

On Layered Video Fairness on IP Networks

Ashraf Matrawy Ioannis Lambadaris Changcheng Huang

Broadband Networks Laboratory
Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Dr., Ottawa, ON K1S 5B6, Canada

Abstract—In this paper, we present a study of layered video fairness on IP networks. Our study is based on simulation. We investigate some issues that have direct impact on fair allocation of bandwidth between layered video and TCP, in particular: a) Congestion control mechanisms employed by layered video transfer protocols. For this part we studied the interaction of RLM with TCP. b) The effect of the distribution of video traffic across layers in layered multicast video. c) The effect of VBR video on fairness to TCP. We show that fairness is affected by all the above factors. We also show that fairness of layered video comes at the expense of instability of the video quality and poor link utilization. We conclude by discussing the performance of layered video protocols in general and recommendations on the design of video transfer systems on IP networks.

I. INTRODUCTION

The distribution of digital video is an important component of any existing and future networked service. In many situations, this involves delivering video to many receivers. Multicasting is a good solution to save the bandwidth in this case. On the Internet, receivers have different networking capabilities as well as quality requirements. Due to this, we must deal with the challenge of heterogeneity. In the case of real-time video, some guarantees on upper bounds of delay, loss rate, and delay jitter should be met.

On today's Internet, more than one approach exist for handling these requirements. In [1], a discussion of the earlier approaches can be found. Here we consider the solution that became the basis for all the works coming after it: Coupling layered video encoding and layered multicasting (sending each layer as a separate multicast group) was a good solution for the problems of heterogeneity and scalability. This is the Receiver-driven Layered Multicasting (RLM) proposed by McCanne [2], [3] which we will present in Section III. RLM is an elegant solution to the heterogeneity problem as it moves the control (of the rate that a certain receiver gets) to the receiver rather as opposed to the sender-driven approaches. A few proposals have attempted to improve RLM's performance from the video receivers point of view [4], [5], [6], [7], [8].

One of the features that all these approaches share is that they rely on a non-TCP protocol (usually UDP with some application level control or UDP with RTP). This is because video applications are both loss-tolerant, to some extent, and time-sensitive. Using TCP for these applications introduces delays because of the congestion control mechanisms that TCP uses. On the other hand, the level of reliability that TCP provides is not required

This research was funded in part by grants from: Communications and Information Technology Ontario (CITO), Natural Sciences and Engineering Research of Canada (NSERC), Mathematics of Information Technology and Complex Systems (MITACS), and Nortel Networks

for video applications as it is in the case of data transfer. That's why most of the video applications on the Internet are based on UDP.

As these applications become widespread, it is important to ensure that they are *able to co-exist* with the current TCP-based applications on the Internet. TCP is the base for most of the existing applications on the Internet. The use of the end-to-end congestion control mechanisms of TCP has been a critical factor in the robustness of the Internet today [9]. End-to-end control mechanisms are a key corollary to the scalability argument of the Internet: to maintain scalability, complexity should be pushed to the edges of the network whenever possible [10]. On the other hand, UDP does not provide this end-to-end control and hence when it competes for bandwidth with TCP, which strictly adheres to congestion control, the result is *unfair* bandwidth allocation to TCP. In the case of video applications, congestion control should be augmented in a level higher than UDP. We will overview the problem of *fairness* on the Internet in Section II.

In this paper, we address the question of *fairness to TCP* in the case of layered video. We considered RLM for our study because it is the base for all recent layered video protocols. In addition to the way RLM handles congestion, we believe that fairness is affected by other factors as well. Hence, we first study the bandwidth allocation between *competing flows of RLM and TCP*. Both RLM and TCP are non-stationary and non-linear [2] which makes a formal analysis of their interaction very difficult. Our study is based on simulation. Then, we show that rate distribution across layers has an impact on fairness. Finally, we show that with VBR video, fairness may be achieved at the expense of instability and poor link utilization. Our results show that fairness is sensitive to RLM's congestion control mechanism, to the rate distribution across layers and to the burstiness of video traffic. We also present some recommendations on designing video transfer systems.

The rest of the paper is organized as follows: Section II presents the problem of fairness on the Internet. Section III explains how RLM works. In Section IV, we describe our simulation setup. Section V presents our study of the interaction of RLM and TCP. Section VI presents the effect of rate distribution across layers on fairness and in Section VII we show how VBR video has an impact on fairness. Section VIII concludes the paper.

II. FAIRNESS ON THE INTERNET

The Transmission Control Protocol (TCP), widely used in the Internet for providing reliable end-to-end transport, incorporates a window-based congestion control algorithm to reduce congestion in the network. This results in a fair sharing of network bandwidth even though the protocol operates in a distributed manner [11]. Unfortunately, not all the protocols on the Internet behave in this way which creates problems of unfairness in terms of bandwidth allocation between competing flows.

