Multicasting of Adaptively-encoded MPEG4 over QoS-aware IP Networks

Ashraf Matrawy Ioannis Lambadaris Changcheng Huang
Broadband Networks Laboratory
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada

Abstract— **We propose a novel architecture for multicasting of adaptively-encoded layered MPEG4 over a QoS-aware IP network. We require a QoS-aware IP network in this case to (1) Support priority dropping of packets in time of congestion. (2) Provide congestion notification to the multicast sender. For the first requirement, we use RED's extension for service differentiation. It recognizes the priority of packets when they need to be dropped and drops lower priority packets first. We couple RED with our proposal for the second requirement which is the adoption of Backward Explicit Congestion Notification (BECN) for use with IP multicast. BECN will provide early congestion notification at the IP layer level to the video sender. BECN detects upcoming congestion based on size of the RED queue in the routers. The MPEG4 adaptive-encoder can change the sending rate and also can divide the video packets into lower priority packets and high priority packets. Based on BECN messages from the routers, a simple flow controller at the sender sets the rate for the adaptive MPEG4 encoder and also sets the ratio between the high priority and low priority packets within the video stream. We use a TES model for generating the MPEG4 traffic that is based on real video traces. Simulation results show that combining priority dropping, MPEG4 adaptive encoding, and multicast BECN: (1) Improves bandwidth utilization (2) Reduces time to react to congestion and hence improves the received video quality (3) Maintains graceful degradation in quality with congestion and provides minimum quality even if congestion persists.**

I. INTRODUCTION

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 receivers 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, video multicast congestion control techniques can be classified into two categories: the sender-based (single-rate) techniques and the receiver-based (multi-rate, layered) techniques.

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

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 [1], [2], [3], [4], [5]. A *non-cumulative* approach was proposed in [6] in which receivers can get any subset of the layers. This is based on a special encoding technique presented in [7]. 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 [8], [9], [10].
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 [11] 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 [12] and presented mechanisms to select representatives. Changing representatives however is major overhead for this approach. PGMCC [13] 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 [14]. An extension for equation-based congestion control to multicast applications was recently presented in [14] where a calculation of the wound trip time is needed. Form these proposals, we can identify two major problems with single-rate techniques:

0-7803-7400-2/02/$17.00 © 2002 IEEE

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.

In this paper, we propose a novel approach for multicast congestion control for single-rate single-source IP multicast sessions. The approach is based on router support in the form of early Backward Explicit Congestion Notification (BECN) messages to the sender. This feedback mechanism is more scalable than earlier proposals for feedback control in multicast environments since it is only provided by the routers not by all the receivers of a multicast session. There is no overhead in selecting representatives in this case [12]. This approach also has the advantage of compatibility with a variety of transport protocols since BECN [15] was based on an IP-level signaling protocol (ICMP).[1] We couple Multicast BECN with RED's extension for service differentiation [16] to send MPEG4 packets with different priorities in the network and provide BECN at each priority level. 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. Using the feedback messages rate as an indication of congestion at a certain priority level, a flow controller calculates a new target rate for the MPEG4 encoder and also the ratio between the two priorities.

The rest of this paper is organized as follows. Section II presents our work on generating adaptively-encoded MPEG4 traffic for simulation purposes using TES [17] models. In Section III, we present our network model. First, in Section III-A, we show the RED mechanism that we use in this paper and then, in Section III-B, we discuss our proposal of using BECN coupled with RED in multicast congestion control for video. In Section IV, we present the end-to-end architecture for MPEG4 multicasting on our network model. Evaluation of this architecture using simulation is given in Section V. Section VI concludes the paper and gives an overview of our future work.

II. Generating adaptively-encoded MPEG4 using a TES model

Our proposal is based on MPEG4's ability of adaptive encoding [18]. We developed a traffic generator [19] that can be used for studying MPEG4 behavior and performance through simulation using the Transform Expand Sample Methodology [17], [20], [21]. The traffic we generate closely matches the statistical characteristics (in terms of marginal distribution and autocorrelation 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* [22].

[1] In the multicast case, the equivalent of ICMP should be used, i.e., the Internet Group Management Protocol (IGMP).

III. Network model

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 multicast sender.

In the following subsections we discuss our choices for fulfilling these two requirements. For the first requirement, we use RED's extension for service differentiation [16]. Packets are marked with different priorities and are being treated according to this priority when they need to be dropped during network congestion. For the second requirement, we extend the proposal in [15] for use in multicast applications.

A. RED multiple-buffer management

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 average 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 [16].

