Router Support for Better Video Transmission

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

ABSTRACT

Multimedia streaming is difficult on the Internet, especially for video due to the large frame size and inter-frame dependencies resulted from high compression rates. This paper shows that router support for priority-based queue management could significantly improve performance of video streaming. We extend Rate-Based RED (Rb-RED) to Rate-Based RIO (Rb-RIO) which support three priority classes and apply it to MPEG. The performance of the mechanism on the video streams is measured, analyzed and compared with RED in terms of transport layer throughput, video stream quality and system fairness through simulation using NS. The study shows that Rb-RIO improves MPEG video stream quality and network system fairness.

1. INTRODUCTION

The Internet has moved from a data communication network for a few privileged professions to an essential part of public life similar to the public telephone networks, while assuming the role of the underlying communication network for multimedia streaming applications such as Internet phone, video conferencing and video on demand (VOD). These relatively new applications have different service requirements from traditional data communication applications. That is streaming media applications are less sensitive to data losses but have tight timing constraints such as end-to-end delay and jitter (or delay variance), which, as well as data loss rate and distribution, directly affect user perceived quality of the media [10]. Therefore, streaming media applications tend not to use TCP that provides reliable data transmission with no control over transmission timings, but use UDP and/or their own transport protocols that compensates some date loss for transmission timings. Also, they often use data repair and correction techniques on the application layer to minimize the effect of data loss.

Currently on the Internet, voice streaming is practically in use and quite a number of Internet phone companies are operational although the quality of voice stream is not guaranteed due to the "best effort" nature of the Internet service. However, video streaming is hardly practical to date. In general, the frame loss rate for video streaming on the Internet is much higher than for audio streaming. Although there could be many others, the two main reasons we see are the large size of video frames and inter-frame dependencies resulted from video compression schemes.

Compared to voice streams in which one or more frames can be fit into one network packet, video has very large frame sizes such that one video frames are usually broken down to several packets within the transport layer and transmitted. In such a case, one network packet loss results in a whole video frame loss assuming no special packet recovery technique is used. Moreover, video streams have inter-frame dependencies that result from video compression mechanisms such as MPEG that make use of temporal as well as special dependencies of pixels to achieve higher compression ratios. In other words, successful decoding of some compressed picture frames depends upon the successful decoding of other frames. This implies that loss of a primary frame will result in consecutive frame losses, which we call chain frame loss effect.

Under the current Internet environment, where majority of router queue management mechanisms are simple Drop-Tail mechanisms that have no knowledge on the inter-frame dependencies, we suspect that video streaming will continue to suffer from high frame loss rate as long as inter-frame dependencies of video frames exist. Currently, video compression schemes such as MPEG-2, which reduce the inter-frame dependencies using redundancies within each video frame, were proposed to minimize the chain frame loss effect. However, the loss rate gain from weakening the dependencies is not promising, since this would enlarge the frame size, which might increase the ratio between the network packet loss rate and receiver end's video frame loss rate.

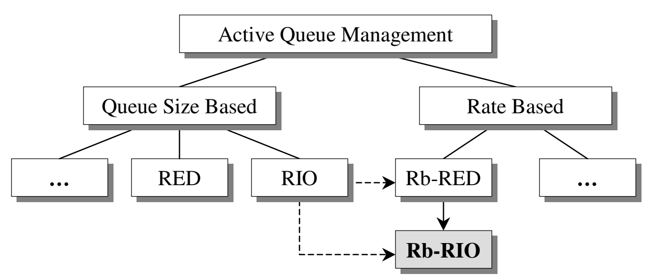

Figure 1. AQM Mechanisms (shaded is proposed)

This paper shows and evaluates how a priority-based Active Queue Management (AQM) [2] mechanism could be used to improve the performance of the Internet on streams with inter-frame dependencies, especially on MPEG video streams. Figure 1 shows various AQMs classified by the congestion detections mechanisms they are based on, where RIO [6] is a priority-classes-based queue management extension to RED [7] and Rate-Based RED (Rb-RED) [5] is a rate-based AQM approach that is proposed as an alternative to RED, which also proposes to support priority class based management.

