Rate-Based Active Queue Management with Priority Classes for Better Video Transmission

Jae Chung and Mark Claypool
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609, USA
1-508-831-5357
{goos|claypool}@cs.wpi.edu

Abstract

Video streaming on the Internet often suffers from high frame loss rates due to fragmentation of large frames and inter-frame dependencies needed for high compression. We propose adding lightweight, priority-based queue management to Internet routers to significantly improve performance of video streaming. We extend a Rate-Based RED approach to support three priority classes and apply it to MPEG. The performance of the mechanism on video streams is measured, analyzed and compared with Drop-Tail and RED in terms of transport layer throughput, system fairness, application layer throughput and goodput, and video stream quality. Extensive simulation shows that our approach improves MPEG video stream quality and network system fairness over traditional RED under a variety of workloads.

1 Introduction

The Internet has become an essential part of public life while assuming the role of the underlying communication network for multimedia streaming applications such as audio conferencing and video on demand (VOD). These relatively new applications have service requirements that are different than traditional data communication applications, such as FTP or Web browsing. Streaming media applications are less sensitive to data loss than are traditional applications but have strict constraints on end-to-end delay and jitter. Therefore, streaming media applications tend not to use TCP since it provides fully reliable data transmission with no control over transmission timings, and instead use UDP or their own transport protocols that accept some data loss for better transmission timing.

Currently, a number of Internet phone applications provide acceptable voice streaming although the quality of a voice stream is not guaranteed due to the "best effort" nature of the Internet. However, video streaming quality is typically of much lower quality than that of audio. In general, the frame loss rate for video streaming on the Internet is much higher than for audio streaming because of the large size of video frames and inter-frame dependencies inherent in video compression schemes.

Compared to voice streams in which one or more audio frames can fit into one typical IP network packet, video has very large frame sizes such that one video frame is usually broken down into several IP packets when transmitted. In such a case, one network packet loss results in an entire video frame loss, unless additional packet recovery techniques are used. Moreover, video streams have inter-frame dependencies since typical video compression makes use of temporal as well as spatial dependencies of pixels to achieve higher compression ratios. Thus, successful decoding of some compressed frames depends upon the successful decoding of other, primary frames. The loss of a primary frame will result in consecutive frame losses as seen by the user, which we call chain frame loss effect.

Video streaming will continue to suffer from high frame loss rates since the majority of router queues drop packets without knowledge on the inter-frame dependencies. Currently, video compression schemes such as MPEG, which reduce the inter-frame dependencies using redundancies within each video frame, seek to minimize the chain frame loss effect. However, the benefits to loss rates from weakening the dependencies or by adding redundancy to recover loss has the disadvantage of enlarging the frame size, which might increase the ratio between the network packet loss rate and receivers' video frame loss rate.

Improvements to video streaming quality over the Internet may be better served with more direct support from the network. Differentiated Service (DiffServ) [2] is considered to be the next generation Internet service architecture

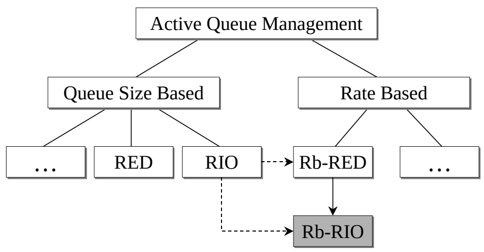

Figure 1. AQMs (shaded is proposed).

where users are provided with multiple levels of network services with different Quality of Service (QoS) or transmission characteristics. Although no agreement has been made yet on the requirements or number of services, it is likely that streaming applications may use one or more services to meet their needs. [11] illustrates an example use of prioritized service classes and proportional loss-rate differentiated classes in H.263+ video streaming.

In this paper, we propose IP router support for video streaming, which is the lowest level support from the IP network. Our approach is a part of, yet different from network service level approaches as the same network service can be supported using different queuing mechanisms at routers, which may have a significantly different impact on application and transport layer network performance. To date, there has been little research exploring router queue mechanisms to improve video streaming and evaluating the effect on transport and application performances. As a step forward in this research area, we introduce and evaluate a rate-based Active Queue Management (AQM) [3] mechanism that enables smoother, fairer video streaming.

