Comparison of the Use of Different ECN Techniques for IP Multicast Congestion Control

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
{amatrawy,lambadaris,huang}@sce.carleton.ca

Abstract—**We have recently proposed the use of Backward Explicit Congestion Notification (BECN) with video multicasting over IP networks [1]. This proposal improved bandwidth utilization and reduced time to react to congestion. In this paper, we present a comparison of the use of different ECN techniques, namely ECN, BECN, and MECN, for IP multicast congestion control. We investigate this in the context of MPEG4 multicasting over IP. We use these ECN techniques in an IP network that supports priority dropping of packets during congestion using RED's extension for service differentiation. This extension recognizes the priority of packets when they need to be dropped and drops lower priority packets first. ECN techniques will provide early notification to the sender and/or the receivers about congestion while it is developing in the network. Based on which ECN technique is used, the sender and/or the receivers will react to this notification to reduce the congestion in the network. We show the advantages and disadvantages of each ECN technique. Also, through the results, we advocate the use of network support in the form of explicit congestion notification for IP multicast congestion control.**

I. Introduction

In [1], we have proposed the use of Backward Explicit Congestion Notification (BECN) with video multicasting over IP networks. This proposal improved bandwidth utilization and reduced time to react to congestion. In this work, we extend this proposal and investigate the different options of network support, in the form of congestion notification, to multicast congestion control. We present a comparison between the use of Explicit Congestion Control (ECN), Backward ECN (BECN), and Multi-level ECN (MECN), for IP multicast congestion control. We investigate this in the context of multicasting of adaptively-encoded MPEG4 over IP. We use two MPEG4's properties in our proposal: First, the ability to encode real-time video adaptively to target a certain rate. Second the ability to generate encoded video in two priority levels, one with basic information and the other with enhancement information. We implement these two ECN techniques in an IP network that supports priority dropping of packets during congestion using RED's extension for service differentiation. This extension recognizes the priority of MPEG4 packets when they need to be dropped and drops lower priority packets first. ECN techniques will provide early notification to the sender and/or the receivers about congestion that it is developing in the network. Based on which ECN technique is used, the sender and/or the receivers will react to this notification to reduce the congestion in the network. We discuss the validity of using ECN as an end-to-end mechanism before presenting the use of BECN in our proposal. After that, we use Multi-level ECN (MECN) that sends back quantitative reports that indicates *how congested* the router is. We evaluate how MECN can improve the performance of our proposed architecture.

The rest of this paper is organized as follows. Section II presents an overview of the multicast congestion control problem. Section III presents our work on generating adaptively-encoded MPEG4 traffic for simulation purposes using TES [2] models. In Section IV, we show the RED mechanism that we use in this paper. In Section V, discuss how appropriate is the use of ECN in real-time video applications. Section VI, presents our proposal of using BECN coupled with RED in multicast congestion control for video. This includes the discussion of the end-to-end architecture, the sender rate adaptation algorithm, and evaluation using simulation. Section VII shows the use of MECN in our proposal and compares it to the use of BECN. Section VIII concludes the work done so far, and gives an overview of our future work.

II. Multicast Congestion Control

Multicasting digital video over IP networks faces a number of challenges. The major challenges in this area are:

1. Issues common to all IP multicasting applications:

- The heterogeneity of receivers' networking capabilities as well as the heterogeneity of their QoS-requirements.
- Maintaining the scalability of the multicast congestion control technique is a difficult task as the number of re-

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

ceivers is unknown to the sender and may grow significantly.
2. For video applications, packet loss can be tolerated to some extent. This is the reason those applications can achieve high link utilization as they do not have to respond to packet loss in the same fashion TCP responds. A trade-off between achieving high link utilization and fairness to flows that use TCP-like congestion control is highly desirable.
3. For real-time video, the time to converge to a stable quality is an important issue and so is maintaining this stability.
From the literature, multicast congestion control techniques can be classified into two categories: the sender-based (single-rate) techniques and the receiver-based (multi-rate, layered) techniques.

A. Receiver-based techniques

Receiver-based techniques are based on the ability to generate the source data in a layered format and sending the layers as different multicast groups. Receivers decide on how many layers (or equivalently, multicast groups) they can join using some bandwidth inference technique. Layers should be joined in a *cumulative* manner which means joining them in order of their relevance. Basic layers will contain minimum information necessary to get basic quality and they should be joined first. Different approaches exist for organizing the layers and for bandwidth inference [3], [4], [5], [6], [7]. A *non-cumulative* approach was proposed in [8] in which receivers can get any subset of the layers. This is based on a special encoding technique presented in [9].
Although the receiver-based techniques are a good solution to the heterogeneity problem, they have a number of other problems that are common to most of these techniques:
1. Most of these techniques have fairness problems due to the way they react to congestion and the distribution of data across the layers [10], [11], [12].
2. In a best-effort IP network, which drops packets uniformly at congestion time, packets from the basic layer may be lost which makes receiving higher layers useless.
3. Layered techniques assume that all layers (multicast groups) will follow the same multicast tree even when they are sent separately. This can not be guaranteed in IP networks.