Rb-RED, suggests using estimated (or average) packet arrival rate to determine impending congestion rather than using average queue size to achieve RED like congestion avoidance performance with fewer configuration parameters. It compares the estimated packet arrival rate with the service rate (available bandwidth) and probabilistically drops packets when the estimated arrival rate exceeds the service rate. The mechanism also proposes support for RIO like priority traffic classes. That is, classify incoming packets into *n*-priority classes, maintaining estimated arrival rates for each class and drop all lower priority class packets before probabilistically dropping packs from the next priority class.

The priority classification mechanism fits well with multimedia transmission characteristics, especially with video transmission, where inter-frame dependencies exist as discussed earlier. Therefore, 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 should reduce user-perceived frame loss dramatically. Yet, another gain of deploying our priority class mechanism is that it reinforces fairness among multimedia flows with upper bounds determined by priority class packet ratios.

To test these hypotheses, we extended Rb-RED to support 3 priority classes, which we call *Rate-Based RIO (Rb-RIO)*, and applied it to an MPEG-1 [9] stream. We implement Rb-RIO in NS (version 2) [11] to simulate a variety of IP networks including the RED queue mechanism. This paper verifies the comparability of the rate-based congestion avoidance mechanism of Rb-RIO (also Rb-RED) on TCP flows with that of RED in terms of link utilization and fairness among flows. Then, it describes the issues and analysis of using Rb-RIO on MPEG-1 streams, measured in terms of user-perceived frame loss and fairness among flows.

The rest of this paper is as follows: Section 2 describes Rb-RIO in detail; Section 3 describes the simulation setup; Section 4 analyzes and evaluates Rb-RIO; Section 5 summarizes our conclusions.

2. RATE-BASED RIO (Rb-RIO)

As discussed briefly in Section 1, Rate-Based RIO (Rb-RIO) supporting 3 priority class for multimedia is a specific case of Rate-Based RED (Rb-RED) [5] that uses estimated packet arrival rate (EAR) along with service rate (available bandwidth) to determine congestion rather than using average queue size and thresholds. The objective behind the mechanism is to achieve RED-like congestion avoidance performance with fewer configuration parameters and support for multiple priority classes with different early congestion packet drop rates.

Rb-RIO maintains EAR, which is a weighted-average of periodical incoming packet rate, for each class. In the original design, each class EAR is updated on each packet arrival event, and the average weight factor is dynamically calculated considering the period between last EAR update and current time as follows:

$$EAR_{new}(c,t) = \left(1 - e^{-T/K}\right)\frac{L}{T} + e^{-T/K} EAR_{old}(c,t)$$

where, c is priority class, t is current time, K a constant (typical values for K are 0.1, 0.5, or 1.0 second), L the packet size, and T is the time between the current time and the last update of the class EAR [5]. In our Rb-RIO implementation, we fixed the EAR update time interval using a timer and updated the EAR of all classes at the same time to reduce computational complexity. This also reduces implementation complexity, since it is not necessary to consider events without packet arrivals for one or more priority classes, where the original mechanism updates the class EAR.

