

# On End-to-End QOS Mapping

*Jean-François Huard and Aurel A. Lazar*  
*Department of Electrical Engineering and*  
*Center for Telecommunications Research*  
*Columbia University, New York, NY, 10027-6699*  
*<http://comet.ctr.columbia.edu/~{jfhuard, aurel}>*

## Abstract

A framework for studying the end-to-end QOS mapping between the various levels of the transport protocol stack is presented. A platform for evaluating end-to-end QOS that supports concurrent network, transport and application level measurements is described. QOS measurements for various video clips indicate that the loss bound obtained under the assumption of uniformly distributed cell losses within a video frame is too conservative.

## Keywords

Quality of Service, QOS Mapping, End-to-End QOS, Multimedia Networks

## 1 INTRODUCTION

The task of guaranteeing end-to-end QOS requires the understanding of the various levels of specification of QOS of the protocol stack. For example, switch vendors are concerned only with per-hop, per-class cell-level QOS while multimedia applications are concerned with end-to-end, per connection frame-level QOS. It is the end-to-end QOS, rather than the per-hop QOS, that is perceived and important to the users. Furthermore, network operators use the latter for guaranteeing QOS to their users and need to translate user QOS into network QOS. Each of these QOS specifications deal with different statistical quantities that have to be mapped into each other.

The focus of this paper is on the mapping of loss characteristics between the application level and the network level. In particular, we consider frames (video frames and audio packets) at the application level, and ATM cells at the network level. That is, we are investigating the relationship between frame loss and cell loss. Application-to-network QOS mapping is needed to reserve the appropriate amount of network resources at connection establishment time. Furthermore, good mapping rules are essential in order to avoid reserving too much (or too little) resources. Finally, we are concerned with end-to-end QOS mapping. We assume that it is the responsibility of the routing to determine the per-hop QOS given the end-to-end network QOS.

This paper is organized as follows: Section 2 presents the QOS mapping rule widely used in the literature. To validate this mapping rule, we present in Section 3 a measurement platform for evaluating the end-to-end QOS. The platform supports concurrent network, transport and application level measurements of QOS. In Section 4 measurement results are presented. Finally, we review some related work in Section 5.

## 2 END-TO-END QOS MAPPING

The process of translating QOS specifications between different levels of the protocol stack is called QOS mapping. User-to-application mapping is needed to ease the process of selecting QOS at the human-machine interface. It is a mapping from a set of user preferences to a quantitative description of the service desired. Application-to-network mapping is needed to reserve the appropriate network resources at connection establishment time. It is assumed that it is the responsibility of the QOS routing system to find a route providing the required end-to-end cell level QOS. Thus, it is the responsibility of the routing to aggregate the per-hop QOS on a route and ensure that it satisfies the end-to-end QOS requirements. Finally, application-to-transport mapping is needed for monitoring and adapting to the rapid network fluctuations of QOS. In this paper, we address only on the application-to-network parameter mapping and focus mainly on loss.

Let us define what we mean by end-to-end QOS specification: for any level in the protocol stack, the QOS is specified (and measured) from the moment a level  $L$  protocol data unit (PDU) crosses the boundary from level  $L$  to  $L-1$  at the source endpoint to the moment it crosses the boundary from level  $L-1$  to  $L$  at its destination endpoint. In particular, the end-to-end network QOS will be given in terms of cell level statistics between the network adapter of the source endpoint to the network adapter of the destination endpoint and the application level QOS will be given in terms of frames from the moment a frame is grabbed and sent to the moment it is received and played back.

Three descriptors are used for traffic characterization: the maximum protocol data unit size (e.g., maximum video frame size, audio packet size and ATM cell size), average PDU size and the maximum PDU rate (e.g., video frame rate, audio packet rate, peak cell rate); when combined, the descriptors give the peak rate, the average rate and the burstiness of the media stream. The parameters used in the QOS profile are the maximum PDU delay, the maximum PDU loss rate and the average PDU gap loss\*. At the destination, these parameters need to be measured over a time interval called the *measurement era*. The era can be finite or infinite; i.e., a fixed measurement window or the duration of a connection since its beginning,  $t_0$ . In the latter case, at any time  $t$ , the era is dynamically set to  $t - t_0$ . The era is also part of the

---

\*The PDU gap loss is the number of consecutively lost PDUs.