B. Sender-based techniques

In sender-based techniques, a single rate is sent to all receivers. Scalable Feedback Control [13] is one of the earliest works in this area. It uses feedback messages from receivers with information on packet loss to estimate the "group" reception status. Scalability is an obvious problem with this approach because receiving feedback from all receivers simply overwhelms the network. A proposal to use representatives of receivers groups was introduced in [14] and presented mechanisms to select representatives. Changing representatives however is major overhead for this approach. PGMCC [15] is a TCP-friendly protocol which is suitable for applications that can cope with larger variation in the sending rate. However, selection of the *acker* is very crucial to the performance of PGMCC [16]. An extension for equation-based congestion control to multicast applications was recently presented in [16] where a calculation of the wound trip time is needed. Form these proposals, we can identify two major problems with single-rate techniques:
1. Relying on feedback from receivers, a single slow receiver may drag down the data rate for the whole group.
2. Feedback from all receivers is not scalable. Solutions that are based on selecting an agent or a representative of the group presents the overhead of selecting this agent and changing it with changing network conditions.

III. Generating adaptively-encoded MPEG4 using a TES model

Our proposal is based on MPEG4's ability of adaptive encoding [17]. We developed a traffic generator [18] that can be used for studying MPEG4 behavior and performance through simulation using the Transform Expand Sample Methodology [2], [19], [20]. The traffic we generate closely matches the statistical characteristics (in terms of marginal distribution and auto-correlation function) of an original real trace of an MPEG4-encoded video. MPEG4 encoders generate video in three different frame types (I, P, and B) that serve to encode different portions of the video signal in different levels of quality. We modeled the I, P, and B using three TES models and used multiplexing to generate the original sequence of frames for MPEG4. Using feedback messages from the network, we recalculate a new target rate for the MPEG4 encoder and generate video packets based on this rate while maintaining the statistical properties of the original MPEG4 trace. We implemented this generator in software and integrated it into the network simulator *ns-2* [21].

IV. RED multiple-buffer management

We define a QoS-aware network model that supports our architecture for multicast MPEG4. We require the network to:
1. Support priority dropping of packets in time of congestion
2. Provide congestion notification from routers to the end systems (sender and/or receiver).
In this section, we discuss our choice for fulfilling the first requirement. We use RED's extension for service differentiation [22]. Packets are marked with different priorities and are being treated according to this priority when they need to be dropped during network congestion.

Random Early Detection (RED) [23] is a buffer management technique that is used for congestion avoidance in IP networks. RED routers try to early detect upcoming congestion by computing an average of the queue size in the router. A sustained long queue is a sign of network congestion. When a packet arrives, a RED router checks the av-

erage queue size against specified *min* and *max* thresholds. Based on this check, one of three actions is taken:
1. IF Queue-Average is less than *min*
THEN no action is taken
2. IF Queue-Average is greater than *min* but less than *max*
THEN with probability, the packet is dropped
3. IF Queue-Average is greater than *max*
THEN packet is dropped

To achieve differentiation between different priority traffic classes, different sets of RED parameter values would need to be maintained for each class. Thus if there are two priority classes, two sets of parameters need to be maintained. Each set would affect arriving packets in its priority class based on its own RED parameters. RED will be managing a separate virtual queue for each traffic class. Certain calculations are performed to get the probability for dropping a packet. These calculations can be based on each virtual queue or by coupling them together [22].

V. Issues With The Use of Explicit Congestion Notification (ECN)

As mentioned in Section IV, in RED buffer management, if the queue size is between its *min* and *max* thresholds, the packet is dropped with a probability. In the case of unicast, dropping this packet will signal the end-to-end congestion avoidance mechanism of TCP that the network is congested.

Explicit Congestion Notification (ECN) [24] was proposed for congestion detection and avoidance with TCP. In ECN, the packet is marked and sent to the receiver if the queue size is between its *min* and *max* thresholds,. The receiver in this case, marks a flag in the TCP header of an ACK message and sends it back to the sender. Based on the information in this ACK the sender reacts by reducing its congestion window as well as its slow start threshold. The sender then sends some notification to the receivers that it did that to stop the receiver from sending more ACKs back. This mechanism forces the TCP sender to react early before congestion develops without the need to drop packets.

However, considering this methodology for real-time video multicasting has some limitations:
1. This approach is not appropriate for use in real-time cases because it is an end-to-end solution that requires *every* receiver to send a message back. This will lead to the *feedback implosion* problem we mentioned in Section II-B.
2. It takes a round trip time (RTT) before the sender reacts. This is not suitable for delay-sensitive real-time video applications.