Figure 1 shows various AQM mechanisms classified by the congestion detection mechanisms on which they are based. RED with *In/Out* (RIO) [6] is a priority-class-based queue management extension to RED [8] that fits well with video transmission characteristics. As shown in [11], classifying packets of different frame types into priority classes and dropping packets with higher dependencies (lower priority class packets) prior to dropping packets with lower dependencies (higher priority class packets) at congestion reduces user-perceived frame losses. In this work, in addition to further evaluation and application to an alternative video protocol, we identify yet another gain of deploying the priority class mechanism – reinforcing fairness among multimedia flows, with a lower bound on fairness determined by priority class packet ratios.

Rate-Based RED (Rb-RED) [7] is a rate-based AQM that is proposed as an alternative to RED. Rb-RED suggests using the estimated (or average) packet arrival rate to determine impending congestion rather than using the average queue size to achieve RED-like congestion avoidance performance. Rb-RED compares the estimated packet arrival rate with the service rate (link bandwidth) and probabilistically drops packets when the estimated arrival rate exceeds the service rate. [7] also proposes, but does not thoroughly evaluate, support for RIO-like priority traffic classes. That is, incoming packets are classified into *n*-priority classes, maintaining estimated arrival rates for each class and dropping all lower priority class packets before probabilistically dropping packets from the next priority class.

We extend Rb-RED to support 3 priority classes, which we call *Rate-Based RIO (Rb-RIO)*, and apply it to an MPEG-1 [10] stream. We implement Rb-RIO in NS (version 2) and evaluated it with our responsive (congestion-controlled) and unresponsive MPEG-1 video stream flows [4] as well as with long-lived TCP flows. Our results show that rate-based AQM, used as either a DiffServ router for prioritized service or as a stand alone "better than best effort" Internet router, improves network system fairness as well as dramatically enhances application layer goodput and video stream quality. The rest of paper describes the design of Rb-RIO and discusses the issues and analysis of using Rb-RIO on MPEG-1 streams, measured in terms of user-perceived frame loss and fairness among flows.

2 Rate-Based RIO

As discussed briefly in Section 1, Rate-Based RIO (Rb-RIO) supporting 3 priority classes for congestion marking decisions is an extention to Rate-Based RED (Rb-RED) [7] that uses estimated packet arrival rate (EAR) along with service rate (SR) to achieve RED-like congestion avoidance performance with fewer configuration parameters. We believe that rate-based AQMs that make congestion notification decisions directly based on current traffic load (EAR/SR) may better manage congestion over RED-like AQMs that make decisions based on queue size, an indirect measure of traffic load. Furthermore, rate-based AQM has advantage over queue-based approaches in that it is easier to be extended to support traffic classes.

Rb-RIO maintains the EAR, which is a weighted-average of the incoming packet rate, for each class. When a packet arrives, Rb-RIO classifies the packet and compares the combination of all class' EAR to the SR to decide whether to accept, probabilistically drop or drop with force the packet according to the algorithm shown in Figure 2. In Figure 2, *class1* is the lowest and *class3* is the highest priority class, and *TEAR* (Total EAR) is the sum of all class' EAR. Thus, all incoming packets that belong to a lower priority class are dropped when probabilistically dropping packets from the next priority class. When a packet needs to be probabilistically dropped, the drop probability P is calculated using EAR and SR, adjusted considering the average queue length and the *aggressiveness parameter* (AP) to

```
if (EAR(class1) + EAR(class2) + EAR(class1)) ≤ SR
      accept all packets
else if (EAR(class3) + EAR(class2)) ≤ SR
      accept all class3 packets
      accept all class2 packets
      drop class1 packets with probability:
        P = (TEAR – SR) / EAR(class1)
else if EAR(class3) ≤ SR
      accept all class3 packets
      drop all class1 packets
      drop class2 packets with probability:
        P = (EAR(class3) + EAR(class2) – SR) / EAR(class2)
else
      drop all class1 packets
      drop all class2 packets
      drop class3 packets with probability:
        P = (EAR(class3) – SR) / EAR(class3)
end if
```

