Experimental Evaluation of ATM Congestion Control Mechanisms

Prashant Chandra, Allan Fisher, Corey Kosak, and Peter Steenkiste
School of Computer Science
Carnegie Mellon University

1 Introduction

Tt is widely believed that closed-loop per-VC flow control is needed to effectively deal with congestion for ABR traffic in ATM networks. The many ATM flow control mechanisms that have been proposed can be divided in two classes. A first class consists of roundtrip-based rate control algorithms in which a source selects a transmit rate for each VC, based on feedback from the switches along its path [2, 4, 11, 3]. Feedback can for example consist of the EFCI bit in the ATM cell header, or of an explicit rate calculated by the switches. A second class consists of hop-by-hop, credit-based flow control, in which the receiver on each link provides the sender with information on buffer status for each VC. Examples include DEC AN2 [12] and Credit Net [9, 8]. An overview of these algorithms can be found in [6, 13]. A class of rate-based algorithms is currently being standardized by the ATM Forum [14], although several companies are still pursuing a credit-based alternative called Quantum Flow Control [5].

The various flow control algorithms have been evaluated and compared extensively using simulation, for example using the ATM Forum test configurations [1, 15]. Experimental results are however limited and focus on persistent sources [16, 10]. In this paper we present an experimental comparison of several ATM flow control mechanisms using a more diverse set of benchmarks including both persisten and busty sources for a single-congestion link configurations.

2 Evaluating Congestion Control Mechanisms

ATM flow control algorithms can be evaluated along many dimensions including data loss, responsiveness to changes in the network and application demands, overhead, effective utilization of network resources and fairness. We describe a test suite that combines a suitable set of metrics, traffic models and configurations, to evaluate single-congested-link scenarios in a quantitative way; we have to limit the scope of our investigation because we are currently working within a relatively small testbed. The configuration used is a peer-to-peer configuration shown in Figure 1.

Experiment 1: Static rate matching In this experiment, only S1 is used. It is a persistent packet source, i.e. it sends packets at line rate. The output link of SW is run at a lower rate than the input link, and the congestion control mechanism is used to provide rate-matching. We vary switch buffer sizes and link distances, and measure data loss and throughput. This experiment provides insight into the efficiency, stability and resource requirements of the various mechanisms.

Experiment 2: Dynamic rate matching In this experiment S1 sends packets of varying lengths, and receives short acknowledgements form S3. The delay of the link is varied, and the request-response latency is measured for a variety of request sizes. This experiment shows how effectively the congestion control mechanisms can provide bandwidth in response to bursty demands.

Experiment 3: Static congestion In this experiment, S1 and S2 are persistent packet sources. The delay of the input link of S1 is varied between LAN and WAN dimension. We measure data loss and throughput of each stream, as well as utilization of the link to S3, as buffer sizes at the switch SW are varied. This experiment sheds light on data integrity, efficiency, scalability and long-term fairness of the congestion control method under consideration.

Experiment 4: Dynamic congestion In this experiment, S1 is a persistent source and S2 is a ON-OFF source. S1 is connected via a WAN link to the switch SW. The ON-OFF duty cycle of S2 is varied. We measure utilization of the link to S3 and the throughput and loss rate of S1. This provides insight into the responsiveness of the congestion control algorithm, and hence its efficiency under dynamic conditions.

Experiment 5: Dynamic fairness In this experiment S1 is again connected via a long link to SW. Both S1 and S2 are request/response sources and they query a server S3. The lengths of the requests are varied and response time and number of requests received at S3 over the long term are measured. This gives insight about flexibility and fairness under dynamic conditions.

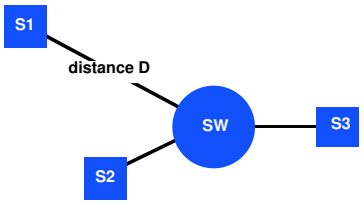

Figure 1: Test configuration

3 Experimental Infrastructure

Experiments were run on Credit Net LAN, an experimental ATM local area network. Credit Net LAN incorporates OC-3 and OC-12 SONET switch and host interfaces, though this study uses only the OC-3 hardware. The Credit Net LAN switch hardware[10] provides native support for per-VC buffering and feedback of absolute buffer credits. The Credit Net LAN adapter[7] provides a variety of mechanisms for implementing congestion control algorithms, including a microengine and an optional imbedded microprocessor.