B. Backward Explicit Congestion Notification with Multiple-buffer RED

As we mentioned above, 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 TCP, if Explicit Congestion Notification (ECN) [24] is used, the packet is marked and sent to the receiver. 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, this method has some limitations:

1. This approach is coupled with TCP
2. It takes a round trip time (RTT) before the sender reacts

In [15], the authors proposed using feedback at the IP layer that should result in the same sender reaction. 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. They called it Backward ECN (BECN). This is done using the existing IP signaling mechanism, ICMP. Sending an ICMP Source Quench (ICMP SQ) 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. 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 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* [22]. The authors in [15] 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 [25].

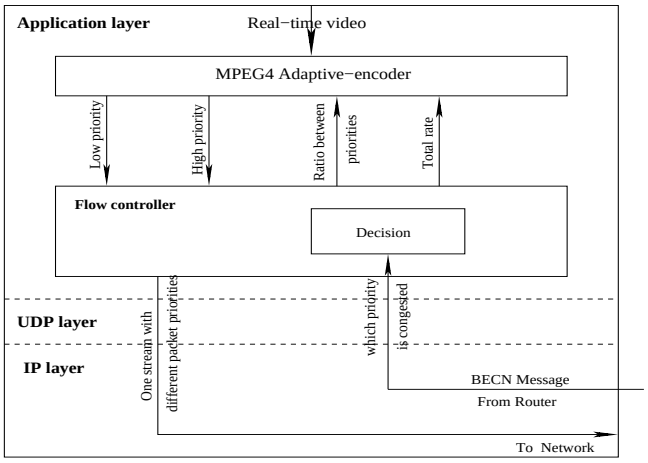

Fig. 1. Protocol stack at the sender

IV. End-to-end architecture

In this section, we overview our proposed architecture for video multicasting over the QoS-aware model we defined and implemented in Section III.