In [12], a set of tests were provided to classify flows as *unresponsive*, *not TCP-Friendly* or simply using disproportionate bandwidth. An *Unresponsive* flow [9] is one failing to reduce its offered load at a router in response to an increased packet drop rate. A flow that is *not TCP-Friendly* [9] is one whose long-term arrival rate exceeds that of any conformant TCP in the same circumstances. Unresponsive flows do not use end-to-end congestion control, and do not reduce their load on the network when subjected to packet drops. Such behavior will result in both unfair allocation of bandwidth and the danger of congestion collapse. Congestion collapse occurs when an increase in the network load results in a decrease in the useful work done by the network. It was first reported in the mid 1980's [13]. Unfair allocation occurs when TCP responds to congestion by reducing its transmission window while other non-TCP flows do not. The largest unresolved issue with respect to either unfair allocation or congestion collapse is due primarily to the increasing deployment of open-loop applications not using end-to-end congestion control [9]. Video applications are a key example in this category of applications. Some work has been done on providing rate control and adaptation techniques [7], [14], [15], [16].

III. RLM FUNCTIONALITY

In this section, we overview the RLM protocol and its features that are relevant to our work in this paper. Receiver-driven Layered Multicast (RLM) [2], [3] is a transport protocol that allows receivers to adapt the quality of the video they receive according to their available bandwidth. In RLM, layered video coding is used where the video signal is encoded into a number of layers. The lowest layer contains the basic information, and each subsequent layer provides progressive enhancement. The sender sends each video layer to a separate IP multicast group and takes no active role in rate adaptation. Each receiver subscribes to a certain set of video layers by joining the corresponding IP multicast group. The advantage of receiver-based control over sender-based control is that the burden of adaptation is moved from the sender to the receivers, resulting in enhanced system scalability.

Each receiver tries to achieve the optimal level of subscription of video layers. Optimal level of subscription is the maximum level a user can get within the *available* link capacity. The basic adaptive control works as follows: when a receiver detects congestion, it drops a layer, and when there is spare bandwidth available, it adds a layer. Deciding whether the current reception level is optimal is a crucial step in the control loop. If congestion is detected, this means that the current subscription level is too

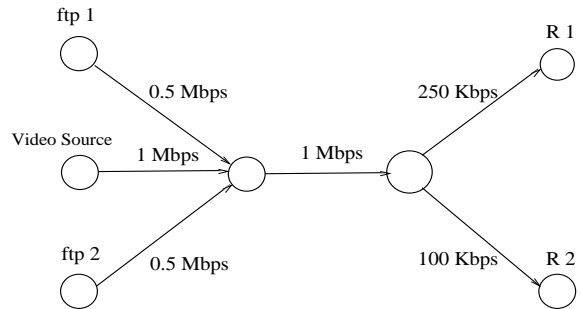


Fig. 1. Simulation setup

high. In RLM, congestion is detected with packet loss rate. On the other hand, there is no explicit feedback on whether the current subscription level is too low, so in RLM, join-experiments are carried out to find if the receiver is capable of handling the next video layer. If congestion is detected after the experiment, the receiver drops the newly added layer. If congestion is not detected the experiment is successful. A failed join-experiment can bring congestion to the network, resulting in degraded video quality to both the receiver that initiated the experiment and possibly other receivers that share the congested link. Therefore, a learning algorithm is proposed so as to minimize the frequency and duration of join-experiments without impacting the video layer convergence rate. A detailed discussion of the protocol receiver state machine can be found in [2].

IV. SIMULATIONS

In this section, we describe the simulation setup used for our experiments. All the simulations were carried out using the LBNL network simulator *ns*[17]. The topology is illustrated in Fig. 1. One source of video traffic is used. As for TCP traffic, ftp1 and ftp2 are two sources representing ftp applications. There are two receivers, R1 and R2 with different link capacities, 250 Kbps and 100 Kbps respectively. The ftp application is running on top of TCP where ftp1 is sending its packets to R1 and ftp2 is sending to R2. All simulations were run for 6000 simulated seconds. The senders and receivers are connected via a link of 1 Mbps as shown in the figure. The video source is modeled as a set of traffic sources, each source is representing one layer of video. We will describe the characteristics of each source and the number of layers later with each experiment. Buffers at the nodes are managed using a simple drop-tail strategy where incoming packets are dropped when the buffer is full. All simulations are done using Tahoe TCP. In the rest of this paper, when we refer to TCP we mean Tahoe TCP unless otherwise stated. We used the RLM implementation in *ns*. The packet size is fixed in all our simulations to 1000 bytes for both protocols.

V. CONGESTION CONTROL OF LAYERED VIDEO

In Section III, we described how RLM reacts to congestion. Now we present our study of its interaction with TCP. This is important as TCP traffic constitutes the majority of Internet traffic. Hence, before introducing layered video to the Internet, it

has to be tested for fairness to TCP. We want to show that the way RLM (and any layered video transport protocol) changes its layer subscription has a direct impact on TCP.