VI. Backward Explicit Congestion Notification (BECN)

The authors in [25] proposed an extension to ECN using feedback from the router to the sender as an indication of congestion. This feedback message is sent if the queue size is between its *min* and *max* thresholds or if it is greater than *max* threshold. The packet is still marked to prevent other routers from sending more feedback messages for the same packets. The sender responds to that in the same manner it does in the case of ECN. They called it Backward ECN (BECN). This is done using the existing IP signaling mechanism the Internet Control Message Protocol (ICMP). Sending an ICMP message to the sender from the router has an advantage over ECN which is the lower time it takes before the sender can react. Also, because it is an IP level mechanism it can work with transport protocols other than TCP.

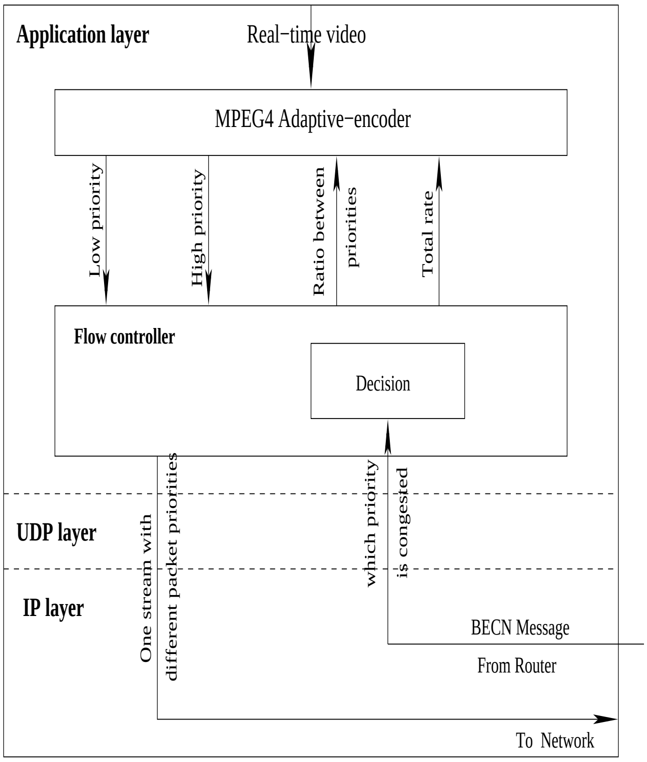

Fig. 1. Protocol stack at the sender

In our architecture, we use BECN with the flow controller of the video application that works on top of UDP. We implemented BECN in our simulations to work on parameters of the virtual queues. We send BECN[1] messages back to the video sender based on the status of every virtual queue, thus sending back information on which priority level is experiencing problems. We implemented this and integrated it in *ns-2* [21]. The authors in [25] also proposed a variation of BECN that sends some quantitative indication of the level of the congestion that is developing in the router. They called that Multilevel BECN or MECN. A performance study for BECN can be found in [26].

A. End-to-end architecture

In this section, we overview our proposed end-to-end architecture for video multicasting:

[1] In the case of multicast, the Internet Group Management Protocol (IGMP) is used instead of ICMP

```
REPEAT every t Seconds
    IF nofeedback messages received
    THEN increase Total−Rate;
            reduce BasicLayerPortion;
            reduce t
    ELSEIF feddback increasing at ALL priority levels
    THEN reduce TotalRate;
            increase t
    ELSEIF feedback increasing at basic layer
    THEN reduce BasicLayerPortion;
             increase t
END REPEAT
```

Fig. 2. Flow control algorithm at the sender

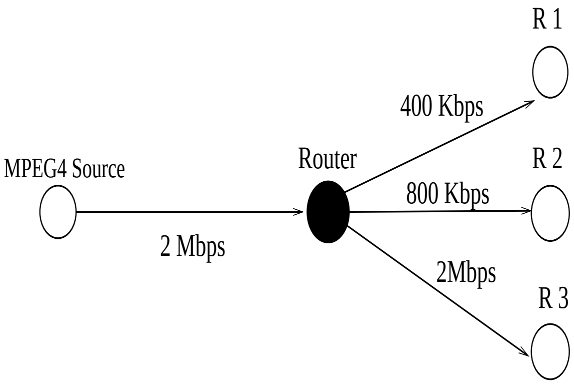

Fig. 3. Simulation setup

1. Up until this point of our work, we did not define an active role for the receivers in our architecture.
2. The sender marks the MPEG4 packets with two different priorities with basic information marked with high priority and enhancement information is marked with lower priority. Thus, during congestion basic video quality can still be received. This is basically sending the video information in layers within one stream. This removes the burden of dealing with different layers (multicast groups) at the receiver and ensuring that all packets will follow the same multicast tree (refer to Section II-A).
3. While congestion is developing, routers that run RED with multiple virtual queues will send back BECN SQ messages to the sender with information on the priority level that caused the problem.