When a packet arrives, Rb-RIO classifies the packet and compares the combination of all class' EAR to the service rate (SR), which can vary over time, to decide whether accept, probabilistically drop or drop with force the packet according to the following algorithm:

```
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 class 1 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 [5]

where, *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, P is calculated as above, adjusted considering the average queue length and *aggressiveness parameter* (AP), and applied with a uniform drop distribution [5]. Note that Rb-RIO also keeps track of the average size of the outbound queue, in order to, different from RED, adjust the drop probability (i.e. higher average queue sizes result in higher drop probability).

Dropping all lower priority class packets before starting to drop next priority class packets is a desired feature for multimedia transmissions, especially for videos where successfully decoding some types of frames depends upon first successfully decoding of other frames. For example, MPEG-1 [9] encodes video at a given frame rate and picture quality, generating a stream of frame types I, P and B, associated with a typical Group of Pictures (GOP), 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. This implies that an I-frame loss results in consecutive user perceived losses on dependent P and B-frames and similarly a P-frame loss will result 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 sequential user-perceived frame loss. In addition, large frames (typical I- and P- frames) can be fragmented into multiple packets at the IP layer, resulting in a sharp increase in perceived-loss even when there is little network congestion. We have tested Rb-RIO with an MPEG-1 application [3] by mapping the I-, P- and B-frames to each priority class (class3, class2 and class1).

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 an upper bound on fairness among flows determined by priority class packet ratios. Currently, there is no widely accepted network protocol for multimedia that supports flow control in general, and each multimedia application typically uses customized flow control mechanisms or uses no flow control at all. In this situation, we believe that router support for fairness is a desirable feature. Even if there comes a widely accepted responsive network protocol for multimedia, unfairness among flows will still be an issue, as the congestion responsiveness of a flow would be affected by end-to-end delay.

3. SIMULATION

We ran a set of simulations to measure effects of Rb-RIO on multimedia video streams in terms of fairness and user perceived frame loss, and compared it with that of RED. First, we ran 60 FTP-TCP flows through once on RED and once on Rb-RIO router to verify our Rb-RIO implementation and measure its performance on TCP flows as shown in Figure 3. We used the RED parameters as shown in Figure 3. Rb-RIO parameters chosen from [5] are also shown in the figure. The simulation started with 40 TCP flows and later at 10 seconds 20 more flows joined making the system more congested with a maximum IP packet size to 1Kbyte.

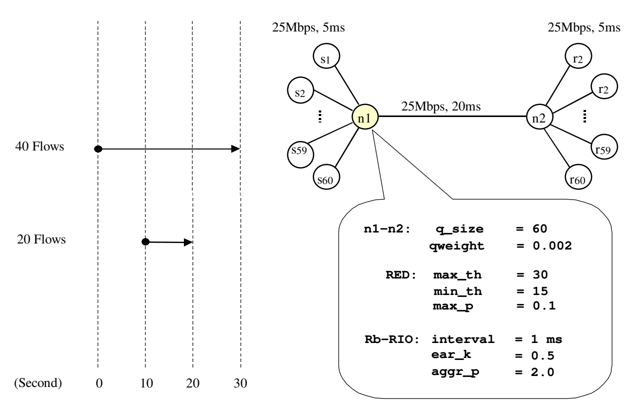

Figure 3. Simulation Scenario and Network Setup