A. Basic experiment

We started by reproducing the results in [2]. Then we built the simulation setup in Fig. 1 to run our experiments using ns [17]. We simulated an RLM video multicasting session from the video source to R1 and R2. In the same simulation, we transferred two files, each of them is 5MB. One file from ftp1 to R1 and the other from ftp2 to R2. The video source is modeled as a set of CBR sources, with 4 layers. The rates for the layers are 32Kbps, 64Kbps, 128Kbps, and 256Kbps. Rates are distributed in an exponential fashion with 32Kbps for the basic layer. The rate is doubled as we go one layer up.

Based on this setup, we simulated different video/data transmission scenarios. a) when RLM starts first, b) when both RLM and TCP start at the same time, c) when TCP starts first. For the lack of space, we show here only the case where RLM starts first as it is the case with severest unfairness. Interested readers may refer to our work in [18] for completed results of all three cases.

For the results we show some measured metrics (number of video layers, RLM throughput, TCP throughput,...etc.) for each of R1 and R2 with different values of W . W is the length of the window in packets during which RLM measures its packet loss (as implemented in ns). This parameter controls how fast RLM reacts to congestion. An RLM receiver drops its highest layer of video if a certain percentage of packets is lost from a measuring window of W packets. With large values of W , RLM reacts to loss very slowly which causes competing protocols (in our case TCP) to lower their sending rates first.

It is worth mentioning that the frequency of joining a higher layer is dependent on a join timer, T_J , which is maintained by each receiver for each layer. This set of timers are given values based on the results of the join experiments [2]. The initial setting and the maintenance of these timers plays a crucial role in shaping the behavior of RLM and how aggressive it is when it tries to join higher layers.

In this set of experiments, first we start RLM at the beginning of the simulation, then after 100 seconds we start TCP (enough time for RLM receivers to reach stability). In Fig. 2, for R1 we show the number of received layers, RLM throughput, TCP window size in packets, TCP throughput, video packet loss ratio¹ shown against time (in log scale), and finally the link utilization.

In Fig. 2, $W = 12$, which is fairly large, this means that when RLM reaches stability (in this case 3 layers for R1), it will respond to packet loss if it loses 25%² of the packets received in a window of 12 packets. The result is a very persistent behavior from R1. It tries to keep layer 3 despite the attempt of TCP to get some bandwidth and despite the increasing packet loss. TCP

¹for an RLM receiver at a given time t , this is the total number of lost packets till time t divided by the total number of packets received till time t