Figure 2. Rb-RIO Algorithm from [7]

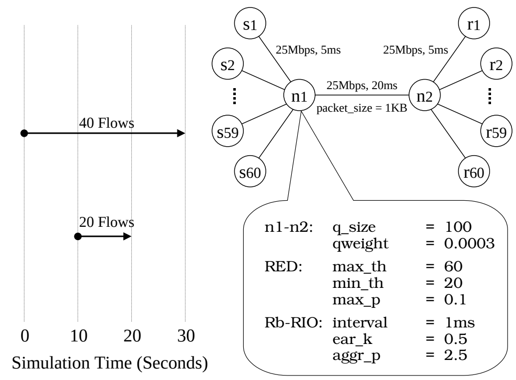

Figure 3. Simulation Network Setup.

keep the average queue within a target range (the middle of the physical queue, by default), and applied with a uniform drop distribution [7].

Dropping all lower priority class packets before starting to drop the next priority class packets is a desired feature for multimedia transmissions, especially for video where successfully decoding some types of frames depends upon first successfully decoding other frames. For example, MPEG [10] encodes video at a given frame rate and picture quality, generating a stream of frame types I, P and B, such as IBBPBBPBB. Among the three frame types, only I-frames can be decoded on their own. The decoding of a B-frame relies on a pair of I-frames and/or P-frames that come before and after the B-frame and the decoding of a P-frame relies on an I-frame or P-frame that comes before the P-frame. I-frame loss results in consecutive user perceived losses on dependent P- and B-frames and similarly a P-frame loss results in a user-perceived loss for the dependent B-frames and possibly the next P-frame. Therefore, dropping B-frames prior to P-frames and P-frames prior to I-frames can drastically reduce user-perceived frame loss. In addition, large frames (typical I- and P- frames) are often fragmented into multiple packets at the IP layer, resulting in a sharp increase in perceived-loss even when there is little network congestion. In the following sections, we test Rb-RIO with an MPEG-1 application [4] by mapping the I-, P- and B-frames to each priority class (class3, class2 and class1, respectively).

Another potential benefit of Rb-RIO is that it guarantees a minimum transmission rate for each flow when packets from a flow are distributed to the priority classes. This provides a lower bound on fairness among flows determined by priority class packet ratios. Even if there becomes a widely accepted responsive network protocol for multimedia, unfairness among flows will still be an issue as shown in [1]. Currently, there is no widely accepted network protocol for multimedia that supports congestion control, and each multimedia application typically uses customized congestion control mechanisms or uses no congestion control at all. In this situation, we believe that router mechanisms that improve fairness are a desirable feature.

One last issue is whether Rb-RIO can be used only as a DiffServ router or whether Rb-RIO can stand alone as an effective “better than best effort router”. Although it requires policy decisions, we also show potential benefits of using Rb-RIO as a stand alone router queue mechanism by randomly assigning TCP packets to classes, while using the above priority class mappings for video packets.

3 Simulation

We ran 3 sets of simulations to measure performance of Rb-RIO: TCP flows, multimedia video streams, and mixed (TCP + video) flows. We measure transport layer throughput, fairness and application layer goodput for video, compared with that of Drop-Tail and a well-configured RED router. We assign TCP packets to class3 and class2 in a 7:3 ratio based on a likely policy that TCP packets are as important as I- and some of P-frame packets of MPEG video streams. For all simulations, we set the maximum IP packet size to 1Kbyte for all traffic agents, and use New-Reno for the TCP agents. Also, we set the physical queue size of network routers to 100 packets. First, we ran 60 FTP-TCP flows on Drop-Tail, RED and on Rb-RIO to compare the performances of just TCP (the typical dominant traffic in Internet routers today) as shown in Figure 3. For RED, we use the parameter settings shown in Figure 3, which are well-tuned RED settings for the simulated network setup and the TCP workload. Rb-RIO parameters are chosen from [7], except for the aggressiveness parameter that is set to 2.5 instead of 2.0 to keep the average queue size at around 40 packets (the middle of the minimum and maximum thresholds for RED), are also shown in Figure 3. The simulation starts with 40 TCP flows and at 10 seconds, 20 more TCP