After analyzing the performance of Rb-RIO on TCP flows, we replaced TCP traffic sources with MPEG-1 traffic generators called MPEG_APP of which behavior are described in [3] in detail, and ran the simulations again on RED and Rb-RIO. In short, MPEG_APP is trace driven traffic generator that runs on UDP and simulates MPEG-1 client-server video application. It has the same 5 level multiplicative decrease additive increase (MDAI) flow control mechanism as MM_APP, and supports 2 common MPEG-1 GOP patterns that are IBBPBBPBBI and IBBPBBPBBPBBI. Figure 4 shows how MPEG-APP maps each scale level to an MPEG-1 transmission policy considering frame dependencies.

Scale	Transmission Policy
4	I B B P B B P B B I
3	I B P B P B I
2	I P P I
1	I P I
0	I I

Figure 4 MPEG Transmission Policy Associated with Scale Values (based on IBBPBBPBBI GOP pattern)

In this simulation, MPEG_APP used a 30 frames 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 a short high quality MPEG-1 news clip.

One thing to note in the MPEG simulation setup is that we turned off the flow control mechanism for the 20 streams that come into the system from 10 to 20 seconds making them transmit at their highest frame rate. This was to measure how fairly RED and Rb-RIO manages multimedia flows of different congestion responsiveness characteristics, and to measure the user perceived frame loss differences that Rb-RIO offers to well and ill-behaving video streaming applications.

4. RESULTS AND ANALYSIS

Before discussing the simulated results and analysis, we briefly present Jain's index [8] that is used to measure the fairness among individual flows in this paper.

$$f(x_0, x_1, x_2, \cdots, x_n) = \frac{\left(\sum_{i=0}^{n} x_i\right)^2}{n \sum_{i=0}^{n} x_i^2}$$

Figure 5. Jain's Fairness Index Equation

Figure 5 shows the formula that calculates Jain's fairness, which gets the average throughputs of the flows (x_i) of which the fairness is measured as an input, and produces a normalized number between 0 and 1, where 0 indicates the greatest unfairness and 1 indicates the greatest fairness.

4.1 Rb-RIO with TCP Flows

This section presents the performance of the Rb-RIO mechanism with TCP flows. As presented in Section 3, 40 TCP flows ran from 0 to 30 seconds and 20 more TCP came into the system at 10 seconds and left at 20 seconds.

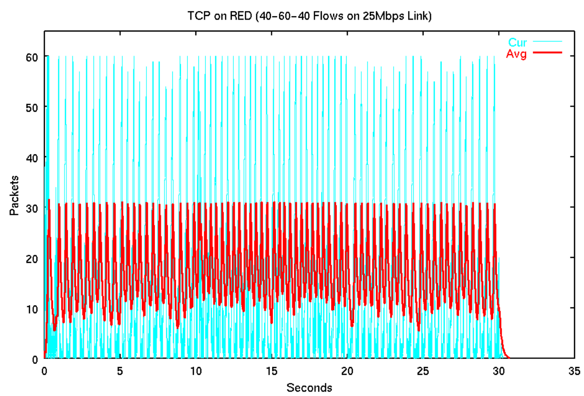

Figure 6 (a). RED Queue Behavior on TCP Flows

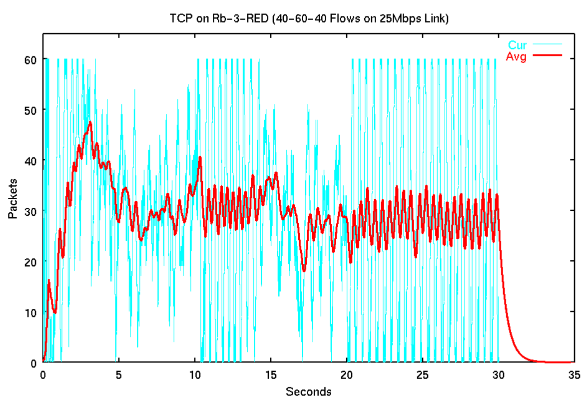

Figure 6 (b). Rb-3-RED Queue Behavior on TCP Flows

Figure 6 (a) and (b) shows the congested router's RED and Rb-RIO queue behavior. Notice the queue length in RED reaches the maximum threshold (set to 30 packets in this simulation) quite often causing forced sequential packet drops, which may cause TCP flows back off to a slow start state or cause a timeout in the worst case. In fact, Rb-RIO has slightly higher congested link utilization than RED as shown in Table 1.

Table 1. Congested Link Throughput (TCP)

	0 ~ 10 Sec	10 ~20 Sec	20 – 30 Sec
RED	22.7 Mbps	23.9 Mbps	23.3 Mbps
Rb-RIO	24.5 Mbps	24.7 Mbps	23.8 Mbps

In Figure 7 Jain's Fairness index shows that the two systems were about equally fair to TCP flows. Indeed, we ran several simulations of RED with changing minimum and maximum thresholds, and concluded that the performances of both mechanisms were very much comparable for a relatively small number of flows.