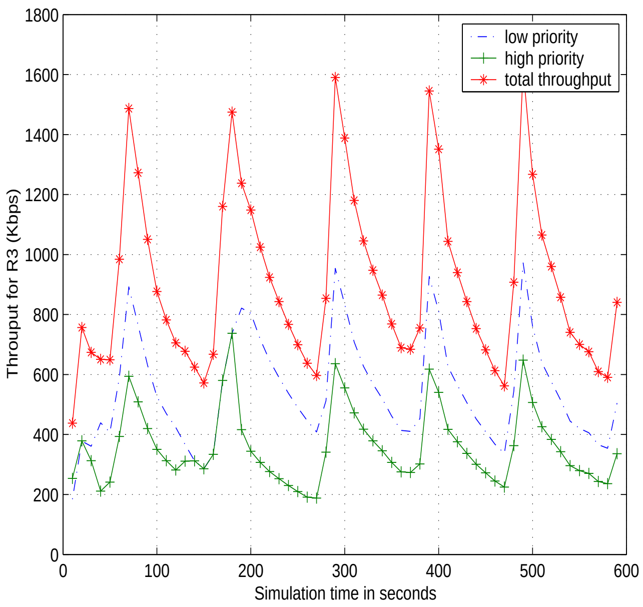

Fig. 4. Throughput for Receiver R3 using BECN

4. Based on the rate of these feedback messages, the sender runs an algorithm to search for an operating point (total sending rate and ratio between priority levels) that will reduce this feedback messages rate. The protocol stack[2] at the sender is shown in Fig. 1. Figure 2 shows a pseudo-code for the flow control algorithm (functionality of the *decide* box in Fig 1). Note that if the problem is only at the enhancement layer it is not fixed until both layers are congested simply because this enhancement layer will get dropped where it is needed.
5. The sender will try to match the sending rate of the high priority video with the sending rate of slow receivers. This will allow them to get useful information in the time of persistent congestion. This is also how we deal with heterogeneity. Note that a receiver may get a 400Kbps out of the 1Mbps original signal and still can reconstruct a comprehensible video because the information was encoded with 400Kbps basic information layer. Without priorities, a slow receiver may still get the same rate from the same source but because the packets carry information with same level of importance nothing comprehensible can be reconstructed.

B. *Simulation and performance*

We used simulations to perform initial testing for our proposal. We used *ns-2* for running the simulations.

B.1 Simulation setup

In Fig. 3, we show the basic topology we used for simulations. There is one MPEG4 source that sends MPEG4 traffic using the generator we introduced in Section III. The trace we used for the generating the model is from a news broadcast. Properties of the trace can be found in [18]. Three receivers R1, R2 and R3 are connected to the sender through