These facilities allow implementation of a variety of endpoint behaviors with high performance. Furthermore, we can use the adapter in a "virtual port card" configuration to provide switch congestion control functionality beyond that provided in switch hardware: we model a switch port using a host with two adapters. In combination, these adapters and the host processor can provide both input port and output port functionality for a single port of the switch. The virtual port card can also model the behavior of long links by delay cells.

4 Congestion Control Implementations

This section briefly describes the congestion control implementations used in our experiments.

Credit with static buffering This method uses the native flow control capabilities of the Credit Net LAN switch, which implements the FCVC N123 algorithm[9]. The N123 algorithm is implemented at the host, for these tests, on the embedded processor described above. Buffer space on the switch is allocated in proportion to the propagation delay of the upstream link, in order to eliminate data loss. Because of limits on buffer space that we have not yet worked around, we report results only for limited latencies in this submission; credit results for all latencies will appear in the final version.

Credit with adaptive buffering This mechanism uses the same host implementation as the previous case; the only change is at the switch. Instead of making a worst-case allocation of buffer space per VC, the switch allocates space to each VC in proportion to its bandwidth, adjusting allocations periodically to track bandwidth changes. We implement adaptive buffering using the virtual port card approach; the host processor of the virtual port card runs the FCVC N123 algorithm and performs periodic buffer reallocation. The implementation of this scheme is under way.

Rate control with EFCI This case and the following one implement two variants of the ATM Forum ABR congestion control mechanism[14]. To make implementation easier, we have simplified the format of the resource management cell, and have left unimplemented a variety of timeout-related behaviors that do not arise in our test cases. The endpoint behaviors are implemented using a combination of the microcode and embedded processor cabilities of the host adapter, while the switch behaviour is implemented on a virtual port card.

Rate control using EPRCA The Enhanced Proportional Rate Control Algorithm (EPRCA) is a variant of the ATM Forum ABR scheme in which switches provide sources with explicit rate bounds. Endpoint behavior is implemented as in the previous case, except that the source updates rates to the value provided in the explicit rate field of returning RM cells. Switch behavior is more complicated: the switch must intercept forward RM cells and periodically use their current rate information to compute a fair share rate. It must also set the explicit rate field in reverse RM cells to the minimum of the incoming rate and the fair share rate. The fair share rate is computed using the exponential averaging algorithm specified by the ATM Forum.

5 Measurements

In this section we present and analyze our performance measurements for the different congestion control implementations. The variable-delay link from S1 to SW was run on a 90 MHz Pentium PC, and the virtual port card leading to S3 was run on a 133 MHz Pentium PC. S1, S2 and S3 were all 90 MHz Pentium PCs.

5.1 Static rate matching

Experiment 1 evaluates rate matching of a static stream traversing a faster link feeding into a slower link. The slow link is throttled to 1/6 of OC-3 bandwidth, and the source attempts to send at 70 Mbps. Figure 2 shows how the throughput of valid data, for IP MTU-sized packets, depends on the propagation delay of the fast link for the three flow control schemes. For EFCI, "-f" (fast) corresponds to a rate increase of 10 Mbps and a congestion slowdown of 50%; "-s" (slow) corresponds to an an increase of 100 Kbps and a decrease factor of 10%. The number following -f or -s is the congestion threshold in cells; all cases use a buffer that holds 4000 cells.

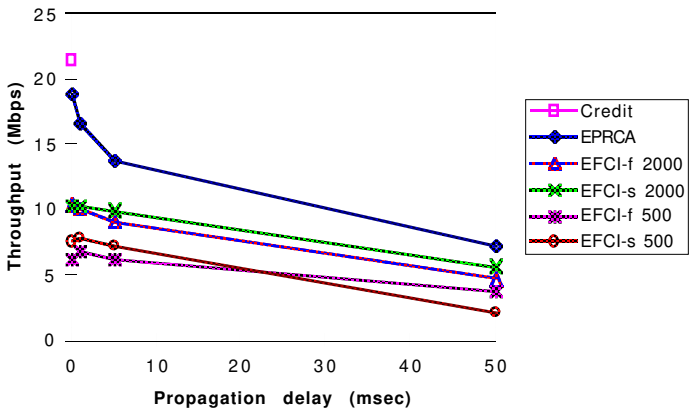

Figure 2: Throughput as a function of link delay

As one would expect, credit with static buffering gives by far the best performance (in the LAN environment), followed by rate control using EPRCA and using EFCI. The performance of both rate control schemes degrades as the distance increases. This loss of efficiency is the result of oscillations. For EFCI, throughput is better using EFCI/drop threshold of 2000/4000 bytes compared to 500/4000 bytes. Note however that there is a cell loss of about 30% using the 2000/4000 threshold for the "fast" rate adaptation parameters, so with a reliable transport protocol added, this case is unlikely to give good application level performance. We are still investigating the reason for the poor EPRCA result.