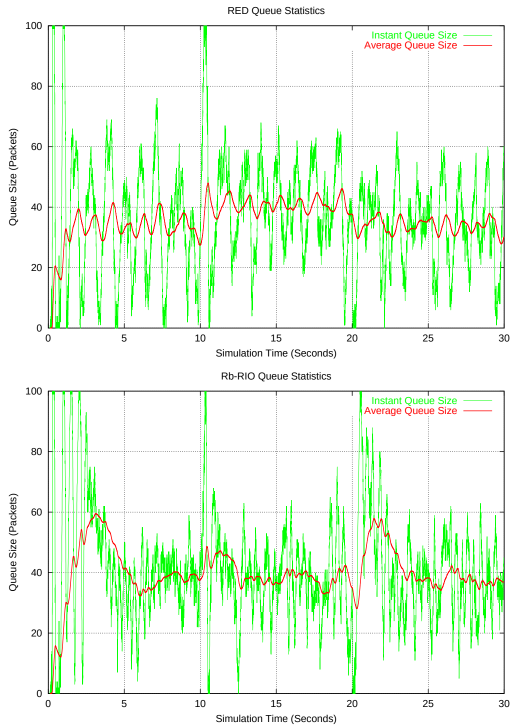

Figure 4. Queue Statistics of RED (top) and Rb-RIO (bottom) with TCP Flows.

flows join making the system more congested.

After analyzing the performance of Rb-RIO on TCP flows, we replaced TCP traffic sources with responsive MPEG-1 traffic generators called MPEG_APP (described in [4] in detail) and ran the simulations again on RED and Rb-RIO. In this simulation, MPEG_APP uses a 30 frame per second IBBPBBPBB pattern stream in which the sizes of the I-, P- and B-frames are 11 KB, 8 KB and 2 KB, respectively. These frame sizes are the mean frame size of each type obtained while playing an 1.1 Mbps MPEG-1 news clip. In the MPEG simulation, we turn off the congestion control mechanism for the 20 streams that come into system from 10 to 20 seconds making them always transmit at their highest frame rate. This was to measure how fairly Drop-Tail, RED and Rb-RIO manage multimedia flows with different congestion responsiveness characteristics, and to measure user perceived frame loss differences that Rb-RIO offers to well- and ill-behaving video streaming applications.

For the last set of simulations, we start 20 FTP-TCP and 20 responsive MPEG-1 flows (0-30 seconds) and start 20 unresponsive MPEG-1 flows at the 10 seconds that run until 20 seconds. i.e., we replace the 20 responsive MPEG-1 flows of the second set of simulations with FTP-TCP flows to measure the performance of Rb-RIO on mixed traffic and to test the feasibility of using it as a stand alone "better than best effort" router queue mechanism.

Periods (Seconds)	Queue Management	Link Utilization (Mbps)	Drop Rate (%)
0–10	Drop-Tail	24.230	2.87
	RED	24.332	2.88
	Rb-RIO	24.367	2.98
10–20	Drop-Tail	24.707	3.41
	RED	24.993	3.51
	Rb-RIO	24.953	3.68
20–30	Drop-Tail	24.724	2.31
	RED	24.929	2.25
	Rb-RIO	24.918	2.27

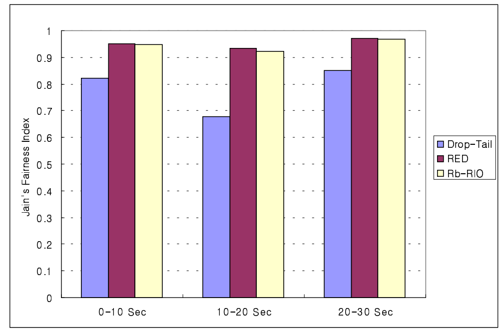

Figure 5. Link Utilization, Drop Rate (top) and Jain's Fairness (bottom) for TCP Flows.

