An Adaptive Feedback Control Based Cell Scheduler for ATM Networks

S. Ramaswamy *P. Gburzyński*

Department of Computing Science
615 GSB, University of Alberta,
Edmonton, AB, Canada T6G 2H1
srini@cs.ualberta.ca
pawel@cs.ualberta.ca

Abstract

We describe a cell scheduler for ATM networks based on a predictive adaptive feedback control methodology, whose goal is to maintain the quality of service (QoS) of a set of calls at the desired level, in spite of traffic fluctuations and/or deviations of the actual traffic parameters from those specified at call setup. The scheme is quick and effective and its performance is independent of the distance between the source and the switch, which makes it ideal for high-speed wide-area networks, in contrast to other (reactive) flow control strategies.

1 Introduction

The cell scheduling disciplines proposed in the literature for ATM switches may be divided into two categories. The first category has one scheduler for each output port. The traffic for this output port may be placed in several queues. The scheduler considers all the queues at the same time in order to determine the next cell to be transmitted. Scheduling policies that fall into this category are Virtual Clock [14], Stop-and-Go Queuing [5], Packet Generalised Processor Sharing [9], and Weighted Fair Queuing [4]. Alternatively, the link bandwidth at the output port may be partitioned among the input queues, each queue handling a particular class of traffic. A virtual server operates for each queue at the specified bandwidth, and schedules cells according to a FCFS policy. While the scheduling discipline is now quite simple, the difficulty lies in partitioning the bandwidth at call admission time so that QoS guarantees for each of the queues are satisfied.

In this paper, we assume the availability of a call admission algorithm that indicates the amount of bandwidth to be allocated to a queue receiving traffic from a group of statistically multiplexed sources. Many such schemes have been proposed in the literature, e.g., [3], [6], [8], [11], and [12].

Instead of a hard partitioning of the link bandwidth among the different queues, we assume that the partitions are movable to a small extent specified as a percentage deviation from the allocated bandwidth. By modulating the queue service rate within this range, we maintain the quality of service in spite of traffic fluctuations and traffic parameter deviations. A similar approach was proposed by Pitsillides [10]. However, they chose to regulate the flow of the traffic entering the queue. One problem with this approach is that its performance is adversely affected by the distance between the furthest source and the queue, making it unsuitable for wide-area networks.

2 Control Methodology

The basic idea of the feedback control is shown in figure 1. A single queue, henceforth called the process, has as its output the quality of service provided to the aggregate traffic from a set of multiplexed calls. The input to the process is the service rate. The error between the desired QoS (the reference signal) and the actual QoS is fed to the controller which manipulates the service rate in an effort to reduce the error to zero. In order to do this, the controller must have an idea of the effect of

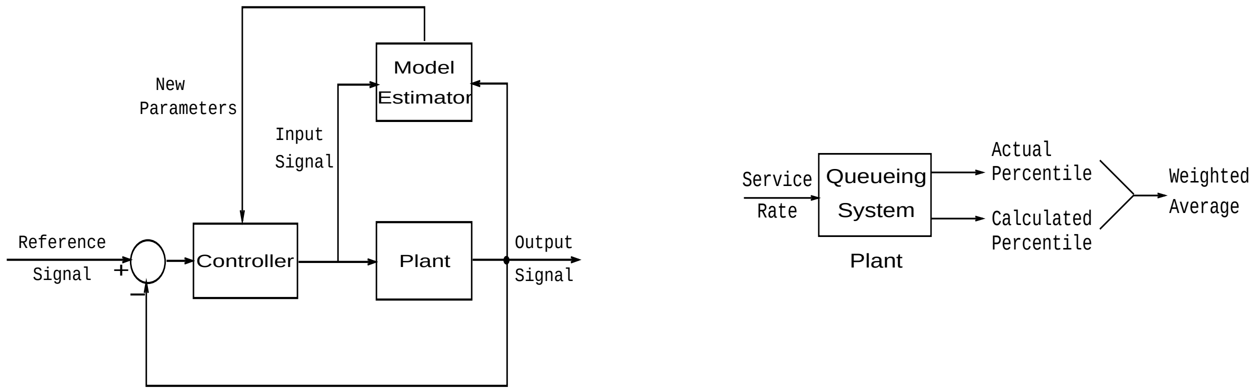

Figure 1. Adaptive Feedback Controller (left) Controlled Process (right)

a change in the service rate (the manipulated variable u) on the QoS (the controlled variable y), i.e., the process gain. For a non-linear process like our queuing system (see section 3), the gain may be a function of the operating point. Therefore, the controller should automatically tune itself to work correctly at any operating point. For this reason, at each sampling instant, we estimate a model of the process from which the gain and dominant time constant can be calculated. This model is then used to tune the controller. Thus, the controller is updated by a tuning device (when the process parameters are constant) and an adapting mechanism (when the process parameters change slowly over time).