1. Up until this point of our work, we did not define an active role for the receivers in our architecture.

```
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

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 I).
3. While congestion is developing, routers that run RED with multiple virtual queues will send back BECN messages to the sender with information on the priority level that caused the problem.
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 at the sender is shown in Fig. 1 where we only show the parts that we added or modified. 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 all 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 for the high priority 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.

V. Simulation and performance

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

A. 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 II. The trace we used for the

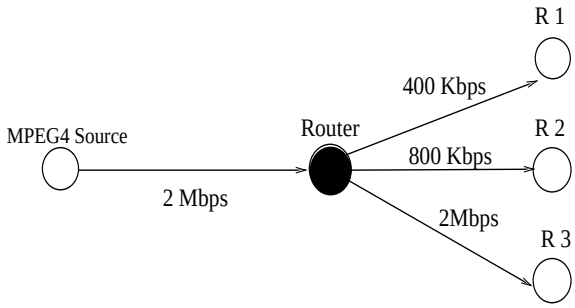

Fig. 3. Simulation setup

generating the model is from a news broadcast. Properties of the trace can be found in [19]. Three receivers R1, R2 and R3 are connected to the sender through 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. Preliminary 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.

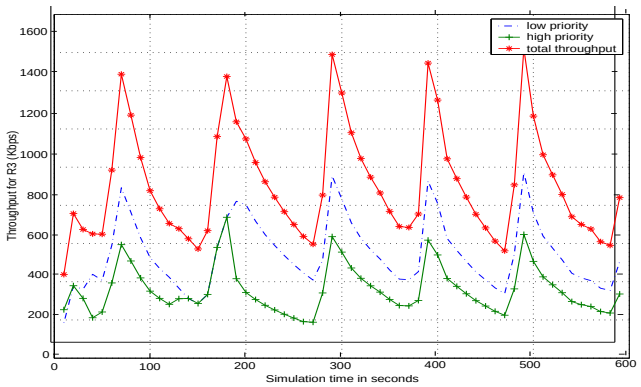

Fig. 4. Throughput for Receiver R3

VI. Conclusion and Future Work

In this work, we investigated the improvement that can be achieved in the performance of video multicasting if the network model is supporting QoS. We proposed a simple control framework for the sender. Our proposal along with the network model that we defined avoids a lot of the problems of earlier work in this area. The advantages of this work can be summarized in the following points:

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 show that this approach is encouraging. However, our results are still preliminary more extensive analysis is desired to get more insight into the performance of this approach. Testing the performance with more priorities and more receivers is needed too. We also intend to use Multilevel BECN (MECN) that provides quantitative information on the congestion status to help the sender make more accurate decisions about the rates and priorities.

Acknowledgments

The authors appreciate the fruitful discussion they had with Nabil Seddigh and Biswajit Nandy of Tropic Networks and with Rupinder Makkar of the Broadband Networks Laboratory at Carleton University.

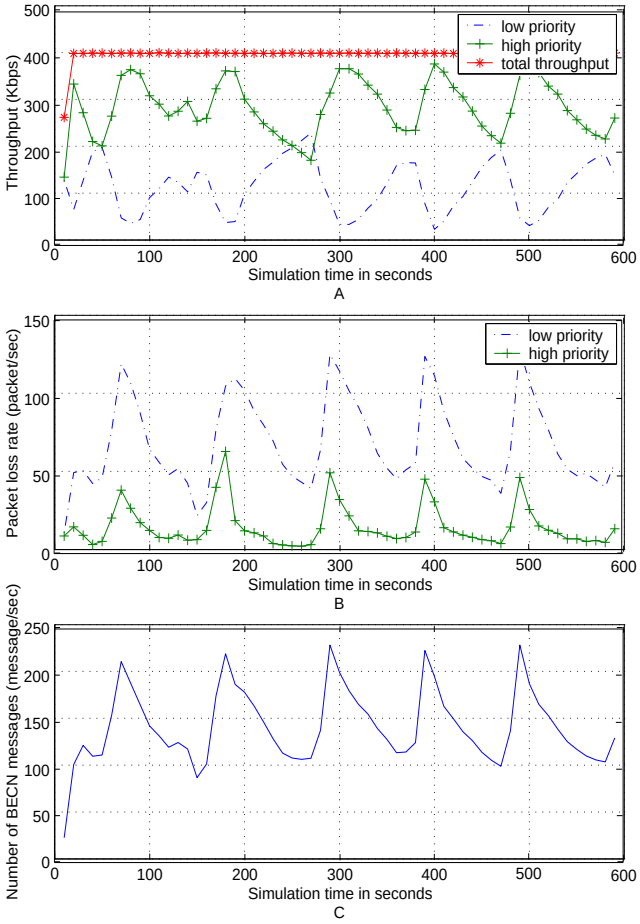

Fig. 5. Performance of Receiver R1

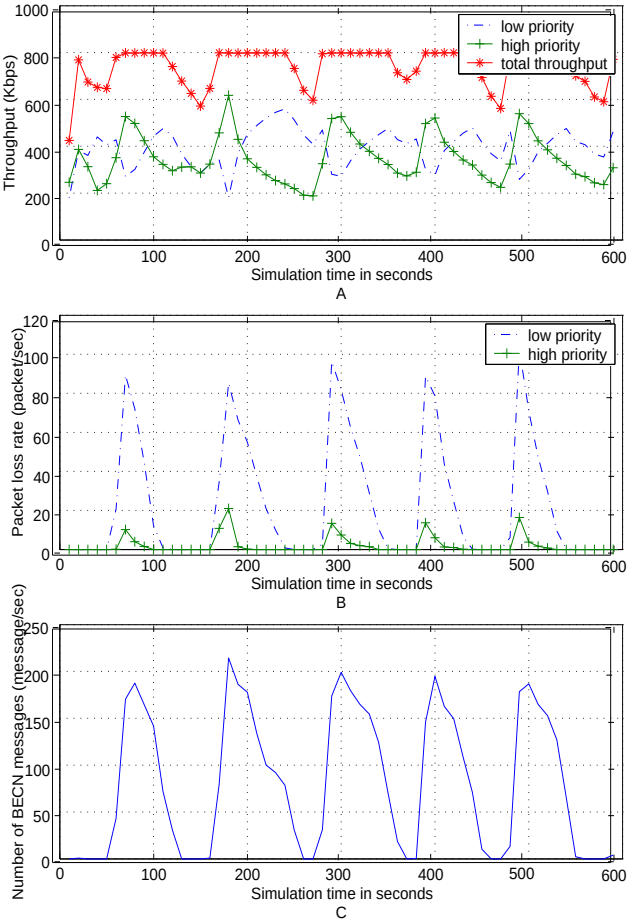

Fig. 6. Performance of Receiver R2

REFERENCES

[1] S. McCanne, "Scalable Compression and Transmission of Internet Multicast Video," *Ph.D. Thesis, University of California-Berkeley*, December 1996.

[2] 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.

[3] A. Legout and E. Biersack, "PLM: Fast Congergence for Cumulative Layered Multicast Transmission Schemes," *In Proc. of SIGMETRICS*, 2000.

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

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

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

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

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

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

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

[11] 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.

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

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

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

[15] 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.*

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

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

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

[19] 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.

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

[21] 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.

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

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

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

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