5.2 Dynamic rate matching

Experiment 2 characterizes the speed with which burst bandwidth is allocated. We measure response time for two link latencies and, in the EFCI case, for several settings of rate adaptation parameters. Requests are sent at 70 Mbps, and the ABR implementations are allowed to settle to their initial cell rates between requests.

Figure 3(left) shows the response time as a function of request size in a LAN with negligible propagation delay. The EFCI curves are marked INIT+INCR, where INIT is the initial rate in Mbps and INCR is the rate increase in Mbps applied each time a resource management cell without a congestion indication is received. Similarly, the EPRCA initial rate is indicated in Mbps. Because this scenario does not induce congestion, rates are never lowered except when the stream is idle. In this graph, the static credit algorithm provides the best response time, since its worst-case buffer allocation requires no ramp-up delay. The EFCI response times are dominated by the initial data rate for short requests, since the entire request may be completed before any returning RM cells are received. For longer packets and lower initial rates, returning RM cells have more effect. EPRCA performs slightly better than EFCI on longer requests, since its first returning RM cell permits it to ramp up to full bandwidth immediately.

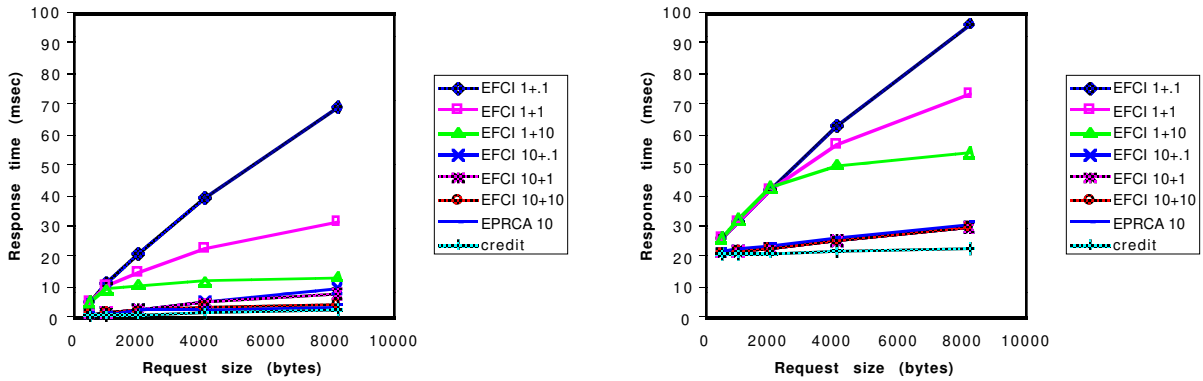

Figure 3: Response time as a function of request size (LAN): regular (left) and magnified scale (right)

Figure 3(right) shows response time using a link with a propagation delay of 10 msec. All response times in this graph have an offset of 20 msec, corresponding to the physical round-trip time of the long link. In this graph, the static credit algorithm again provides the best response time. The reason is again the lack of ramp-up delay: the per-VC buffer in the switch is 210 cells, which is sufficient to hold the Request, so the sender does not have to wait for a credit cell, and the distance has no effect on performance. Because of the long delay before RM cells return, initial rate is an even more important factor for the EFCI scheme than before.

5.3 Static congestion

This experiment creates congestion between a stream traversing a long link and one traversing a short link. Figure 4 shows packet data throughput for this scenario as a function of switch buffer size allocated to the high-latency connection. In this case, the throughput reflects both loss of efficiency, unfairness and loss of data due to the congestion control scheme used. Note that the EPRCA results are for a bottleneck link of 15 Mbps instead of 25 Mbps, but the near connection still achieves a better throughput than either of the EFCI schemes. Rate control using both EFCI and EPRCA results in very unfair behavior, and in the case of EFCI the difference in bandwidth for the near and far sources increases with buffer size. We conjecture that the unfairness of the EPRCA results is due to EPRCA's "very congested" mode, which, using suggested parameter values, is entered when more than 100 cells are queued. We expect that a larger value, more appropriate for large buffers, would provide better fairness.