²value used in ns implementation of RLM

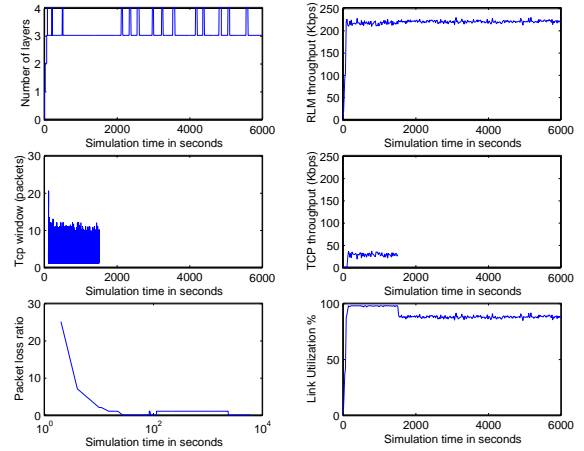


Fig. 2. RLM starts first, R1 with $w=12$

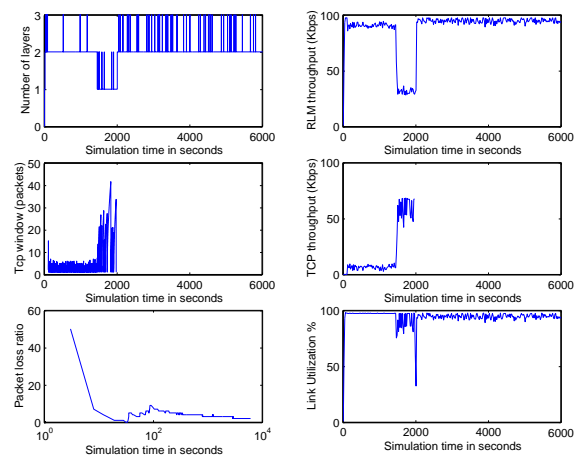


Fig. 3. RLM starts first, R2 with $w=12$

can not finish earlier than 1500 seconds and window size keeps oscillating with a maximum value of 10. This value is fairly small considering the available bandwidth on the link. In the figure, the video packet loss ratio is high only the transient time until RLM reaches stability. Also, we can notice the stability in RLM throughput and the high link utilization. In Fig. 3, for R2 we see almost the same behavior for the same value of $W=12$. The difference here is that when R2 gets two layers (96Kbps), it leaves only 4Kbps for TCP. This is shown in the very small TCP window size and the higher loss than R1's case. This leads to R2 dropping of its highest layer (layer 2) after 1500 seconds. This drop lasts for around 500 seconds that were enough for TCP to finish the file transfer. This drop is shown also in both RLM throughput and in the link utilization. It is clear how slow RLM is in this case in responding to congestion. For a fairly long time it maintained its subscription to three video layers and led TCP to bandwidth starvation. In [18], we repeated our experiment for different values of W . There was no significant difference in comparison with our earlier experiment until W is set to a small value. We selected $W = 4$. This selection results in a much

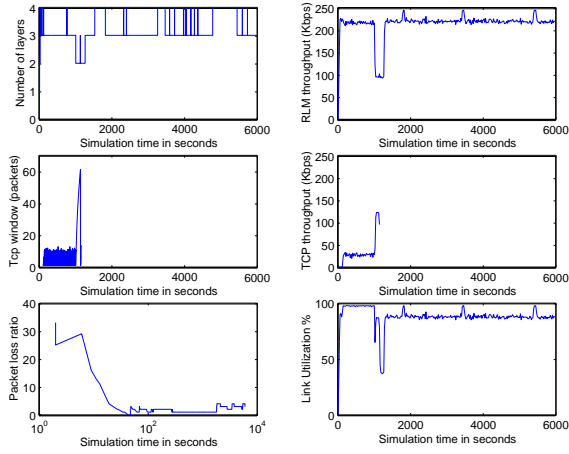


Fig. 4. RLM starts first, R1 with $w=4$

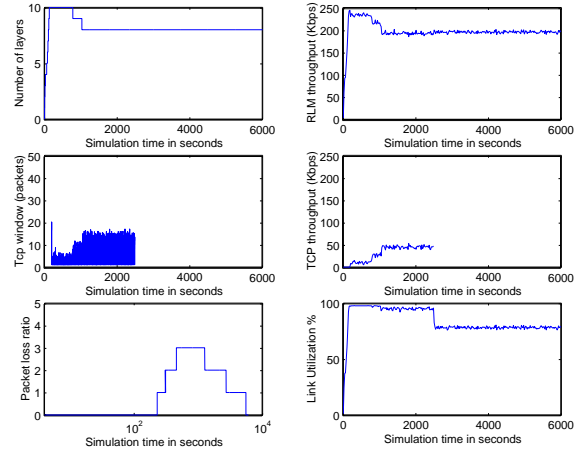


Fig. 6. CBR linear R1

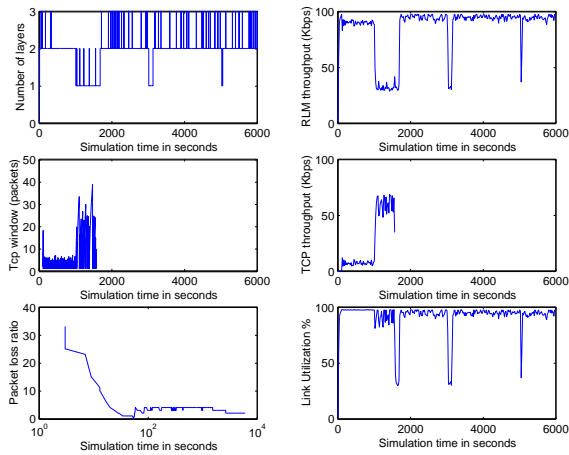


Fig. 5. RLM starts first, R2 with $w=4$

faster response from RLM to packet loss. In Fig. 4, R1 drops layer 3 after 1000 seconds which allows TCP to finish earlier compared to the case shown in Fig. 2 (which corresponds to $W = 12$). R2 in Fig. 5 behaves in the same fashion. It drops its highest layer faster than the earlier case of Fig. 3 and again TCP finishes earlier.

B. Discussion

From these results it is clear that in order for RLM to perform as described in [2], it shows an unfair behavior towards other protocols that try to share the bandwidth with RLM. This unfairness was reported in [19] towards other RLM flows. Our results show that RLM is also unfair to TCP. This unfairness is the result of the way TCP reacts to packet loss by reducing its window size with the first packet loss. Meanwhile, the mechanism that RLM employs to react to congestion is different. It is very persistent to drop its highest layer (and hence reduce its rate) when it reaches its optimal number of layers. Also, RLM performance is greatly dependent of the setting and maintenance of a set of parameters. Setting this parameters to certain val-

ues may result in a more fair behavior from RLM towards TCP. However there is a tradeoff between fairness and the quality of the received video that should be resolved before setting these parameters to certain values. More analysis of RLM's problems can be found in [18], [19], [20].

