layered Multicast Congestion Management, June 1999. (in preparation).

[LPA98] X. Li, S. Paul, and M. Ammar. Layered Video Multicast with Retansmissions (LVRM): Evlauation of Hierarchical Rate Control. In *Proceedings of the INFOCOMM '98 Conference*, 1998.

[LPPA97] X. Li, S. Paul, P. Pancha, and M. Ammar. Layered Video Multicast with Retransmission (LVRM): Evaluation of Error Recovery Schemes. In *Proceedings of the NOSSDAV '97 Conference*, May 1997.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-Driven Layered Multicast. In *Proceedings of the ACM SIGCOMM '96 Conference*, October 1996.

[MVJ99] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity Video Coding for Receiver-driven Layered Multicast. *IEEE Journal on Selected Areas in Communication*, 1999.

[RT99] Jennifer Rexford and Don Towsley. Smoothing variable-bit-rate video in an internetwork. *IEEE/ACM Transactions on Networking*, pages 202–215, April 1999.

[TPB97] T. Turletti, S. F. Parisis, and J-C. Bolot. Experiments with a layered transmission scheme over the Internet. Technical Report 3296, INRIA Sophia Antipolis, France, November 1997.

[VRC98] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of the INFOCOM '98 Conference*, pages 996–1003, March 1998.

[WSS97] L. Wu, R. Sharma, and B. Smith. Thinstreams: An Architecture for Multicast Layered Video. In *Proceedings of NOSSDAV '97*, 1997.