QoS profile. When measurements are obtained over a (short) fixed era, we will sometimes refer to them as instantaneous measurements.

The QoS mapping is generally done in two steps: a mapping between the services and traffic descriptors and a rescaling of the parameters of the QoS profile. Here, we look only at the parameter mapping. The application sends complete frames and receives complete, error free, frames. The transport uses AAL5 framing, and thus, a frame is lost if any of the following occurs: one or more cells are missing (due to buffer overflow, clipping or pure loss), a cell misinsertion or AAL5 CRC failure due to bit error.

Let  $a, n$  denote the application and network, respectively. To write parameter mapping rules, the following symbols are needed:

$R(l)$ :	PDU max rate	(# pdu/sec),
$S(l)$ :	PDU max size	(bytes/pdu),
$A(l)$ :	PDU ave. size	(bytes/pdu),
$L(l)$ :	PDU loss rate	(# pdu/sec),
$D(l)$ :	PDU (end-to-end) delay	(sec),
$G(l)$ :	PDU ave. gap loss	(# pdu),
$eta(l)$ :	measurement era	(sec),

where  $l$  is either  $a$  or  $n$ . Finally, the peak rate at level  $l$  by  $PR(l) = S(l) \cdot R(l)$  (bytes/sec) and the average rate  $AR(l) = A(l) \cdot R(l)$  (bytes/sec).

Intuitively, if the cell loss is small (say less than  $10^{-4}$ ), the frames not too small (say larger than 2k bytes (or 40 cells)) and the losses are uniformly distributed within a frame, then the following relation should hold

$$L(n) = L(a) \cdot A(n)/A(a). \quad (1)$$

For example, if a frame loss of  $10^{-3}$  is desired, with average frame size of 2400 bytes then the cell loss should be about  $10^{-3} \cdot 48/2400$  or  $2 \cdot 10^{-5}$ .

In order to empirically evaluate the loss mapping rule prescribed by Equation 1, we have been performing experimental loss measurements concurrently at the application, transport and network level.

### 3 A QOS MONITORING PLATFORM

We have implemented a QoS measurement system for each level in the protocol stack. To allow for application level measurements, frames are timestamped when grabbed and just before being played back. For the transport level measurements, TPDU are timestamped when sent by the application and when available for delivery to the application. Finally, for network QoS measurements, a probing system has been implemented on the firmware of an HP Broadband network analyzer. The system permits cell delay measure-

**Table 1** Motion JPEG video clips statistics.

	medium	large
frame rate ( $\text{sec}^{-1}$ )	25	20
number of frames	15 000	15 000
ave. frame size (bytes)	6394	14970
min. frame size (bytes)	2052	7131
max. frame size (bytes)	13 364	45 234
std deviation of frame size	2132	4936

ments from one network adapter to another with a precision in the order of microseconds.

The cell level measurement system locks on a stream and measures the network QOS by injecting probe cells. Two probing methods were implemented. Both methods detect the beginning of a frame and then inject probe cells until the end of the frame is detected. The first method injects one probe cell every eight data cells. The second method injects a burst of twelve cells every sixty-four data cells. As the probes are interleaved with the video data streams, the QOS experienced by the probe cells should be very similar to the actual QOS received by the video stream cells. The second method allows for better cell gap loss measurements while the first for average cell loss.

