Providing Application-Level QoS*

Simon S. Lam
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712-1188

TR-96-26 October 14, 1996

1 Introduction

Most application-level data units are too large to be carried in a single packet (or ATM cell) and must be segmented for network delivery. To an application, the end-to-end delays and loss rate of its data units are more relevant performance measures than ones specified for individual packets or cells. For example, a video picture may be segmented and carried in a sequence of IP packets for delivery over an IP network, or in a sequence of ATM cells for delivery over an ATM network. To an end user, the delay incurred to deliver the entire video picture is more important than the delays of individual IP packets or ATM cells.

The objective of this paper is to briefly review the issues of providing application-level quality of service (QoS) and different approaches for addressing these issues.

In what follows, we use "ADU" to denote an application data unit, or a protocol data unit, that is segmented and carried by a sequence of cells (or packets) for network delivery. In some of the references cited, an ADU is also called a *message* or a *packet*. The sequence of cells carrying the ADU is also called a *block* or a *burst*.

2 Analyzing application-level performance

The approach of Cidon et al. is to employ analytic modeling [4] and simulations [5] to determine the performance of messages (ADUs) delivered by an ATM network that provides cell-level guarantees (cell loss probability, cell delay, and cell delay jitter). It was observed that the message loss probability of a connection differs greatly from the cell loss probability guaranteed. The relation between message and cell loss probabilities depends strongly on characteristics of the connection, particularly, its peak rate and message size. It also depends on whether the cell loss probability is a per-connection guarantee or a global guarantee. The message mean delay and delay jitter were also determined empirically and compared with cell mean delay and delay jitter.

*Research supported in part by National Science Foundation under grant no. NCR-9506048 and the Texas Advanced Research Program grant no. 003658-220.

3 Improving application-level performance

The objective of the previous approach is to gain insights into factors that affect application-level performance in existing ATM networks with no additional support. In this section, we review approaches to improve application-level performance by providing additional support in switches and/or end systems.

3.1 Early packet discard

Consider a packet (ADU) that is segmented and carried in a number of cells. The packet is lost and requires retransmission by its source if any one of the cells is lost. Therefore, when a channel is congested (i.e., buffer utilization exceeding a threshold) and cell discard is needed, entire packets should be discarded rather than individual cells. There are several variants of this idea: early packet discard (EPD) [10], and EPD with hysteresis and fair EPD with hysteresis [12].

Clearly, using any one of these methods would improve the packet loss probability of an ATM network for the same cell loss guarantee. Implementing these methods would require that switches keep additional per-connection state information. Switches also need to recognize the last cell of each packet. For an AAL5 packet, in particular, it is suggested that switches determine the end of an AAL5 packet by examining the AAL5 headers of cells [10].

3.2 Forward error correction

The use of forward error correction (FEC) has been proposed to improve not only the application-level loss probability but end-to-end delay as well [11]. Consider a message (ADU) that is segmented and carried in n cells from a source to a destination. Suppose the source also sends a redundant "parity cell." The destination can reconstruct the original message as soon as n out of the $n+1$ cells have arrived. Thus the use of a parity cell would likely improve both the loss probability and end-to-end delay of the message.

However, since the use of FEC generates additional traffic (parity cells) it is not immediately obvious that there would be a net gain in overall network performance. Furthermore, if only some connections use FEC, the improved performance of these connections is at the expense of connections that do not use FEC. These issues were investigated, and it was observed that the use of FEC would provide a net performance gain only for networks with heterogeneous traffic [1].

The use of FEC requires additional support at the two end points of a connection (e.g., in the end systems). Also, support can be added in switches to selectively discard cells; if each message has only one parity cell, the objective would be to avoid discarding more than one cell from a message so that the message loss probability is further improved [11].

3.3 ATM block transfer

The ATM block transfer (ABT) capability being standardized by ITU-T [7, 3] has been known for some time as a fast reservation protocol [2]. The concept of a block was introduced to represent some high-level protocol data unit(s). Conceptually, a traffic source generates a sequence of blocks, each of which is a sequence of cells bracketed by two RM cells. The leading RM cell of a block requests a reserved bandwidth for the block from each switch along its path, and the trailing RM cell releases the reserved bandwidth. Initially, an ABT connection is set up with a peak cell rate and a peak rate for RM cells (but no allocated bandwidth).

Since a block of cells is either discarded or accepted entirely by a switch, ABT improves block loss probability in the same manner as early packet discard. Compared to early packet discard, ABT requires switches to maintain more per-connection state information and ABT uses a different admission control policy (on the basis of reserved bandwidth rather than buffer utilization).