4 Results and Analysis

In this section, we compare the performance of Rb-RIO on long-lived TCP flows with that of Drop-Tail and RED to verify our Rb-RIO implementation on NS and to evaluate the potential benefit of using a rate-based AQM mechanism. Then we exhibit an in depth analysis of Drop-Tail, RED and Rb-RIO performance on MPEG-1 video streams, followed by analysis of the performance of Rb-RIO on the traffic mix of TCP flows and video streams.

4.1 Rb-RIO with TCP Flows

As presented in Section 3, 40 TCP flows ran from 0 to 30 seconds and 20 more TCP flows came into the system at 10 seconds and left at 20 seconds. For Rb-RIO, the results shown in this section is for a 7:3 random TCP mapping to class3 and class2. We also ran the same simulation with mapping TCP only to class3, however the results was very similar to that of 7:3 mapping. Figure 4 shows the congested RED and Rb-RIO router queue statistics. The RED statistics, where the average queue size was well under the maximum threshold (60) and well over the minimum threshold (20), and the instant queue size seldom hit the maximum queue limit nor 0, show that RED was well-configured for the TCP workload. The corresponding Rb-

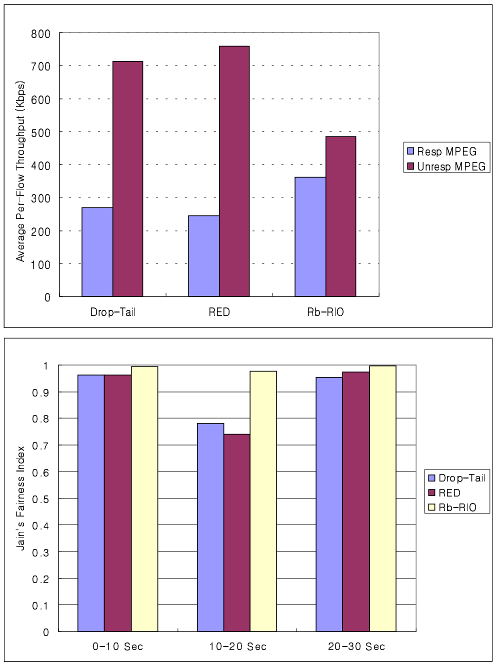

Figure 6. Average Per-Flow Class Throughput (top, 10–20 Seconds) and Jain's Fairness (bottom) for MPEG Streams.

RIO queue statistics show that the router was also operating in a similarly "healthy" condition. Note that Rb-RIO's congestion detection and marking probability is calculated primarily based on EAR and SR, and the average queue size is used to slightly adjust the marking probability to keep the queue size at a desired position (40 packets in this case). An interesting observation is that the RED queue oscillates periodically while the rate-based mechanism does not, resulting in smoother average queue movement. Although further investigation is required, this could be evidence suggesting an advantage of using packet arrival rate to control congestion rather than just average queue size.

Figure 5 shows the congested link utilization, network packet drop rate and the system fairness of Drop-Tail, RED and Rb-RIO for periods 0–10, 10–20 and 20–30 seconds. For system fairness measurement, we use Jain's Fairness Index [9], which takes the average throughput for all flows and gives a normalized number between 0 and 1, where 0 indicates the greatest unfairness and 1 indicates the greatest fairness. For all periods, the link utilization of the well-configured RED and Rb-RIO were comparable with each other and were slightly better than that of Drop-Tail, while the packet drop rate for the three queuing mechanisms were comparable. According to the Jain's index, RED and Rb-RIO were nearly equally fair, while Drop-Tail system was much more unfair than RED or Rb-RIO. Although additional evaluation with different workloads and network configurations is required, our result suggests that Rb-RIO (or Rb-RED in a broader sense) has performance comparable with well-configured RED for a TCP-only workload.