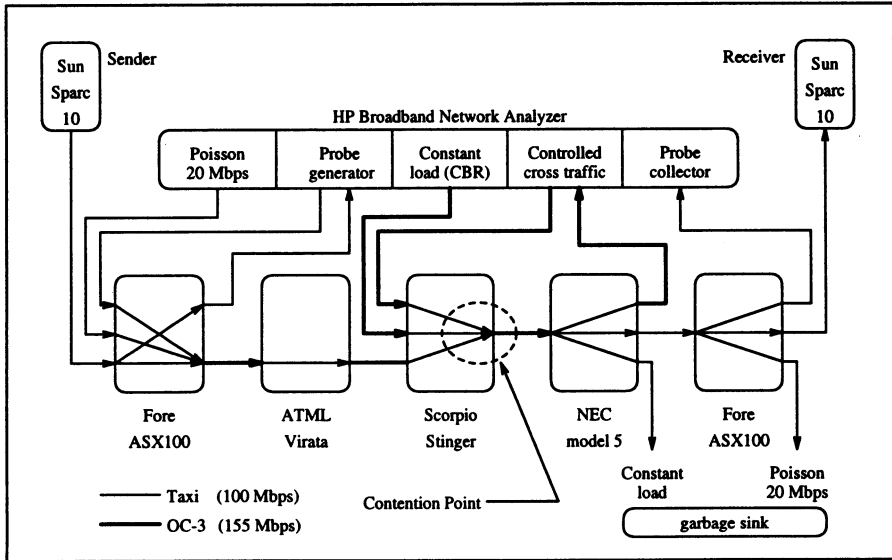
## Experimental setup

We have been using two Sun SparcStation 10 as our end stations. For transport, **qStack** (Huard et al. 1996), a native ATM protocol stack is used. Two motion JPEG and one MPEG-2 VBR video clips were used as multimedia streams. The motion JPEG clips are composed of 15 000 frames and were recorded with different window sizes: medium ( $240 \times 320$  pixels) and large ( $480 \times 640$  pixels). The MPEG-2 video clip has 54 020 frames (approximately 30 minutes at 30 frames per second) with a window size of  $296 \times 720$  pixels and GOP of 12/3. The statistics of the video clips are given in Tables 1 and 2.

The network topology and the interference traffic streams are illustrated in Figure 1. The end-to-end video stream is generated by the workstation on the left. At the first hop (the Fore ASX-100 switch), the video stream is multicasted to the second hop and to the broadband analyzer. Selective probing is performed by the broadband analyzer and probe cells are injected into the network. The probe cells enter the network at first switch and are multiplexed with the video cells. The probes and video cells are also multiplexed with a Poisson cell stream with an average of 20 Mbps. The Poisson stream is used to

**Table 2** MPEG-2 VBR video clip statistics.

	I frames	P frames	B frames	average
number of frames	4502	13 505	36 013	54 020
ave. frame size (bytes)	6657	2886	1945	2573
min. frame size (bytes)	1062	355	322	322
max. frame size (bytes)	16 219	14 913	15 348	16 219
std deviation of frame size	4721	4376	1825	2238

**Figure 1** Network topology.

add some cell delay variation and to interleave the video cells so that they are not all consecutive. The combined flow (video cells, probe cells and Poisson cross traffic) goes through a set of four ATM switches and OC-3 links. At the Scorpio switch (in the middle), two cross traffic streams are injected: a constant bit rate (CBR) stream and a controlled bulk arrival cross traffic stream. The CBR stream is used to help filling up the queue at the contention point so that buffer overflow can occur more easily (77.5 Mbps CBR is injected when the medium window motion JPEG clip or the MPEG-2 VBR clip are played, and 74 Mbps CBR is injected when the large window motion JPEG clip is played). The controlled cross traffic consists of batches of cells injected into the network at line speed (155.52 Mbps). The batch arrival process is Poisson with an average arrival rate that can be set from 0.016 to 100 (i.e., mean batch interarrival time from 10 msec to 60 sec). The batch sizes are geometrically distributed with an average that can be set anywhere from 10 to 50 000 cells.

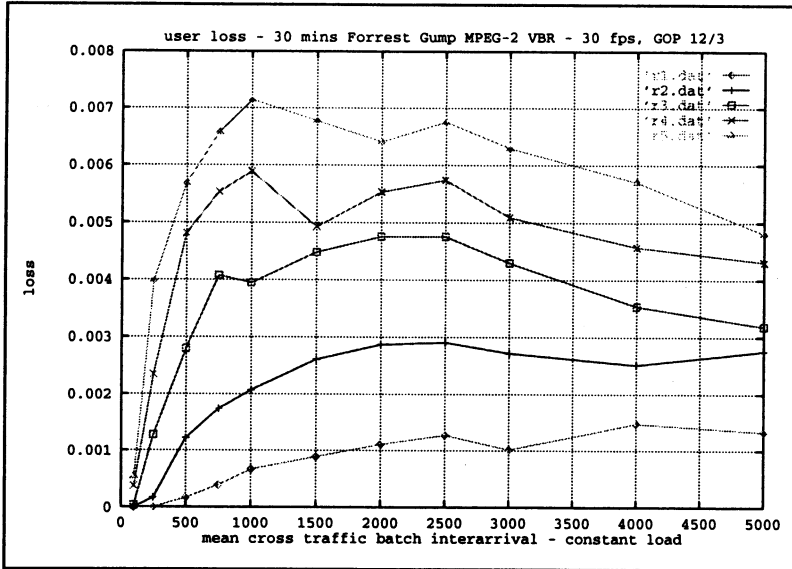


Figure 2 User frame loss for the MPEG-2 clip.