VI. RATE DISTRIBUTION FOR LAYERED VIDEO

In this section, we study the effect of the distribution of the bandwidth of the video source across the layers in layered multicast video. In particular we compare the impact on fairness by distributing the layered video rates in linear, exponential or hyperbolic fashions. We are using RLM and the simulation setup of Fig. 1. Also, for RLM we are using $W = 4$ to insure fast response from RLM. The ftp file size in these experiments is 10MB. The video source is modeled as a set of CBR sources to represent the layers. In the three cases, we are producing approximately the same average rate for the whole source.

A. Linear increase of layered video rates

In this case, the source is modeled as a set of 10 CBR layers each with rate of 25Kbps. This distribution has an advantage of stability for the video and reduction in big oscillations in the network. Also a good link utilization as the amount of bandwidth gain or released by RLM is not large. The disadvantages are, unfairness to TCP. Compare results in Fig. 6 and Fig. 7 to Sections VI-B and VI-C. TCP takes longer time to finish because in this case RLM release only 25Kbps at time when congestion is detected. From the video application point of view, convergence in this case takes longer time. There is also the overhead of managing 10 layers of video at the receiver (note that they are sent as 10 different multicast groups).

B. Exponential increase of layered video rates

In this case, the source is modeled as a set of 4 CBR layers with rates, 16Kbps, 32Kbps, 64Kbps, and 128Kbps. The basic layer is 16Kbps. With this distribution, TCP gets a bigger share of bandwidth compared to the two other methods of distributing

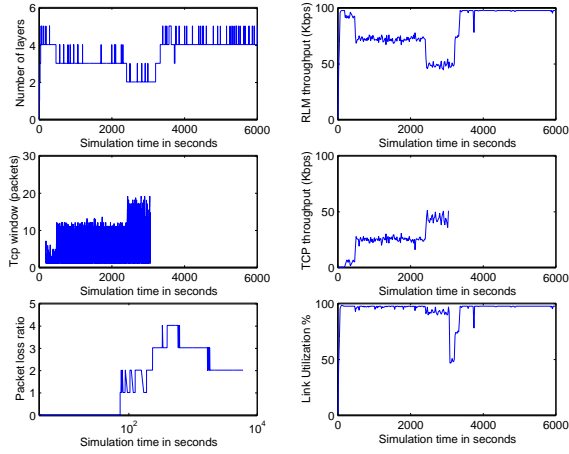


Fig. 7. CBR linear R2

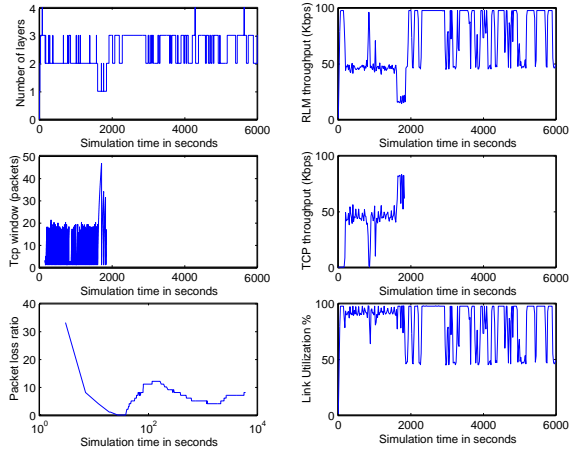


Fig. 9. CBR Exponential R2

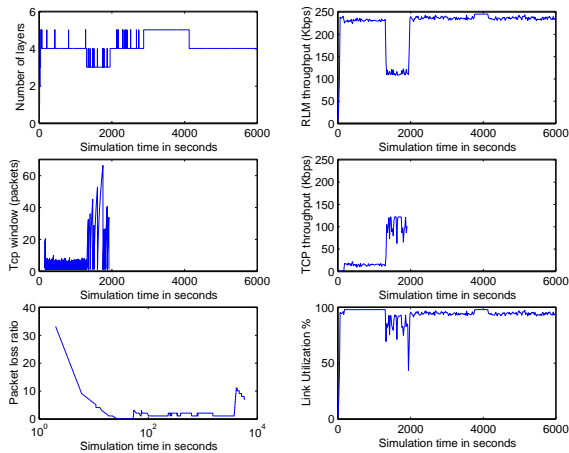


Fig. 8. CBR Exponential R1

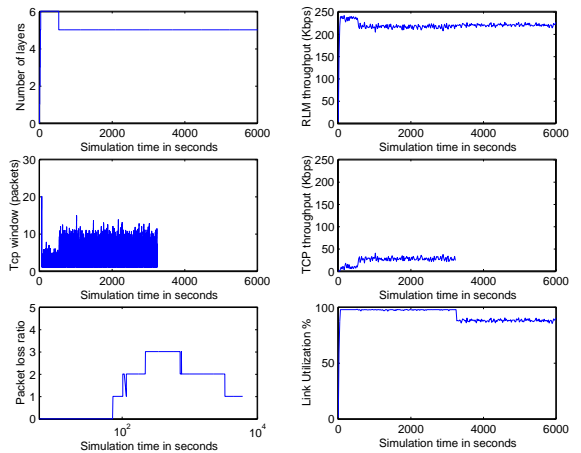


Fig. 10. CBR Hyperbolic R1

the rates (linear and hyperbolic). It is clear that this happens only because of the granularity of the layers. Upon congestion detection, RLM drops the highest layer which corresponds to the highest bandwidth among all layers. The disadvantages are clear too. Large oscillations in subscription level, throughput and most importantly, link utilization. Results for this distribution are in Fig. 8 and Fig. 9.