4.2 Rb-RIO with MPEG Streams

Next, we show the results and analysis of the MPEG simulation. Figure 6 shows the average per-flow throughput of the responsive and unresponsive MPEG-1 flows during 10–20 seconds, and Jain's fairness index for Drop-Tail, RED and Rb-RIO for the three time periods. During periods 0–10 and 20–30 when only responsive MPEG video flows are active, all three queue mechanisms provide reasonably fair allocation of outgoing bandwidth although Rb-RIO provides the best fairness of the three. However, during the 10–20 second period when the 20 unresponsive MPEG-1 video flows arrive, Drop-Tail and RED's fairness degrade significantly as all the unresponsive flows get a larger share of the bandwidth than the responsive flows. Rb-RIO, however, only suffers a slight degradation in fairness as it protects the class2 and class3 packets (the P- and I-frames, respectively) and drops most of the class1 packets (the B-frames) from the unresponsive flows.

Many continuous media streaming protocols do not provide for a mechanism for retransmission, since interactive multimedia is often time sensitive and large delay buffers can be required to wait for a re-transmitted packet. Since the successful decoding of P- and B-frames depends upon the successful arrival of other P- and I-frames, the effects of packets dropped by a router can be compounded when decoded into the frames played to the user. Furthermore, most P- and I-frames are fragmented by the router since they are larger than a default IP packet (1 Kbyte in our simulation), further exacerbating the problem.

Table 1 shows aggregated throughput and loss rates of the application layer as well as of the transport layer for the 40 responsive MPEG-1 flows from each experimental

Table 1. Statistics for 40 Responsive MPEG Streams (0–30 Seconds).

	Throughput (Mbps)	Frms Recvd	Deco Frms	Pkt Loss%	Frm Loss%	Deco Loss%
Drop-Tail	19.819	10369	7620	13.08	22.8	43.2
RED	18.650	7706	5054	8.99	29.4	53.7
Rb-RIO	20.812	7553	7211	11.19	36.9	39.8

Table 2. Statistics for 20 Unresponsive MPEG Streams (10–20 Seconds).

	Throughput (Mbps)	Frms Recvd	Deco Frms	Pkt Loss%	Frm Loss%	Deco Loss%
Drop-Tail	14.272	3616	768	23.4	38.7	87.0
RED	15.182	3331	467	20.6	43.5	92.1
Rb-RIO	9.696	656	656	34.5	88.8	88.8

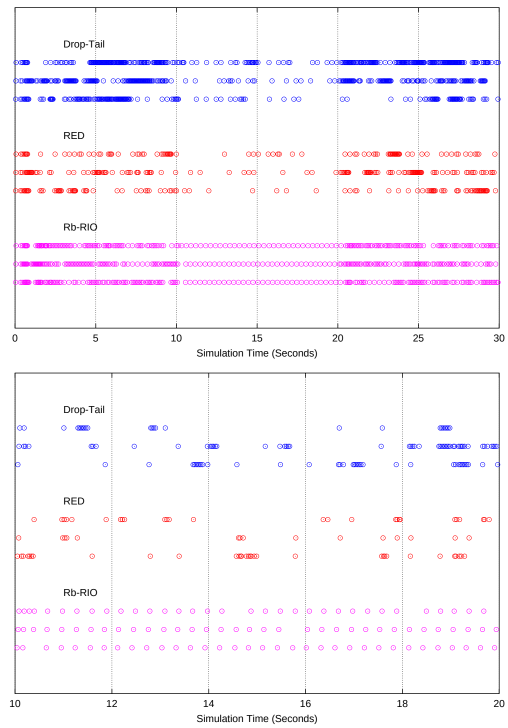

Figure 7. User Perceived Frames for 3 Sampled Responsive (top, 0–30 Seconds), and Unresponsive (bottom, 10–20 Seconds) MPEG Streams.