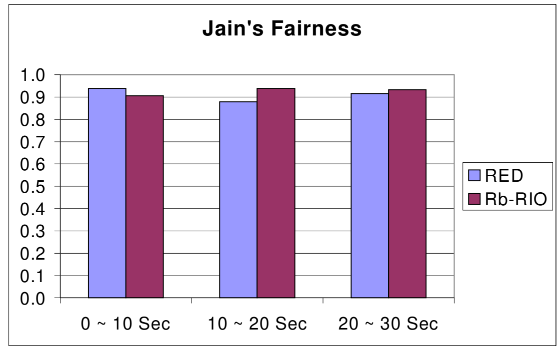

Figure 7. Jain's Fairness Comparison on RED and Rb-RIO with TCP Flows

4.2 Rb-RIO with MPEG-1 Streams

We next analyze the performance of Rb-RIO when all the flows are MPEG-1 video streams instead of bulk-transfer TCP streams. Figure 8 depicts Jain's Fairness Index for both RED and Rb-RIO for the three time periods, 0-10, 10-20 and 20-30 seconds.

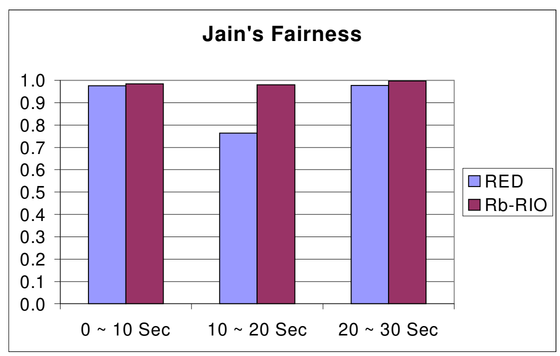

Figure 8. Jain's Fairness Comparison on RED and Rb-RIO with MPEG-1 Flow

During periods 0-10 and 20-30 when only responsive MPEG-1 video flows are active, both RED and Rb-RIO equally provide fair allocation of outgoing bandwidth. However, during the 10-20 second period when the 20 unresponsive MPEG-1 video flows arrive, RED's fairness degrades 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 guarantees that 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 time sensitive and large buffers can be required to wait for a 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 frames played to the user. Furthermore, most P- and I-frames must be fragmented by the router since they are larger than a default IP packet, further exacerbating the problem.

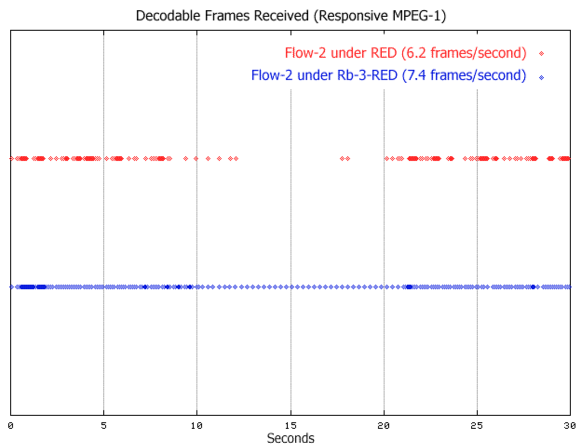

Figure 9 (a). User Perceived Frames for a Sampled Responsive MPEG flow (0 ~ 30 Seconds)

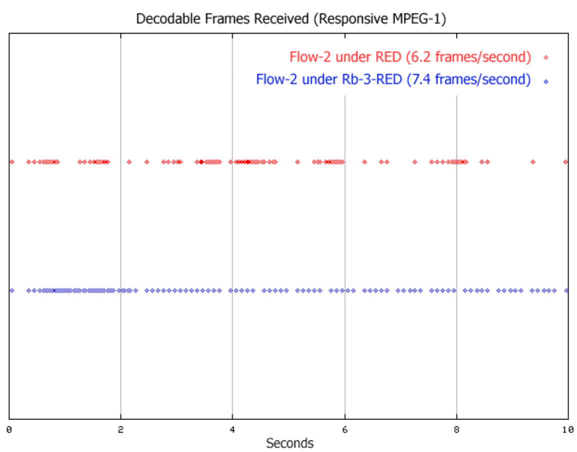

Figure 9 (b). Zoom (0 ~ 10 Seconds) of Figure 9 (a)

Figure 9 shows the loss rate that users of a responsive MPEG-1 flow would perceive given the packets dropped by the router (Figure 9b is a zoom in of the first section of Figure 9a). The top line depicts the sequence of successfully decoded frames under RED. The bottom line depicts the sequence of successfully decoded frames under Rb-RIO. Notice that Rb-RIO provides a far smoother video stream, as indicated by the near continuum of points along the bottom line, while RED provides a more broken stream of video, as indicated by the frequent and often large gaps in the top line. The situation is even more dramatic for unresponsive flows, as depicted in Figure 10. For an unresponsive flow, there are very few successfully decoded frames using a RED router, as shown by the few dots at the top, while, under the same conditions, Rb-RIO provides a nearly unbroken stream of consecutive frames.

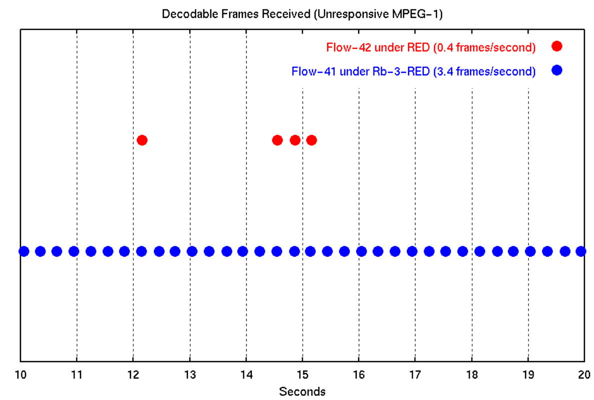

Figure 10. User Perceived Frames for a Sampled Unresponsive MPEG flow (10 ~ 20 Seconds)

Table 2 shows additional analysis of one sampled responsive MPEG flow from each experimental run. With RED, the flow's bandwidth dipped precipitously to 196 Kbps during period 10-20 and recovered once the unresponsive flows ceased. With Rb-RIO, the flow's bandwidth does not drop nearly as much during period 10-20. Notice that with RED, more frames are received (265 vs. 223) but that Rb-RIO has far more “good” frames that could be successfully decoded (222 vs 185, or 30% vs 0.4%) and played because mostly only the B-frames were dropped. As shown in Table 3, a sampled unresponsive flow exhibits even stronger resilience, with the RED router only providing 4 good frames out of 162 frames received while the Rb-RIO router provides 34 good frames out of 34 frames received.

Table 2. Frame Statistics for Responsive MPEG flows shown in Figure 9 (0 ~ 30 Seconds)