The batch arrival rate and the average batch size can be changed dynamically to obtain different cross traffic load scenarios. Finally, at the last hop, the stream is demultiplexed. The probe cells are sent back to the broadband analyzer for real-time cell level measurements of QoS, the video stream cells are sent to the workstation for reassembly, measurements and playback, and, the Poisson cross traffic is sent into the “garbage sink.”

#### 4 EXPERIMENTAL RESULTS

Figures 2 and 3 show the frame loss and the cell loss for the MPEG-2 VBR clip under various cross traffic loads. Each curve corresponds to a constant load of the controlled cross traffic. Along the x-axis, the mean batch interarrival time increases (the arrival rate decreases). To maintain a constant load, the average batch size is proportionally increased such that the ratio of mean batch size to mean interarrival time remains constant.

Figure 4 shows the ratio of the frame loss to the cell loss. Each point was obtained by dividing the measured frame loss by the corresponding measured cell loss. As can be seen from the figure, one can tolerate a cell loss larger than the one prescribed by Equation 1 since all the points in the figure are below 54, the ratio of average frame size to the cell size. Finally, Figures 5 and 6 show the corresponding frame gap loss and cell gap loss of the MPEG-2 clip.

Similar graphs were obtained for the motion JPEG video clips. Figures 7 and 8 show the ratio of frame loss to cell loss for the medium and large

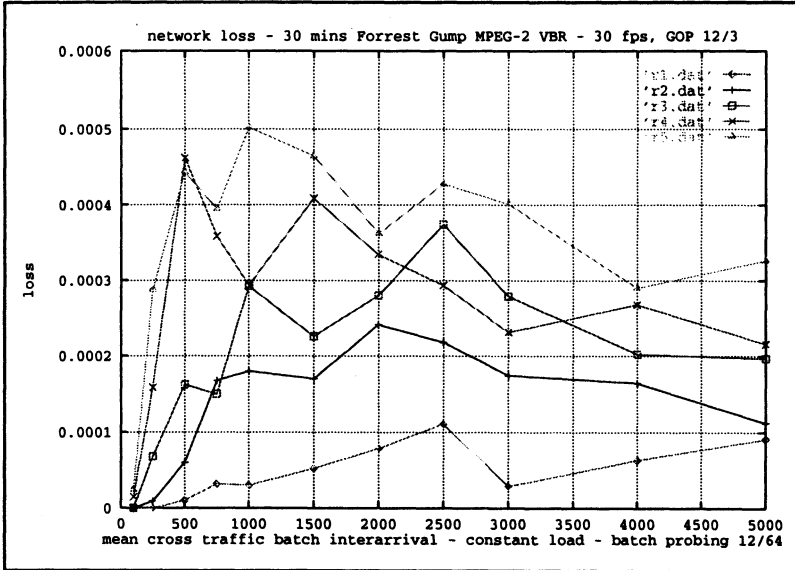


Figure 3 Network cell loss for the MPEG-2 clip.

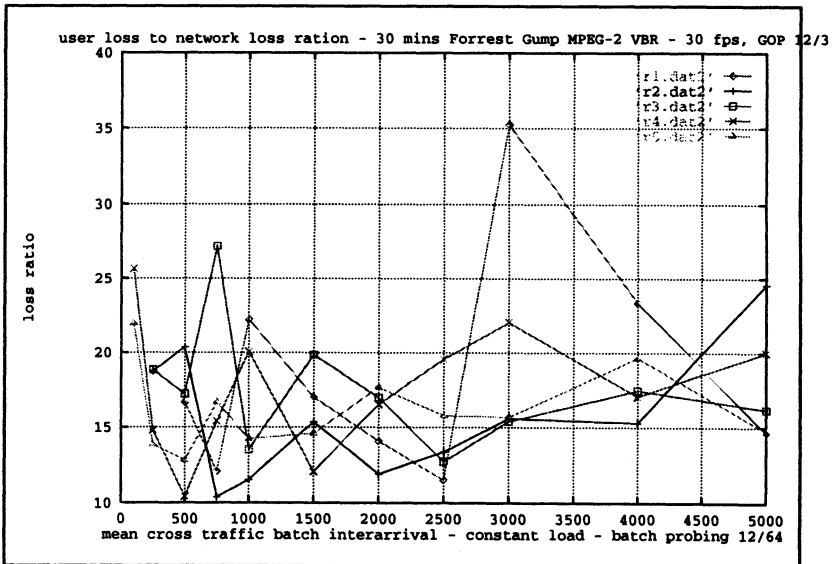


Figure 4 Frame loss to cell loss ratio for the MPEG-2 clip.