[2] We show only the functionalities that we proposed or modified

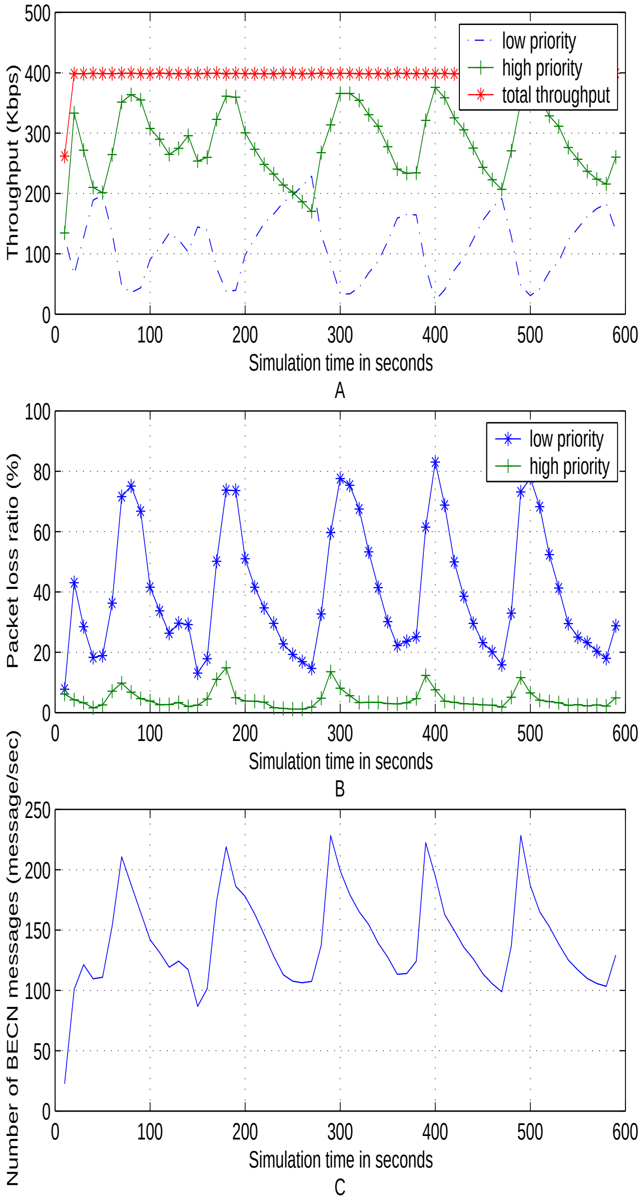

Fig. 5. Performance of Receiver R1 using BECN

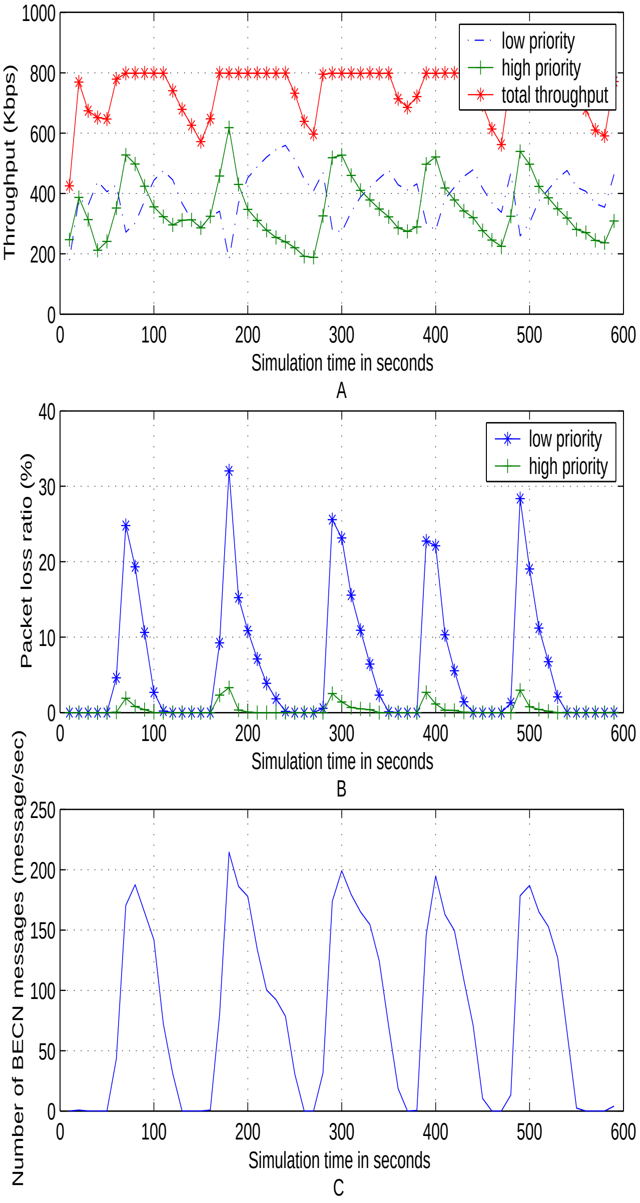

Fig. 6. Performance of Receiver R2 using BECN

a router as shown. The link from the sender to the router is 2Mbps and so is the link from the router to R3. We made this as high as the link from the sender to the router so that R3 will not have any problems receiving the full video multicast with no packet loss. R1 has a 400Kbps link to router and R2 has a 800Kbps one.

B.2 Results

The goal of our simulations is to see if and how the sender flow controller will converge to a point where both receivers will maximize the utilization of their links and keep the information received by each of them reconstructible. From Fig. 4, we can see the performance of R3 which did not have any loss. Whenever the sender rate is too high for the other receivers we see a slow down in the total rate in reaction to the BECN messages that R1 and R2 are sending back. Still the average received rate is around 1Mbps, which results in under-utilizing the link for R3. One solution to this, in the presence of such a high heterogeneity between receivers capacities, is a finer granularity. That is using more priority levels and the modification of the flow controller at the sender to adjust the sending rates at these priority levels to match as many receivers as possible.

In order to see how this works, we check Fig. 5 and Fig. 6. Both receivers R1, and R2 maximized their link utilization and more importantly they secured the basic layer during congestion. For R1, in part A of the figure, we can see that most of the received packets are form the basic layer while R2 is getting the same rate in basic layer and a higher rate in the enhancement layer. The basic layer is not increased more than a level that is not appropriate for the slowest receiver, R1. In part B of both figure we can see that the loss rate is much higher in the enhancement layer, specially for R1. We should also note the relation between the reduction of total rate and the increase of BECN messages. The times of higher loss rates are relatively short because BECN reduces the time the sender takes to react to upcoming congestion. There is a need to reduce the number of BECN messages back to the sender. The same packet may cause both receivers to send a feedback message to the sender. A solution to this is to make the packet carry the number of times it was duplicated and how far it is from the source. The further it is from the source and the more it was duplicated, the less likely it should be marked or an BECN message sent for it.