C. Hyperbolic increase of layered video rates

In this case, we used 6 layers. The rates are distributed as 100Kbps for the base layer, and 50Kbps, 50Kbps, 25Kbps, 25Kbps, 25Kbps for the higher layers in ascending order. Because the basic layer is 100Kbps, we increased the bandwidth for R2 from 100Kbps to 150Kbps. The results in this case depend on both the granularity of the layer and available bandwidth for the receiver. In Fig. 10, R1 has 250Kbps, RLM’s throughput is very stable, and hence RLM is unfair to TCP as shown in the low TCP throughput. Link utilization is very high. In the same experiment, R2 that has only enough bandwidth for the first 2 layers dropped the whole session for around 400 sec-

onds, receiving no layers. This is because in order to get the minimum quality, the receiver has to secure 100Kbps at least. From Fig. 10 and Fig. 11, we can see that TCP got enough bandwidth when RLM dropped to zero layers.

VII. STUDY WITH VBR TRAFFIC

In this section, we study the effect of bursty video sources on RLM performance and on fairness to TCP. The source in these experiments is modeled as a set of VBR sources. Each layer is represented by an on-off source. The distribution of the “on” periods is Pareto with mean 500ms and shape parameter of 1.5. The “off” periods have the exact same distribution. The rates of the “on” periods are set to replicate the experiments in Section VI, i.e., linear, exponential, and hyperbolic. The rate for the “off” periods are all set to zero. Our choice of Pareto is based on the fact that it is the closest to self-similar traffic among the generators available in the network simulator *ns*. We can see from the results in the cases in Sections VII-A, VII-B, and VII-C that they are close to their counterpart in Section VI. But with VBR, RLM is more unstable and link utilization is lower.