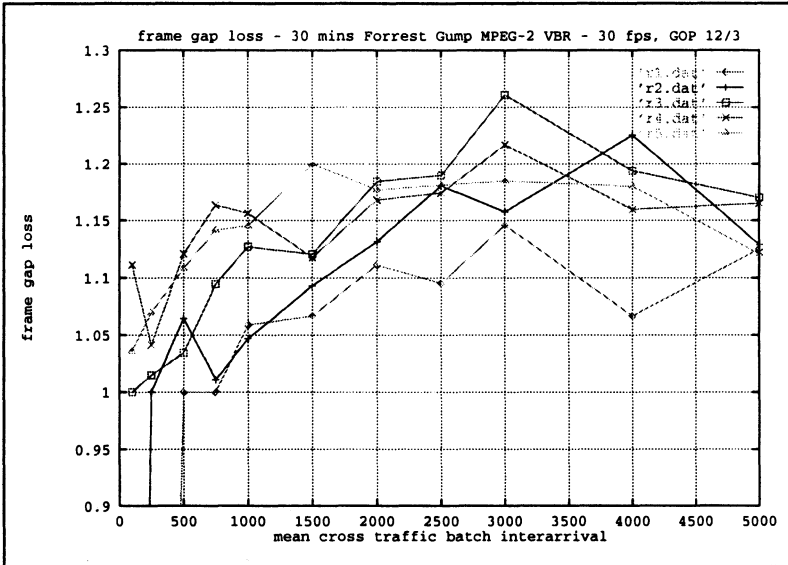


Figure 5 User frame gap loss for the MPEG-2 clip.

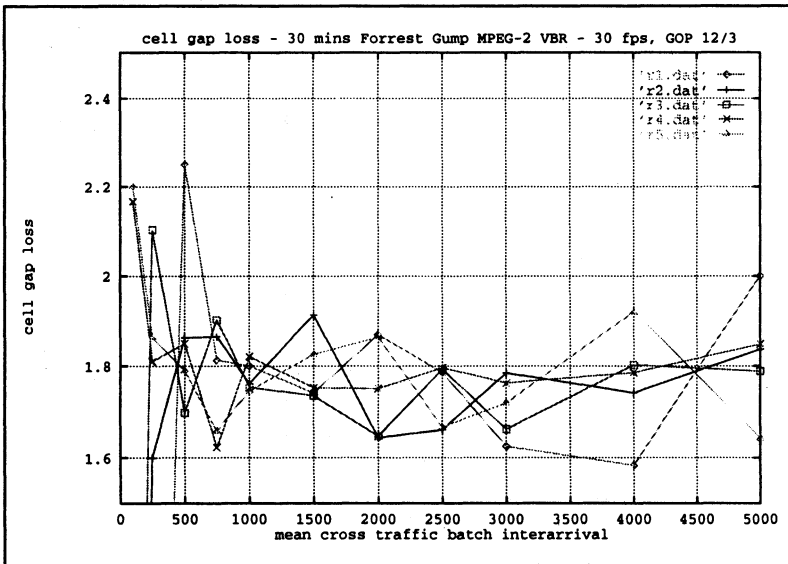


Figure 6 Network cell gap loss for the MPEG-2 clip.