VII. Multi-level Explicit Congestion Notification (MECN)

In this section, we present our use of the Multi-level ECN (MECN) [25] in our proposed architecture.

A. Difference from BECN

The difference is that instead of just sending back a BECN message to the sender, the message contains quantitative information about the congestion status of the router queue that sent the message. One way of doing that is sending different values that indicates how big is the queue size with reference to the RED parameters min and max. In our study, we send back of three values to the sender as follows:

$$\begin{aligned}
v_1 : \quad & if \quad \{min \leq \text{QueueSize}(i) < (min + \frac{max - min}{2})\} \\
v_2 : \quad & if \quad \{(min + \frac{max - min}{2}) \leq \text{QueueSize}(i) < max\} \\
v_3 : \quad & if \quad \{\text{QueueSize}(i) \geq max\}
\end{aligned}$$

The sender reacts to these values with different values for rate reduction following the same algorithm in Fig. 2. Receiving the value v_3 at the sender means that packet are being *dropped* and hence the rate reduction should be considerably big. Values v_2 and v_1 represent lower levels of congestion and should result in lower rate reduction.

B. Results

We repeated the same simulation of Section VI-B using the simulation setup of Fig. 3. The same end-to-end architecture of Section VI-A is used in the study.

The results show that using MECN gives a better convergence for the rates at the two difference priorities than using BECN. We can see that by comparing figures 7, 8, and 9 to their counterparts of Section VI. In Fig. 7, the throughput of R3 is almost the same as that of Fig. 4 with a tendency to increase towards the end of simulation time. The difference is in the distribution of the rate among the two priority levels and the changes in the rate assigned to each layer. High priority layer is adjusted to 400Kbps to match the rate of R1. This is much better than the case of BECN where the rate was kept in that range with a much hight fluctuation that would result in a change in the received quality that would not likely by desirable by the user.

To check how this affects the two receivers individually, we first check Fig. 8. The rate for the high priority layer received by R1 converges to almost 400Kbps with little variation. This comes at the expense of high loss ratio at the low priority layer and an increasing loss at the high priority layer. Utilization in this case is maximized. Checking Fig. 9 shows that a bigger portion of the traffic is from the higher layer specially towards the end of the simulation although it is not as good as it is in case of R1.

So the difference from BECN is a better matching of the higher priority rate (specially for the slowest receiver) and a smoother rate at that layer. The drawback is that it is still not accurate since the amount of increase or decrease of the rate is still determined using experimentation. We are working on formalizing that rate adaptation to get an accurate match of the available bandwidth and reduce the loss ratio.

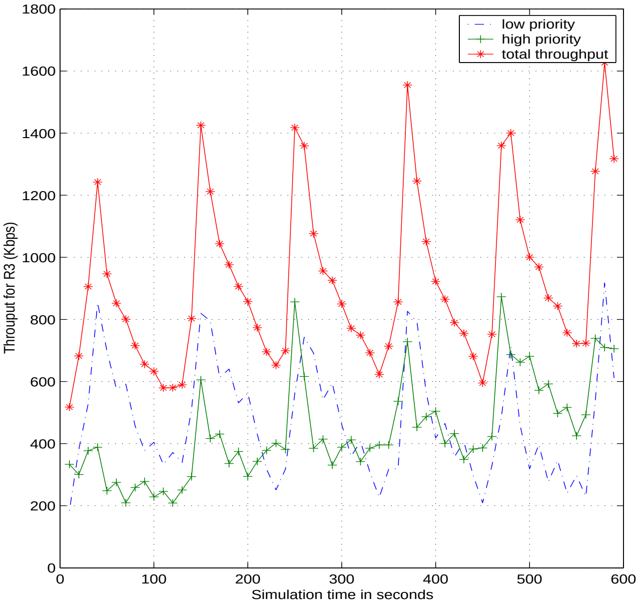

Fig. 7. Performance of Receiver R3 using MECN

VIII. Conclusion

In this work, we build on our earlier proposal of using BECN with IP Multicasting. we compare the improvement that can be achieved in the performance of IP video mul-