<table>
<tr><th></th><th colspan="3">Throughput (Kbps)</th><th># frames received</th><th># good frames</th><th>Useless frame %</th></tr>
<tr><td rowspan="3">RED</td><td>0-10</td><td>10-20</td><td>20-30</td><td rowspan="3">265</td><td rowspan="3">185</td><td rowspan="3">30.2</td></tr>
<tr><td>650</td><td>196</td><td>715</td></tr>
<tr><td colspan="3">521</td></tr>
<tr><td rowspan="3">Rb-RIO</td><td>0-10</td><td>10-20</td><td>20-30</td><td rowspan="3">223</td><td rowspan="3">222</td><td rowspan="3">0.4</td></tr>
<tr><td>675</td><td>376</td><td>593</td></tr>
<tr><td colspan="3">822</td></tr>
</table>

Table 3. Frame Statistics for Unresponsive MPEG flows shown in Figure 10 (10 ~ 20 Seconds)

	Throughput (Kbps)	# frame sent	# frame received	# good frames	Actual loss %
RED	647	299	162	4	98.7
Rb-RIO	509	299	34	34	88.6

5. CONCLUSION

In this paper, we have presented Rb-RIO, which is a specific implementation of Rb-RED, that uses priority-classes based AQM, and showed how it could improve the quality of video streams. It is shown that Rb-RIO dramatically improves MPEG video frame loss perceived by the user, while improving fairness over RED. Currently, we are in the process of redesigning the previously proposed Dynamic-CBT [4] to a rate-based one with Rb-RIO for the queue manager for the multimedia classes. We think Rb-RIO, which could be used for a video transmission service class queue mechanism in IETF's Differentiated Services (Diffserv) [1] architecture, could also be used as a standalone mechanism within the current “best effort” Internet service architecture, for example, by randomly (or in Round Robin fashion) classifying packets that are not multimedia packets to one of the 3 priority classes.

6. REFERENCES

[1] Bernet, Y. et. Al. “A Framework for Differentiated Services”, February 1999, Internet Draft, draft-ietf-diffserv-framework-02.txt

[2] Braden, B. et. Al., “Recommendations on Queue Management and Congestion Avoidance in the Internet”, April 1998, RFC-2309

[3] Chung, J. and Claypool, M., “Better-Behaved, Better-Performing Multimedia Networking”, *SCS Euromedia Conference*, Antwerp, Belgium, May 8-10, 2000

[4] Chung, J. and Claypool, M., “Dynamic-CBT – Better Performing Active Queue Management for Multimedia Networking”, In *Proceedings of Network and Operating System Support for Digital audio and Video* (NOSSDAV) 2000, Chapel Hill, NC, June 2000

[5] Cnodder, S., Pauwels, K. and Elloumi, O., “A Rate Based RED Mechanism”, In *Proceedings of Network and Operating System Support for Digital audio and Video* (NOSSDAV) 2000, Chapel Hill, NC, June 2000

[6] D. Clark and W. Fang, “Explicit Allocation of Best-Effort Service”, IEEE/ACM Transactions on Networking, Vol. 6, No. 4, August 1998.

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

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

[9] Mitchell, J., Pennebaker, W., Fogg, C. and LeGall, D., “MPEG Video Compression Standard”, In *Digital Multimedia Standards Series*, Chapman & Hall, New York, NY, 1997

[10] Multimedia Communications Forum, Inc. “Multimedia Communications Quality of Service”, *MMCF/95-010, Approved Rev 1.0*, 1995, URL: http://www.luxcom.com/library/2000/mm_qos/qos.htm

[11] VINT, “Virtual InterNetwork Testbed, A Collaboration among USC/ISI, Xerox PARC, LBNL, and UCB”, URL: http://netweb.usc.edu/vint