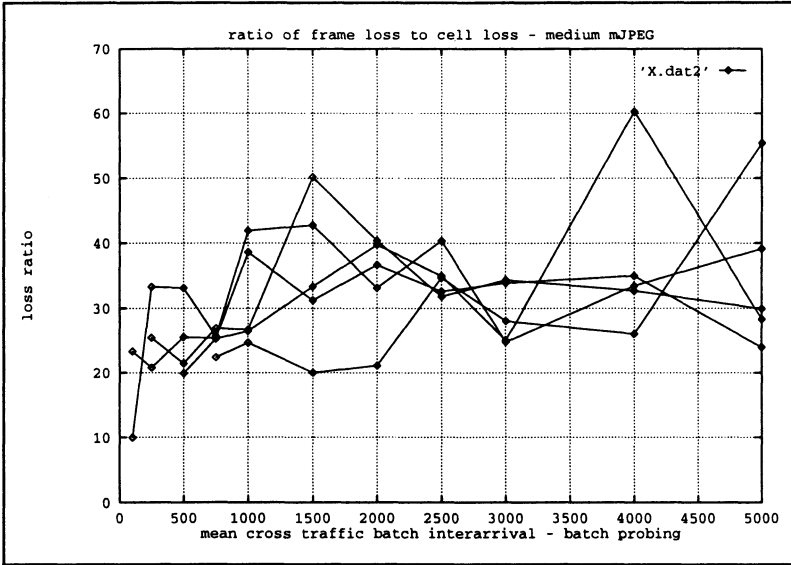


Figure 7 Frame loss to cell loss ratio for the medium motion JPEG.

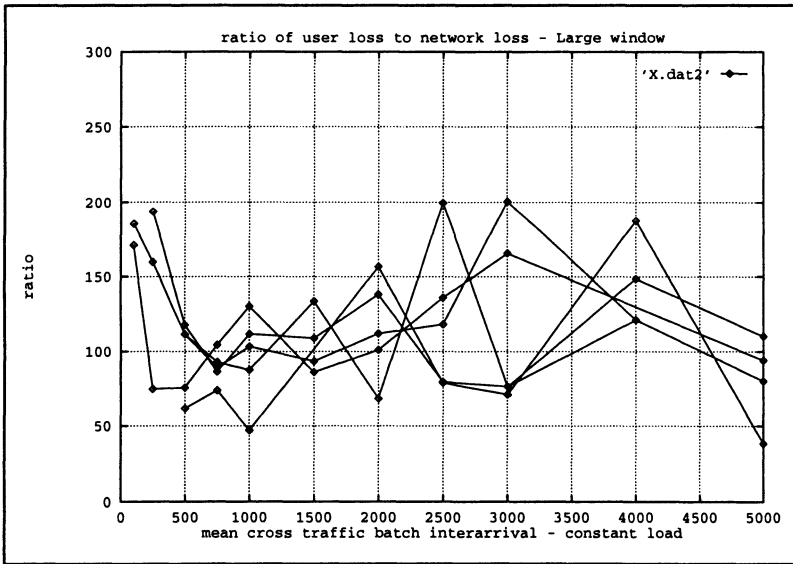


Figure 8 Frame loss to cell loss ratio for the large motion JPEG.

As Figure 4 already indicated, one can tolerate more loss than Equation 1 predicted since all the points on the figures are below the value of the ratio of average frame size to cell size, 133 and 312, respectively. That is, the prescription given by Equation 1 is too conservative by a factor of about 3.

## 5 RELATED WORK

In Zitterbart (1996), a general framework for QOS management from user-to-user is presented. Simple mapping rules such as the one presented in Section 2 are given. A QOS management system that performs QOS monitoring is also described.

The one-to-one translation approach described in (Nahrstedt & Smith 1995) and (Nahrstedt & Smith 1996) is comparable to our approach. It considers mapping between application and network level QOS and formulates arithmetic rules similar to the one proposed in Section 2. The main difference resides in the choice of QOS parameters. In (Nahrstedt & Smith 1995), a model for an endpoint entity called QoS Broker is presented. In (Nahrstedt & Smith 1996) the model is further developed to coordinate the end-system resource management tasks: QOS mapping, admission control and task scheduling. Similar functionality is provided in the **xbind** broadband kernel (Chan et al. 1996) and (Lazar et al. 1996), but not addressed in this paper. The QOS broker of (Nahrstedt & Smith 1995) is comparable to the QOS manager referred to in (Zitterbart 1996).

## 6 CONCLUSION

In this paper, we have presented a framework for studying QOS mapping. As part of this research, we have developed a platform for evaluating end-to-end QOS by performing concurrent network, transport and application level measurements. The early set of concurrent QOS measurements have shown that the typical loss mapping rule given in the literature is too conservative by a factor of about 3.

In order to obtain empirical QOS mapping rules, more data is being collected for future analysis. Furthermore, various network topologies and cross traffic patterns will be tested for validating the empirical mapping rules. Finally, to test the sensitivity to the end-system behavior, we will carry out measurements using various implementation of user space transport protocol stacks (Huard 1996, Huard et al. 1996).