5.4 Dynamic congestion

This experiment examines the behavior of a persistent stream, traversing a long link, as it encounters congestion created by a local ON-OFF source. The persistent stream has a delay of 10 ms, and the congested link has a bandwidth of 25 Mbps, except for the EPRCA case, which has a congested link of 15 Mbps. Figure 5 shows how the persistent stream's throughput of valid packet data, for IP MTU-sized packets, depends on the average offered bandwidth of the bursty source. EPRCA performance is best, even though its bottleneck link is slower.

5.5 Dynamic fairness

The final experimental configuration runs two request-response streams, one entering a congested link from a long-delay link (10ms) and the other from a short-delay link. The bottleneck link is 25 Mbps for EFCI and 15 Mbps for EPRCA. Each source generates requests in a Poisson process, generating traffic that will consume half the bandwidth of the congested link. The rate configurations use a 4000-cell buffer with a congestion threshold of 2000 cells, an initial rate of 10 Mbps, an increase rate of 100 kbps, and a congestion reduction factor of 10%.

Figure 6 shows response time and rate at which requests are received for each stream as a function of the request size used. There were no lost cells in the EPRCA test, so the received rate was identical to the transmit rate (equivalent

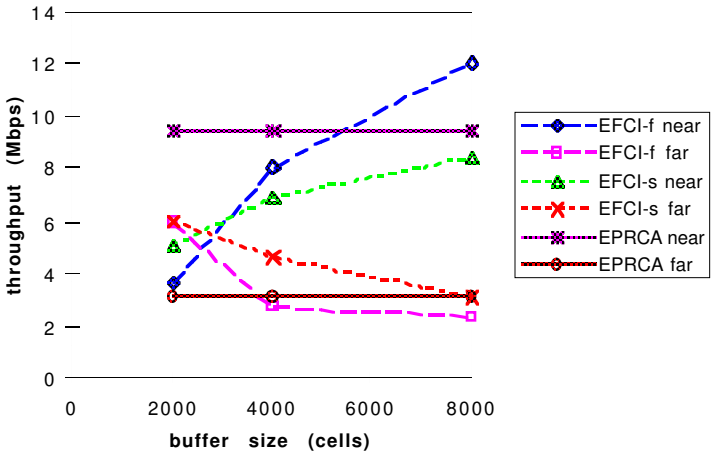

Figure 4: Throughput as a function of buffer size

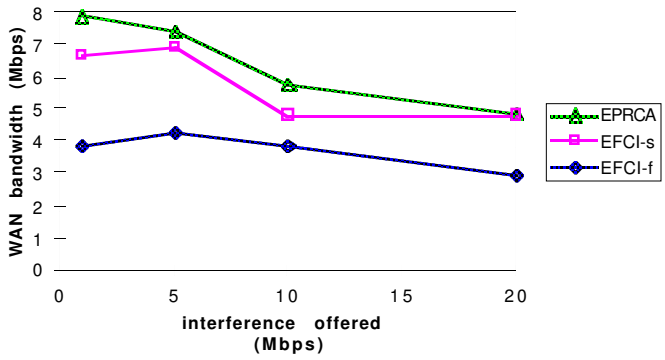

Figure 5: Throughput of a persistent WAN source with bursty interference

to the upper-bound curve). With EFCI, there is considerable data loss. None of the rate-based flow control schemes deal very well with this dynamic scenario. The EPRCA near source is the only one that gets reasonable response time (20-50 msec), while the response time in all cases is unacceptably high. Both the EFCI and EPRCA are very unfair. Again, we conjecture that the unfairness of EPRCA is due to its "very congested" mode.

6 Discussion

This paper presents a comprehensive experimental evaluation of a variety of mechanisms for congestion control in a single-congested-link scenario. A first conclusion that we draw from this study is that, although operating parameters for rate control algorithms can be tuned to individual situations, no parameter setting is best for all conditions. Especially under dynamically changing loads, this may be problematic. Second, we observed that EFCI rate control is subject to very poor performance due to oscillation, confirming simulation results. In effect, it has no memory of prior onsets of congestion, and so even under static conditions it repeatedly causes congestion and then backs off. Finally, fairness is another problem of the rate control schemes. Even EPRCA, using recommended parameter settings, is subject to seriously unfair behavior in the face of large distance disparity with or without dynamic cross-traffic.