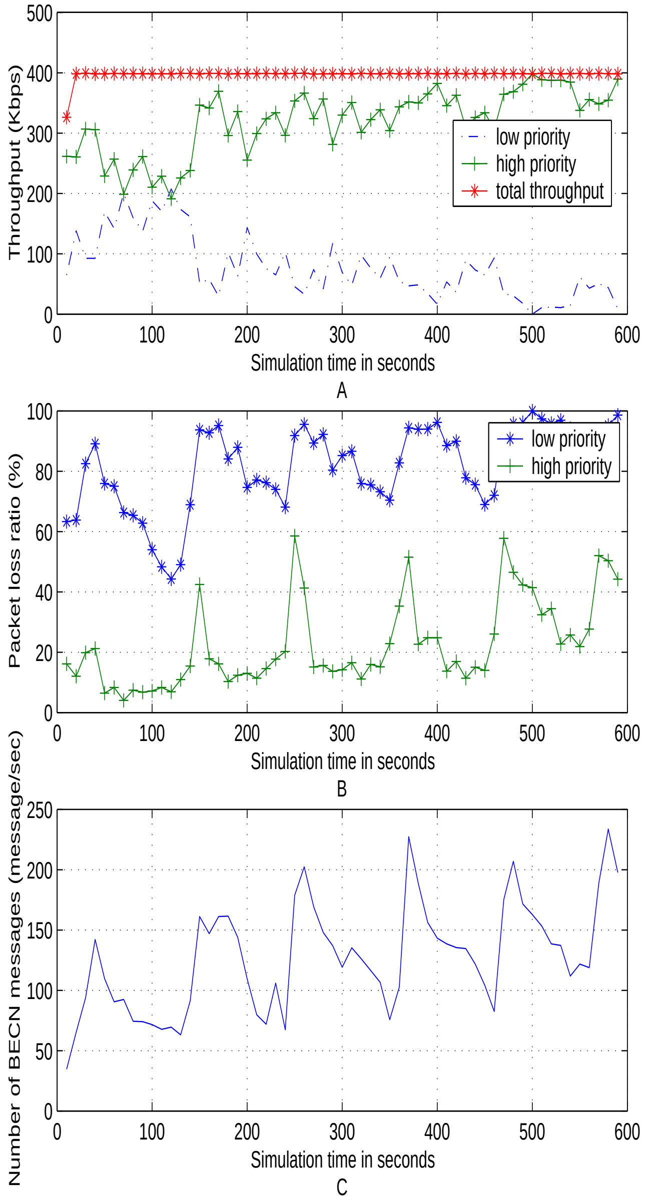

Fig. 8. Performance of Receiver R1 using MECN

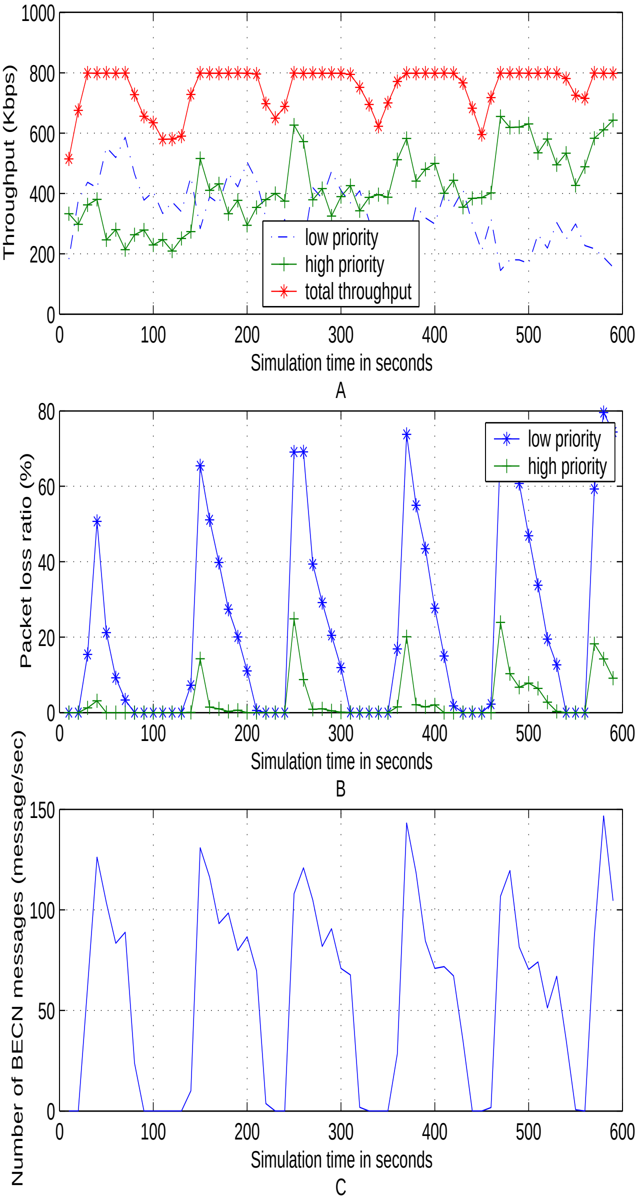

Fig. 9. Performance of Receiver R2 using MECN

ticasting using different techniques of Explicit Congestion Notification.