2.1 Quality of Service

As in [10], we assume that the desired QoS for a connection is specified as a set

$$\text{Desired QoS} = \left\{p_{cell\ loss}^{max}, \tau_{cell\ delay}^{max}, \tau_{CDV}^{max}\right\}$$

where $p_{cell\ loss}^{max}$ is the maximum cell loss rate, $\tau_{cell\ delay}^{max}$ is the maximum allowable cell delay at the queue, and τ_{CDV}^{max} is the maximum allowable variation in cell delay. The target QoS parameters can be mapped to the reference parameters $E_{overflow}$ and y_{ref} as follows:

$$E_{overflow} = p_{cell\ loss}^{max} \quad \text{and} \quad y_{ref} = \left\lfloor \frac{min(\tau_{cell\ delay}^{max}, \tau_{CDV}^{max})}{\tau_{cell\ time}} \right\rfloor$$

where $\tau_{cell\ time}$ is the time (in seconds) to service a cell. The QoS reference value y_{ref} is the number of buffer places that should accommodate $(1 - E_{overflow})$ * 100 % of the served cells.

In order to maintain the QoS at the desired value, we need to be able to calculate the QoS offered to the aggregate traffic. Since our reference signal is in terms of the number of buffer places required to accommodate a certain percentage of the served cells, our feedback signal should be the number of buffer places actually used to accommodate that percentage. The actual p^{th} percentile can be measured on line, but it is not suitable for control as it responds very slowly to a change in the service rate. Instead, we use the sensor developed by [1], which predicts the p^{th} percentile of the buffer occupancy. The predicted value is a function of the mean, variance, and auto-covariance sum of the net arrival process defined as

$$Y_n = A_n - B_n$$

where A_n is the amount of work entering the queue during the n^{th} sampling interval (of duration T_{sample}), and B_n is the amount of work processed by the server during the same interval. For a deterministic server, $B_n = T_{sample}/\tau_{cell\ time}$.

The following two quantities are defined in [1]:

$$s^* = \frac{2m}{\sigma^2 + 2S} \quad \text{and} \quad \tilde{c} = \frac{erfc\left(\frac{-m}{\sigma\sqrt{2}}\right) - e^{(u-m)s^*}\, erfc\left(\frac{2u-m}{\sigma\sqrt{2}}\right)}{s^*\left(\frac{\sigma\sqrt{2}}{\sqrt{\pi}} e^{-\left(\frac{u^2}{2\sigma^2}\right)} - u\, erfc\frac{u}{\sigma\sqrt{2}}\right)}$$

where $u = \sigma^2 s^*/2$, m is the mean value of Y_n, i.e., $m = E\{Y_n\} = E\{A_n\} - E\{B_n\}$, σ^2 is the variance of Y_n, i.e., $\sigma^2 = Var\{Y_n\} = Var\{A_n\}$ since B_n is deterministic, S is the auto-covariance sum of Y_n defined as the sum of the covariances of Y_n for all lags > 1:

$$S \triangleq \sum_{k=1}^{\infty} Cov(Y_n, Y_{n+k})$$

These quantities are used to compute the p^{th} percentile of the buffer occupancy as

$$y(t) = \frac{1}{s^*} ln \left(\frac{1 - \frac{p}{100}}{\tilde{c}} \right) \tag{1}$$

As mentioned in [2], computing the auto-covariance sum (S) on line is not practical. For this reason, we compute an approximate value for the p^{th} percentile using only the mean and variance of the net arrival process (substituting $S = 0$ in the above equation for s^*). For the traffic that we will be considering (e.g. Poisson), S is actually zero, and hence the computed percentile is still fairly accurate. This may not be the case for traffic with non-zero S. In such a case, using this approximation as the feedback value for control may result in the actual percentile being quite different from the computed value. Therefore, the feedback signal that we actually use, $y^*(t)$ is computed as

$$y^*(t) = \begin{cases} y(t) & \text{if } \mid y(t) - y_{act}(t) \mid < 0.2 * y_{act}(t) \\ 0.9 * y_{act}(t) + 0.1 * y(t) & \text{otherwise} \end{cases}$$

where the weighted average of the actual value, $y_{act}(t)$, and computed value, $y(t)$, is used whenever the relative error is greater than 20%.