run. The transport layer throughput indicates that Rb-RIO delivered the largest number of bytes for the responsive flows with Drop-Tail being the second biggest. Looking at the application layer throughput (Frms Recvd) and goodput (Deco Frms), while Drop-Tail delivered the largest number of video frames, it was using the network bandwidth ineffectively since about 27% of the frames received were not decodable (10369 vs. 7620 frames). Similarly, with RED, about 34% of the frames received were not decodable (7706 vs. 5054 frames). Moreover, the application layer throughput and goodput of RED (10369 vs. 7706 frames for throughput and 7620 vs 5054 frames for goodput) was significantly worse than that of Drop-Tail, although the transport layer throughput differences were not significant. This indicates that video steaming will suffer even more under RED queue management than under Drop-Tail, since the uniform packet drop distribution resulting from the random early packet drop behavior of RED will result in more application frame losses.

This phenomenon of RED using network bandwidth less effectively than Drop-Tail can also be seen by comparing the network packet loss rate, video frame loss rate and decodable video frame loss rate of Drop-Tail and RED. Rb-RIO, receiving the least number of frames, used network bandwidth most effectively as almost all video frames received were decodable, while the network bandwidth usage and the number of decodable frames received by Drop-Tail and RED were comparably poor. In addition, Rb-RIO provided significant benefit to the video application by producing far smoother video streams while Drop-Tail and RED provided more broken video streams as shown in Figure 7. In Figure 7, the top 3 lines depict the sequence of successfully decoded frames of 3 sampled MPEG video streams under Drop-Tail. Similarly, the middle 3 and the bottom 3 lines are 3 successfully decoded frames under RED and Rb-RIO respectively.

The benefits of using Rb-RIO is even more dramatic for the unresponsive video streams as depicted in Table 2. By dropping all the lower priority class packets (B- and P-frame packets) first, Rb-RIO used only 68% of the bandwidth used by Drop-Tail for the 20 unresponsive MPEG video flows (14.272 vs. 9.696 Mbps) and achieved an 85% application layer goodput of Drop-Tail (768 vs. 656 decodable frames). Furthermore, Rb-RIO provided far smoother video streams than either Drop-Tail or RED.

4.3 Rb-RIO with TCP Flows and MPEG Streams

In the previous section, we showed the benefit of using Rb-RIO for video streaming only traffic. In this section, we evaluate a mechanism to use Rb-RIO for mixed traffic with both TCP flows and MPEG video streams. This test is not a complete evaluation of feasibility of using Rb-RIO as a “better than best effort” router queue management, yet it is still useful since it shows possible benefits for TCP as well as MPEG video streams over Drop-Tail and RED queue management.

Figure 8 depicts fairness for the TCP, responsive and unresponsive MPEG flows in terms of aggregated class throughput and Jain’s fairness index. During periods 0–10 and 20–30 seconds, where 20 FTP-TCP and 20 responsive MPEG video flows were competing for the bandwidth, respectively, the responsive MPEG flows got significantly more bandwidth than the TCP flows did under Drop-Tail management. RED shows fairer bandwidth allocation, however, as the MPEG flows still got more bandwidth. In contrast, Rb-RIO mapping TCP packets to the highest and middle priority classes in 7:3 ratio, achieved fairer bandwidth allocation than Drop-Tail and RED while favoring TCP slightly more than the MPEG video steams.

During 10–20 seconds, when 20 unresponsive MPEG streams join, Drop-Tail could not protect TCP, the least aggressive of the three flow types. RED was able to allocate a little more bandwidth to the TCP flows, yet still could not protect the responsive flows from the unresponsive flows. In contrast, Rb-RIO was able to best protect the responsive flows, and TCP flows received more bandwidth than they