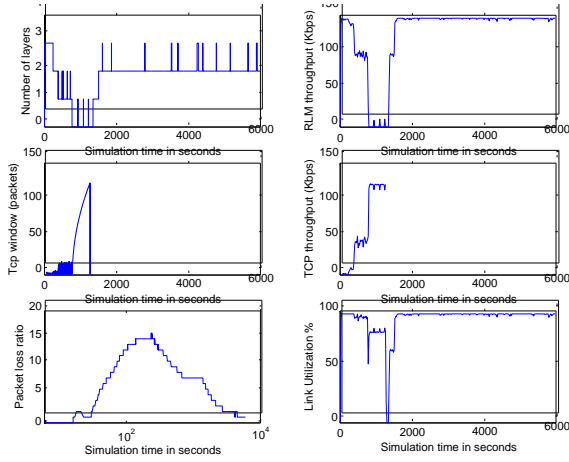


Fig. 11. CBR Hyperbolic R2

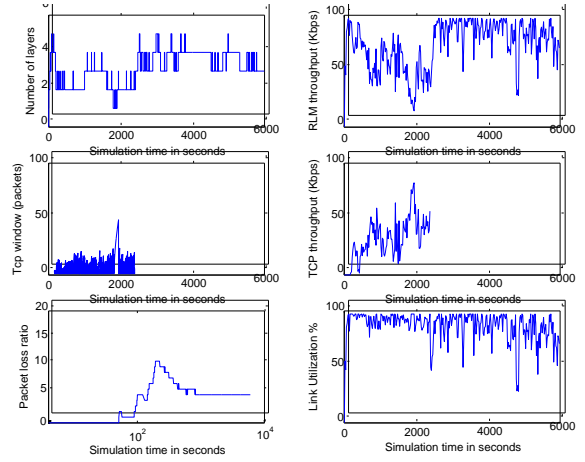


Fig. 13. VBR linear R2

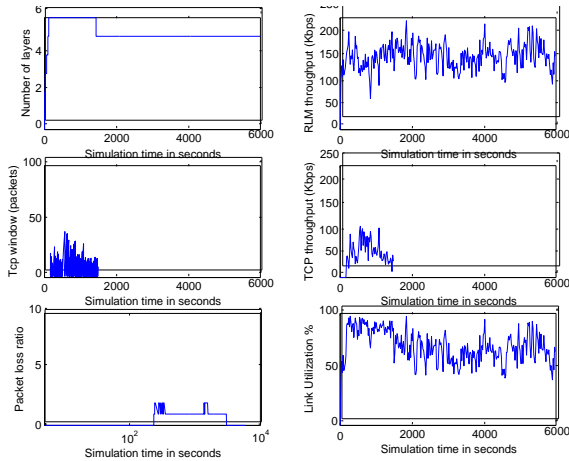


Fig. 12. VBR linear R1

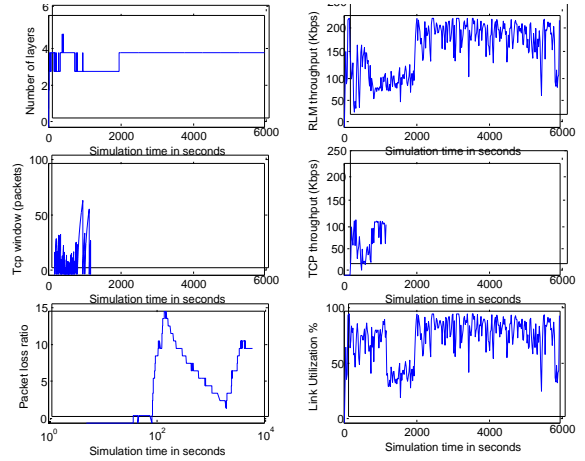


Fig. 14. VBR Exponential R1

A. Linear increase of layered video rates

In this case, source is the on-off source we described earlier. The rates for the “on” periods are 50Kbps each. With the number of layers set to 10. The average output of the source in this case is approximately 250Kbps. Results are in Fig. 12 and Fig. 13. Advantages and disadvantages are the same as those in Section VI-A but here, as we mentioned earlier, RLM is more unstable and the link utilization is low.

B. Exponential increase of layered video rates

We used 6 layers for the source. The rates for the “on” periods in each layer are, 32Kbps, 64Kbps, 128Kbps, 256Kbps, 512Kbps, and 1024Kbps. Results are in Fig. 14 and Fig. 15.

C. Hyperbolic increase of layered video rates

We used 6 layers for the source. The rates for the “on” periods in each layer are, 100Kbps, 75Kbps, 75Kbps, 50Kbps, 50Kbps, and 50Kbps. The bandwidth for R2 is increased to 150Kbps. Results are in Fig. 16 and Fig. 17.

VIII. CONCLUSION

We have presented a study of layered video fairness on IP networks. We have used RLM as an example of layered video transport protocol. Our results show that fairness is affected by the congestion control mechanism of the video protocol, by the granularity of the layers rates, and by the burstiness of the video source.

RLM (or any video transport protocol) needs to employ a congestion control mechanism that is more conformant to fairness. To achieve this, the way the protocol infers network capacity should not be based on loss. Join-experiments that RLM uses produce loss that pushes TCP to backoff. In [6], a comparison of estimated throughput and the actual received throughput is used to infer available capacity or detect congestion.

From the results in Section VI, there are advantages and disadvantages for each of the distributions used in layers rates. We recommend using the hyperbolic distribution of rates, i.e., a basic layer with the highest rate and subsequent layers with less rates. This achieves more stability of the network and in the same time with a good choice of the higher rates will allow TCP