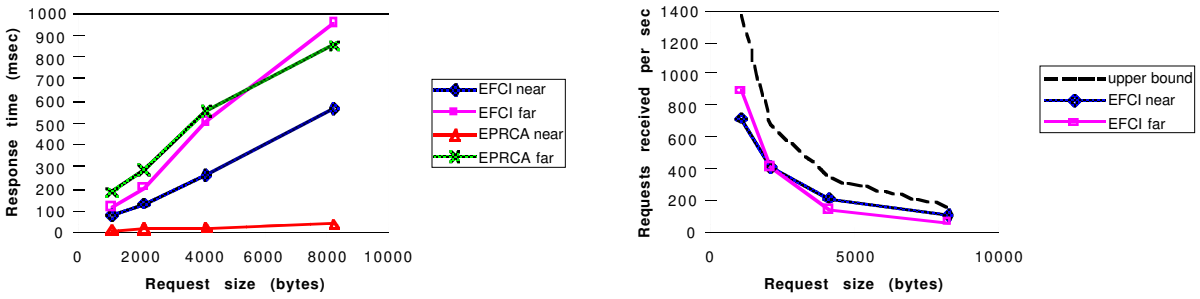

Figure 6: Response time (left) and requests received (right) as a function of request size

References

[1] Andrew W. Barnhart. Baseline model for rate-control simulation. ATM Forum, Traffic Management Working Group submission 94-0399, Hughes Network System, May 1994.

[2] Flavio Bonomi and Kerry Fendick. The rate-based flow control framework for the available bit rate ATM service. *IEEE Network Magazine*, 9(2):25–39, March/April 1995.

[3] Anna Charny and K. K. Ramakrishnan. Time scale analysis of explicit rate allocation in ATM networks. In *Proceedings of INFOCOMM 96*, March 1996.

[4] Anna Charny, K.K. Ramakrishnan, and Anthony Lauck. Scalability issues for distributed explicit rate allocation in ATM. In *Proceedings of INFOCOMM 96*, March 1996.

[5] Mike Gaddis and Walter Kelt (eds). Quantum flow control. Technical Report FCC-SPEC-95-1, Flow Control Consortium, July 1995.

[6] Raj Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, February 1995.

[7] Corey Kosak, David Eckhardt, Todd Mummert, Peter Steenkiste, and Allan Fisher. Buffer management and flow control in the credit net ATM host interface. In *Proceedings of the 20th Conference on Local Computer Networks*, pages 370–378, Minneapolis, October 1995. IEEE.

[8] H.T. Kung, T. Blackwell, and A. Chapman. Credit update protocol for flow-controlled ATM networks: Statistical mulitplexing and adaptive credit allocation. In *Proceedings of the SIGCOMM '94 Symposium on Communications Architectures and Protocols*, pages 101–114. ACM, August 1994.

[9] H.T. Kung and A. Chapman. The FCVC (flow controlled virtual channels) proposal for ATM networks. In *International Conference on Network Protocols*, pages –, San Francisco, CA, Oct 1993. IEEE.

[10] R. Morris and H. T. Kung. Impact of ATM switching and flow control on TCP performance: Measurements on an experimental switch. *GLOBECOM*, November 1995.

[11] Hiroyuki Ohsaki, Masayuki Murata, Hiroshi Suzuki, Chinatsu Ikeda, and Hideo Miyahara. Rate-based congestion control for ATM networks. *COMPCOMM*, April 1995.

[12] Cüneyt Özveren, Robert Simcoe, and George Varghese. Reliable and efficient hop-by-hop flow control. In *Proceedings of the SIGCOMM '94 Symposium on Communications Architectures and Protocols*, pages 89–100, University College, London, London, UK, October 1994. ACM.

[13] K. K. Ramakrishnan and Peter Neuman. Integration of rate and credit schemes for ATM flow control. *IEEE Network Magazine*, 9(2):49–56, March/April 1995.

[14] Shirish S. Sathaye. ATM Forum traffic management specification version 4.0. ATM Forum, Traffic Management Working Group submission 95-0013R6, Fore Systems, June 1995.

[15] Robert J. Simcoe. Test configurations for fairness and other tests. ATM Forum, Traffic Management Working Group submission 94-0557, Digital Equipment Corporation, July 1994.

[16] Hongbo Zhu, Luiz A. Dasilva, Joseph B. Evans, and Victor S. Frost. Performance evaluation of congestion control mechanisms in ATM networks. In *Computer Measurement Group Annual Conference (CMG '95)*, December 1995.