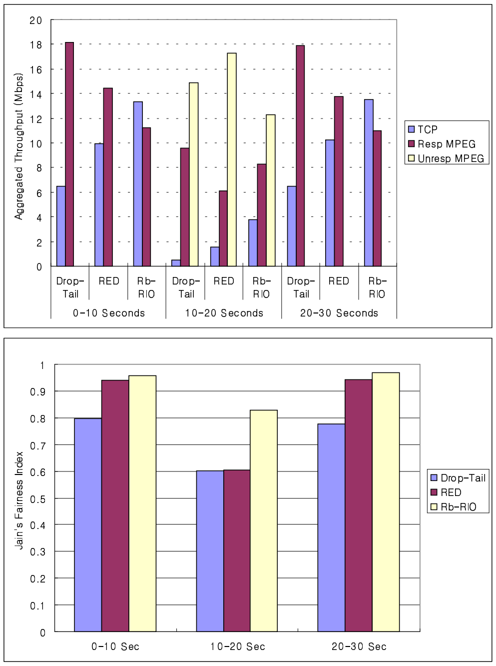

Figure 8. Aggregate Class Throughput (top) and Jain's Fairness (bottom) for TCP and MPEG Traffic Mix

did under Drop-Tail or RED. Rb-RIO still provided a far smoother video streams than Drop-Tail and RED, similar to Figure 7, but omitted here due to lack of space.

We note that the 7:3 TCP mapping we chose could be replaced with other TCP mappings such as mapping to only the highest priority class, which would favor TCP the most. In our case, we chose the 7:3 mapping since it produces reasonable performances for both the TCP and the video streams. How to assign bandwidth among TCP and non-TCP flows is a policy decision, and Rb-RIO may be used to effectively enforce the policy chosen.

5 Conclusion

In this paper, we have presented Rb-RIO, a specific implementation of Rb-RED that uses a priority class-based AQM, and evaluated it with TCP and video traffic. We find that Rb-RIO can dramatically improve MPEG video frame loss as perceived by the user, while improving fairness over Drop-Tail and RED. We also identify how Rb-RIO can be configured to outperform Drop-Tail and RED on traffic with both TCP and video streams. Rb-RIO could be used for a video transmission service class queue mechanism in IETF's DiffServ architecture, or it could be used as a standalone mechanism within the current "best effort" Internet service architecture.

We believe that the benefits we showed in this paper for video streams are not the only benefits that Rb-RIO, or priority class-based AQM in a larger sense, could offer. Future work includes applying Rb-RIO to improve the performance of short-lived Web TCP traffic (mice) over long-lived TCP bulk transfers (elephants). Assigning short-lived TCP packets to the highest priority class trying to avoid drops may improve the performance of Web TCP traffic. Other future work may compare the performance of multimedia and TCP flows under priority class-based AQM such as Rb-RIO to that of traffic class-based AQM such as Dynamic Class-Based Threshold (D-CBT) [5]. Future work also includes a more thorough evaluation of rate-based versus average queue size-based AQM.

References

[1] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic Behavior of Slowly Responsive Congestion Control Algorithms. In *Proceedings of Sigcomm 2001*, August 2001.

[2] Y. Bernet, J. Binder, S. Blake, M. Carlson, B. E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss. A Framework for Differentiated Services. Internet Draft draft-ietf-diffserv-framework-02.txt, IETF, February 1999.

[3] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC-2309, April 1998.

[4] J. Chung and M. Claypool. Better-Behaved, Better-Performing Multimedia Networking. In *Proceedings of SCS Euromedia*, May 2000.

[5] J. Chung and M. Claypool. Dynamic-CBT - Better Performing Active Queue Management for Multimedia Networking. In *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 2000.

[6] D. Clark and W. Fang. Explicit Allocation of Best-Effort Service. *IEEE/ACM Transactions on Networking*, Aug. 1998.

[7] S. DeCnodder, K. Pauwels, and O. Elloumi. A Rate Based RED Mechanism. In *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, June 2000.

[8] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, Aug. 1993.

[9] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, Inc., New York, NY, 1991.

[10] J. Mitchell and W. Pennebaker. *MPEG Video: Compression Standard*. Chapman and Hall, 1996. ISBN 0412087715.

[11] J. Shin, J. Kim, and C.-C. J. Kuo. Content-Based Packet Video Forwarding Mechanism in Differentiated Service Networks. In *Proceedings of Packet Video 2000*, May 2000.