## REFERENCES

- Chan, M. C., Huard, J.-F., Lazar, A. A. & Lim, K.-S. (1996), On realizing a broadband kernel for multimedia networks, in 'Proceedings of the Third COST 237 International Workshop on Multimedia Telecommunications and Applications', Barcelona, Spain.
- Huard, J.-F. (1996), **kStack**: A user space native-mode ATM transport layer with QOS support, Technical Report CU/CTR TR 463-96-29, Center for Telecommunications Research, Columbia University, New York,

- NY. URL: <http://comet.ctr.columbia.edu/software/kstack>.
- Huard, J.-F., Inoue, I., Lazar, A. A. & Yamanaka, H. (1996), Meeting QOS guarantees by end-to-end QOS monitoring and adaptation, in 'Workshop on Multimedia and Collaborative Environments of the Fifth IEEE International Symposium On High Performance Distributed Computing (HPDC-5)', Syracuse, NY, pp. 348–355.
- Lazar, A. A., Lim, K.-S. & Marconcini, F. (1996), 'Realizing a foundation for ATM interoperability with the binding architecture', *IEEE J. Select. Areas Commun.* 14(7), 1214–1227.
- Nahrstedt, K. & Smith, J. M. (1995), 'The qos broker', *IEEE Multimedia* 2(1), 53–67.
- Nahrstedt, K. & Smith, J. M. (1996), 'Design, implementation and experiences with the omega end-point architecture', *IEEE J. Select. Areas Commun.* 14(7), 1263–1279.
- Zitterbart, M. (1996), User-to-user qos—management and monitoring, in 'Fifth IFIP Int'l Workshop on Protocols for High-Speed Networks (PfHSN '96)', Sofia-Antipolis, France, pp. 219–234.

## 7 BIOGRAPHY

*Jean-François Huard* received the B.Eng (EE) in 1990 from University of Montréal (Ecole Polytechnique), the M.A.Sc. (EE) in 1992 from Concordia University (Montréal) and the M.Phil. in 1994 from Columbia University, New York. He is currently a Ph.D. candidate in the Department of Electrical Engineering and a Graduate Research Assistant with the COMET Group in the Center for Telecommunications Research at Columbia University. His current research interests are in the area of high performance QOS-aware transport protocols. During the summers of 94 and 95, he was with AT&T Bell Labs, Murray Hill, NJ. In 94 he was involved in the XUNET project, working on fault management. In 95, he was involved in the development of AT&T WATSON ASR for Windows 95 and NT. Mr. Huard was awarded a Centennial Scholarship of the NSERC of Canada (1990–94).

*Aurel A. Lazar* is a professor of Electrical Engineering at Columbia University. His research interests span both theoretical and experimental studies of telecommunication networks and multimedia systems.

The theoretical research he conducted during the 1980s pertains to the modeling, analysis and control of broadband networks. He formulated optimal flow and admission control problems and, by building upon the theory of point processes, derived control laws for Markovian queueing network models in single control as well as game theoretic settings. He was the chief architect of two experimental networks, generically called MAGNET. This work introduced traffic classes with explicit quality of service constraints to broadband

switching and led to the concepts of schedulable, admissible load and contract regions in real-time control of broadband networks.

In the early 1990s his research efforts shifted to the foundations of the control, management and telemedia architecture of future multimedia networks. His involvement with gigabit networking research led to the first fully operational service management system on ATM based broadband networks. The system was implemented on top of AT&T's XUNET III gigabit platform spanning the continental US. His management and control research pioneered the application of virtual reality to the management of ATM-based broadband networks.

His current research in broadband networking with quality of service guarantees focuses on modeling of video streams and analyzing their multiplexing behavior, with emphasis on multiple time scales and subexponentiality.

Professor Lazar is also leading investigations into multimedia networking architectures supporting interoperable exchange mechanisms for interactive and on demand multimedia applications with quality of service requirements. The main focus of this work is on building a broadband kernel (also known as **xbind**) that enables the rapid creation, deployment and management of multimedia services. Resource allocation and networking games algorithms form the core of this architecture. Professor Lazar was instrumental in establishing the OPENSIG (<http://comet.ctr.columbia.edu/opensig>) international working group with the goal of exploring network programmability and next generation signalling technology.