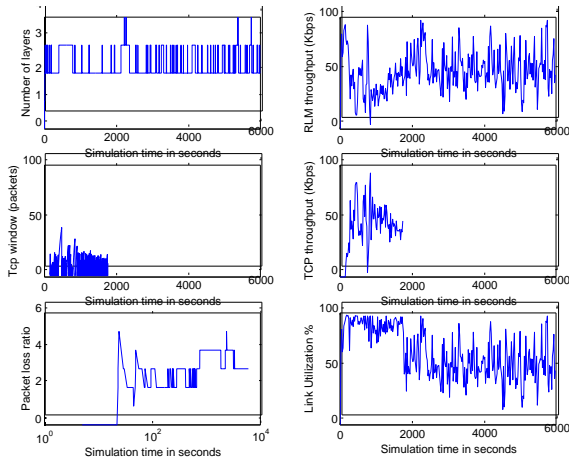


Fig. 15. VBR Exponential R2

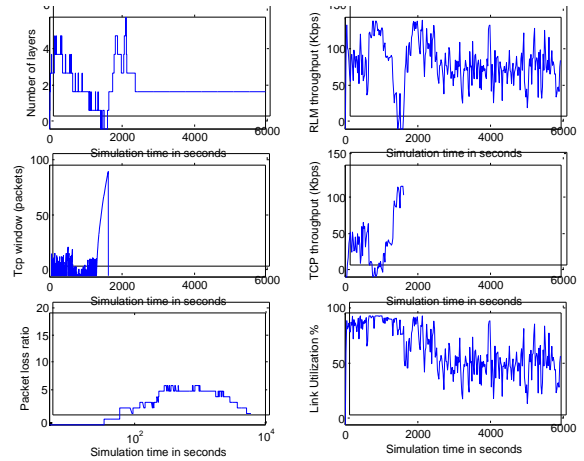


Fig. 17. VBR Hyperbolic R2

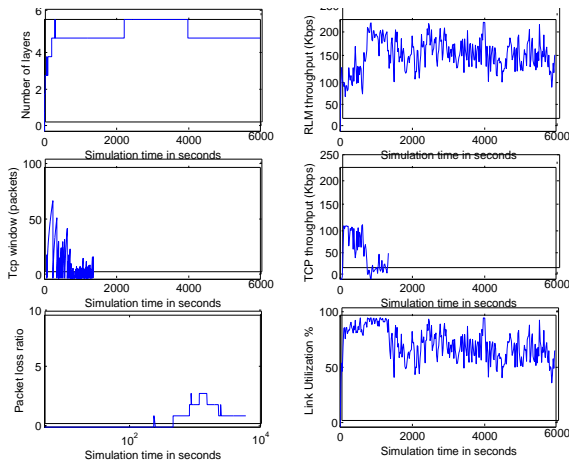


Fig. 16. VBR Hyperbolic R1

to get a better share of the bandwidth than the exponential and linear cases. The basic layer, however, should not be too big in terms of rate. Otherwise, it will prevent many receivers from subscribing to the session.

We also have shown that burstiness of the video source results in more fairness to TCP at the expense of poor link utilization and unstable video behavior.

We believe that meeting all the requirements of video multicasting, i.e., stability, scalability, fairness, utilization, and fast convergence, is hard to achieve by application adaptation approaches alone. Some network support is required. This is why we build our future work in this area on a QoS-aware network architecture such as diffserv.

REFERENCES

- [1] X. Li, M. H. Ammar, and S. Paul, "Video Multicast over the Internet?," *IEEE Network Magazine*, March/April 1999.
- [2] S. McCanne, "Scalable Compression and Transmission of Internet Multicast Video," *Ph.D. Thesis, University of California-Berkeley*, December 1996.
- [3] S. McCanne and V. Jacobson, "Receiver-driven Layered Multicast," in *Proc. of ACM SIGCOMM*, pp. 117–130, September 1996.

- [4] X. Li, S. Paul, and M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Hierarchical Rate Control," in *Proc. of INFOCOM*, pp. 1062–1072, 1998.
- [5] X. Li, S. Paul, P. Pancha, and M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes," in *Proc. of NOSSDAV*, 1997.
- [6] L. Wu, R. Sharma, and B. Smith, "Thin Streams: An Architecture for Multicasting Layered Video," in *Proc. of NOSSDAV*, 1997.
- [7] L. Vicisano, J. Crowcroft, and L. Rizzo, "TCP-like Congestion Control for Layered Multicast Data Transfer," in *Proc. of INFOCOM*, 1998.
- [8] A. Legout and E. Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes," in *Proc. of SIGMETRICS*, pp. 13–22, 2000.
- [9] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Trans. on Networking*, August 1999.
- [10] C. Albuquerque, T. Suda, and B. J. Vickers, "Fair Queueing with Feedback-based Policing: Promoting Fairness and Preventing Congestion Collapse in the Internet," in *Proc. of Global Internet Symposium*, 1999.
- [11] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, pp. 17:1–14, 1989.
- [12] J. Mahdavi and S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control," *submitted to end-2-end mailing list*, January 1997.
- [13] J. Nagle, "Congestion Control in IP/TCP Internetworks," *RFC 896, IETF*, January 1984.
- [14] D. Sisalem, F. Emanuel, and H. Schulzrine, "The Loss-delay based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme," in *Proc. of NOSSDAV*, 1998.
- [15] D. Sisalem and F. Emmanuel, "QoS Control using Adaptive Layered Data Transmission," in *Proc. of IEEE Intl. Conf. On Multimedia Computing*, pp. 4–12, 1998.
- [16] D. Towsley, J. Padhye, J. Kurose and R. Koodli, "A Model Based TCP-Friendly Control Protocol?," in *Proc. of NOSSDAV*, 1999.
- [17] "NS. UCB/LBNL/VINT Network Simulator (version 2)," <http://www-mash.cs.berkeley.edu/ns/>.
- [18] A. Matrawy, I. Lambadaris, C. Huang, and S. Archambault, "Layered Multicast Video Interaction with TCP: Congestion Control and Fairness Issues," Available via <http://www.sce.carleton.ca/faculty/lambadaris.html>.
- [19] R. Gopalakrishnan et al., "Stability and Fairness Issues in Layered Multicast," in *Proc. of NOSSDAV*, 1999.
- [20] A. Legout and E. Biersack, "Pathological Behaviors for RLM and RLC," in *Proc. of NOSSDAV*, 2000.