2.2 The control algorithm

We use the multi-model minimum-bias controller developed by Liu et al. [7]. In this scheme, a model of the process to be controlled is identified online. Using the model, the output of the plant several time steps into the future is predicted for all possible control inputs. The control input that is finally chosen is the one that minimises the average error between the predicted plant outputs and the reference signal. The process is modeled by an Auto-Regressive Moving Average equation of the form

$$y_{k+1} = A_1\, y_k + \cdots + A_n\, y_{k+1-n} + B_1\, u_k + \cdots + B_m\, u_{k+1-m} + c + v_{k+1}$$

For a single-input, single-output system, y_k is the process output sampled at time k, u_k is the process input at time k and $\{v_k\}$ is white or colored noise with zero mean. Estimation of a constant offset c means that the sampled y and u values need not be mean-centered and auto-scaled as must otherwise be done. m and n indicate the order of the model.

Initially, during a startup period, the service rate, $u(t)$, is randomly varied and the output of the queueing system, $y^*(t)$, is recorded. When enough samples have been collected, the coefficients $A_1 \cdots A_n$ and $B_1 \cdots B_m$ are estimated by regression. Then, at each successive time step, the additional sample of process input and output is used to obtain a new (or updated) model recursively from the model at the previous time step. The model update step uses a modification of the well-known Recursive Least Squares algorithm.

During the identification process, several models of varying orders are identified. The control input actually sent to the process is the weighted average of the control inputs for the different models. The control input for each model is obtained by minimising the prediction errors, subject to the boundary constraints imposed on the service rate. The weights are obtained based on the quality of the predictions of the models in the past.

3 Results

3.1 Open loop tests - Poisson aggregate traffic

We assume that the traffic arriving at the queue is generated by Poisson sources. When the number of sources is large, the aggregate traffic due to bursty On/Off type sources will generally be smooth. On the other hand, the aggregate traffic due to

Poisson sources will still be Poisson and therefore, bursty. Therefore, since we are trying to control the QoS for the aggregate traffic, the assumption of Poisson arrivals is not an oversimplification, but it can be viewed as a worst-case assumption, because Poisson traffic doesn't aggregate at all.

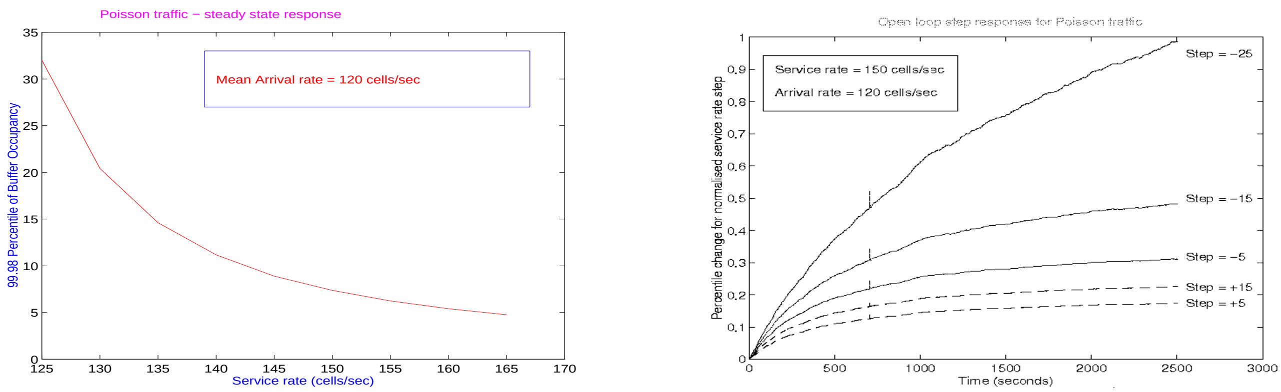

Figure 2. Steady-State Response (left) Step response (right)

Figure 2 (left) shows the steady state percentile value for a fixed arrival rate and different service rates. The relation between the percentile and the service rate can be seen to be non-linear at high utilisations. Figure 2 (right) shows the normalised (w.r.t. step size) step responses of the process, for negative and positive step changes in the service rate (of varying magnitude about a mean value of 150 cells/sec). The arrival rate is fixed at 120 cells/sec.

We can make the following observations

1. The steady state values of the buffer occupancy percentile are higher when a negative step is applied. This implies that the process gain is greater for decreases in service rate.
2. For both positive and negative step changes, the normalised steady state percentile levels depend on the magnitude of the step, showing that the gain is a function of the step size.
3. We notice that the time to reach the steady state is lower for negative step sizes, as can be seen by the higher slope. Thus the response to a negative step has a lower time constant.
4. The time constant for negative (or positive) step sizes is dependent on the magnitude of the step change.