Using ECN in the same fashion it is used with TCP is not suitable for multicasting. Our proposal of using BECN or MECN has the following advantages:

1. Sending the video as one stream is much easier to handle than multiple streams

2. Sending all layers within one stream with different priority labels over a network that supports priority dropping ensures that a minimum quality will be received in the time of network congestion

3. Scalability issues is minimized when the feedback to the sender is provided from the network rather than from receivers

4. Future network architectures support QoS in one way or the other so it is required to study how video multicasting will work in this environment

The results we got in this paper shows that this approach is encouraging. It also shows that using MECN gives the same overall performance as BECN but with the advantage of having a better convergence towards the target rate for the high priority video with a much smoother rate change. Our future work will concentrate on formalizing the rate adaptation process to make it more accurate and to reduce the loss ratio. Testing the performance with more priorities and more receivers is needed too.

REFERENCES

[1] A. Matrawy, I. Lambadaris, and C. Huang, "Multicasting of Adaptively-encoded MPEG4 on a QoS-aware IP Networks," *In Proc. of IEEE ICC*, 2002.

[2] B. Melamed, "An Overview of TES Processes and Modeling Methodology," *In Performance Evaluation of Computer and Communication Systems*, 1993.

[3] S. McCanne, "Scalable Compression and Transmission of Internet Multicast Video," *Ph.D. Thesis, University of California-Berkeley*, December 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] A. Legout and E. Biersack, "PLM: Fast Congergence for Cumulative Layered Multicast Transmission Schemes," *In Proc. of SIGMETRICS*, pp. 13–22, 2000.

[6] L. Vicisano, J. Crowcroft, and L. Rizzo, "TCP-like Congestion Control for Layered Multicast Data Transfer," *in Proc. of INFOCOM*, 1998.

[7] L. Wu, R. Sharma, and B. Smith, "Thin Streams: An Architecture for Multicasting Layered Video," *In Proc. of NOSSDAV*, 1997.

[8] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-Grained Layered Multicast," *In Proc. of INFOCOM*, 2001.

[9] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *In Proc. of SIGCOMM*, 1998.

[10] A. Matrawy, I. Lambadaris, and C. Huang, "On Layered Video Fairness on IP Networks," *In Proc. of IEEE GLOBECOM*, 2001.

[11] A. Legout and E. Biersack, "Pathological Behaviors for RLM and RLC," *In Proc. of NOSSDAV*, 2000.

[12] R. Gopalakrishnan et al., "Stability and Fairness Issues in Layered Multicast," *In Proc. of NOSSDAV*, 1999.

[13] J-C. Bolot, T. Turletti, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," *in Proc. of ACM SIGCOMM*, October 1994.

[14] D. DeLucia and K. Obraczka, "Multicast Feedback Suppression using Representatives," *in Proc. of IEEE INFOCOM*, 1997.

[15] L. Rizzo, "pgmcc: a TCP-friendly single-rate Multicast Congestion Control Scheme," *in Proc. of SIGCOMM*, 2000.

[16] J. Widmer and M. Handley, "Extending Equation-based Congestion Control to Multicast Applications," *In Proc. of SIGCOMM*, 2001.

[17] R. Koenen, "MPEG4 Overview," *In IEEE Spectrum*, February 1999.

[18] A. Matrawy, I. Lambadaris, and C. Huang, "MPEG4 Traffic Modeling using The Transform Expand Sample Methodology," *In Proc. of 4th IEEE International Workshop on Networked Appliances*, January 2002.

[19] D. Reininger et al., "Variable Bit Rate MPEG video: Characterstics, Modeling and Multiplexing," *In Proc. of ITC*, 1994.

[20] M. R. Ismail, I. E. lambadaris, M. Devetsikiotis, and A. R. kaye, "Modeling Prioritized MPEG Video using TES and a Frame Spreading Strategy for Transmission in ATM Networks," *In Proc. of INFOCOM*, 1995.

[21] "NS. UCB/LBNL/VINT Network Simulator (version 2)," *http://www.isi.edu/nsnam/*.

[22] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Trans. on Networking*, vol. 6, no. 4, pp. 362–373, August 1998.

[23] S. Floyed and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. on Networking*, pp. 397–413, August 1993.

[24] S. Floyed, "TCP and Explicit Congestion Notification," *ACM Computer Communications Review*, pp. 10–23, October 1994.

[25] J. Hadi Salim, B. Nandy, and N. Seddigh, "A Proposal for Backward ECN for the Internet Protocol (IPv4/IPv6)," *Internet Draft, draft-salim-jhsbnns-ecn-00.txt*.

[26] R. Makkar, "QoS Control for IP Netowrks: Buffer Management and Backward Explicit Congestion Notification (BECN)," *M.Eng. thesis, Carleton Univeristy, Ottawa, Canada*, December 2000.