4 Guaranteeing application-level performance

As described in Section 3, application-level performance can be improved with additional support in switches and end systems. We can go one step further, namely: generalize the QoS parameters for real-time variable bit rate (VBR) service from performance guarantees for cells to *performance guarantees for blocks*, which represent application data units. Specifically, we propose to generalize the QoS parameters, cell transfer delay (CTD) and cell loss rate (CLR), to block transfer delay and block loss rate, respectively. Note that such a generalization is backward-compatible since a block is a sequence of cells, which subsumes a single cell as a special case.

4.1 Burst scheduling networks

The class of burst scheduling networks presented in [8, 9] supports a real-time VBR service for blocks. Each flow is modeled as a sequence of bursts, each of which is a sequence of packets (cells) that carry an application data unit (same as block in ABT). The packet scheduler is highly efficient and allocates bandwidth to a flow on a per-burst basis (like ABT). Tight bounds on the end-to-end delays of bursts are proved. Two burst-based admission control policies are described in [13] to provide the following classes of service:

- *Real-time VBR service with no loss.* Each flow in this class is allocated its peak bandwidth at connection setup. Subsequently, each block in the flow will be admitted by every switch in its path.[1]

- *Real-time VBR service at a specified loss rate.* For applications that can tolerate a small block loss rate, their flows are admitted at connection setup time on the basis of their peak and sustained rates with overbooking allowed at each switch. The block loss rate of these flows can be calculated and negotiated at connection setup [13].

4.2 End system support

The end points of a connection are typically application programs running in the user address spaces of workstation operating systems. To provide application-level performance

[1] Loss due to buffer overflow is possible.

guarantees between these end points, additional support in the workstation operating systems is needed, particularly, CPU scheduling that provides QoS guarantees, as well as techniques for moving data efficiently between user address space and network interface [6, 14, 15].

References

[1] Ernst W. Biersack. Performance evaluation of forward error correction in ATM networks. In *Proceedings of ACM SIGCOMM '92*, pages 248–257, August 1992.

[2] P. Boyer. A reservation principle with applications to the ATM traffic control. *Comp. Networks and ISDN Sys.*, 24:321–334, 1992.

[3] Thomas M. Chen, Steve S. Liu, and Vijay K. Samalam. The available bit rate service for data in ATM networks. *IEEE Communications Magazine*, pages 56–71, May 1996.

[4] I. Cidon, R. Guerin, and A. Khamisy. An investigation of application level performance in ATM networks. In *Proceedings of IEEE INFOCOM '95*, Boston, MA, April 1995.

[5] I. Cidon, R. Guerin, A. Khamisy, and K. N. Sivarajan. Cell versus message level performances in ATM networks. *Telecommunication Systems*, 5:223–239, 1996.

[6] R. Gopalakrishnan and Guru Parulkar. Efficient user space protocol implementations with QoS guarantees using real-time upcalls. Technical Report WUCS-96-11, Washington University in St. Louis, 1996.

[7] ITU-T Rec. I.371. *Traffic Control and Congestion Control in B-ISDN.* Perth, U.K., Nov. 6-14, 1995.

[8] Simon S. Lam and Geoffrey G. Xie. Burst scheduling networks. Technical Report TR-94-20, University of Texas at Austin, July 1994. Revised, August 1996. An abbreviated version in *Proceedings of IEEE INFOCOM '95*, April 1995.

[9] Simon S. Lam and Geoffrey G. Xie. Group priority scheduling. Technical Report TR-95-28, University of Texas at Austin, July 1995. Revised, September 1996. An abbreviated version in *Proceedings of IEEE INFOCOM '96*, March 1996.

[10] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. In *Proceedings of ACM SIGCOMM '94*, London, England, August 1994.

[11] Nachum Shacham and Paul McKenney. Packet recovery in high-speed networks using coding and buffer management. In *Proceedings of IEEE INFOCOM '90*, pages 124–131, 1990.

[12] J.S. Turner. Maintaining high throughput during overload in ATM switches. *Proceedings of IEEE INFOCOM '96.* pages 287–295, 1996.

[13] Geoffrey G. Xie and Simon S. Lam. Real-time block transfer under a link sharing hierarchy. Technical Report TR-96-19, University of Texas at Austin, June 1996. Available from http://www.cs.utexas.edu/users/lam/NRL/ . To appear in *Proceedings IEEE INFOCOM '97.*

[14] David K. Y. Yau and Simon S. Lam. Operating system techniques for distributed multimedia. Technical Report TR-95-36, University of Texas at Austin, July 1995. Revised, January 1996. An abbreviated version in *Proceedings MMCN '96*, January 1996.

[15] David K. Y. Yau and Simon S. Lam. Adaptive rate-controlled scheduling for multimedia applications. Technical Report TR-96-17, University of Texas at Austin, April 1996. Revised, September 1996. An abbreviated version in *Proceedings ACM Multimedia '96*, November 1996.