From 1 and 2, we can conclude that the process has a gain non-linearity. From 3 and 4, we infer that the process also has non-linear response times. Thus, we can see that the queueing system with Poisson input is non-linear at high utilisations and therefore non-trivial from a control point of view.

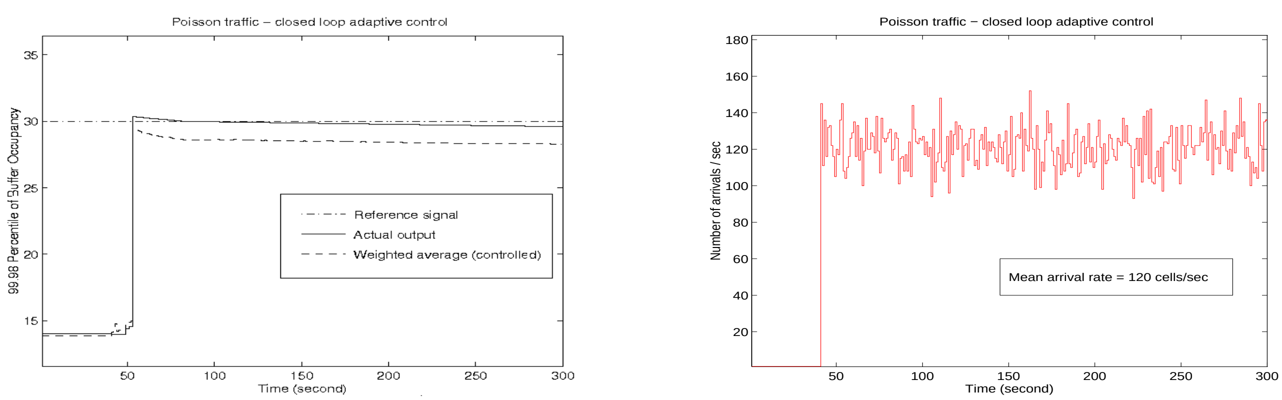

Figure 3. Closed loop control (left) Number of arrivals (right)

To demonstrate the ability of the controller to hold the QoS constant, the service rate was selected to be 150 cells/sec and the arrival rate was chosen as 120 cells/sec (utilisation = 0.8) so as to operate in the non-linear region. As a simplification, we chose the buffer size to be 1000 cells so that there would be no cell loss. The QoS objective is then to maintain the maximum cell overflow probability at 0.0002 and the maximum cell delay at 30 cells. Equivalently, this means maintaining the 99.98 percentile of the buffer occupancy at 30 cells. The controller sampling time was chosen to be 1 second. The variation of the service rate was limited to 15 % of the mean value of 150 cells/sec. The initial startup time was 40 seconds.

Figure 3 (left) shows the performance of the controller. It can be seen that the controller takes the actual buffer occupancy to the desired value very quickly, in spite of the varying and unpredictable number of arrivals in a control interval (figure 3 right).

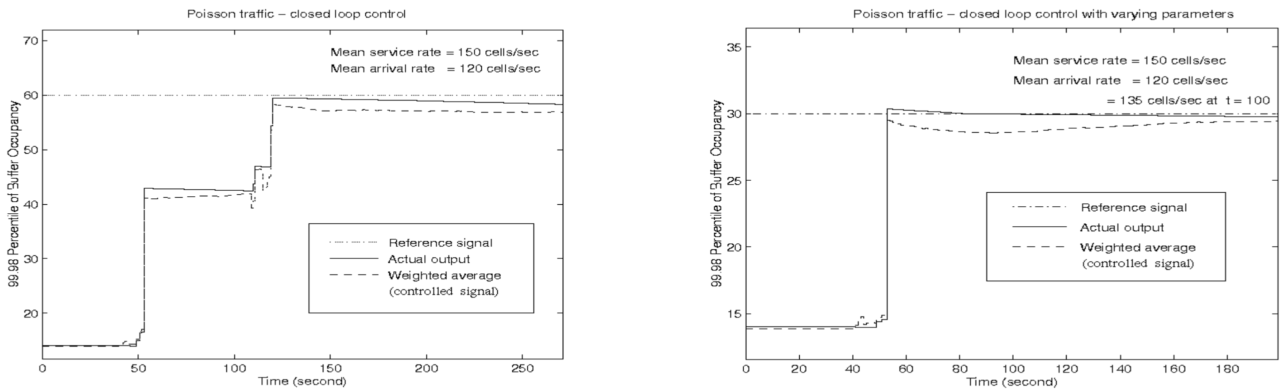

Figure 4. Closed loop control (left)   Adaptive control (right)

To demonstrate the ability of the controller to hold the QoS constant, Figure 4 (left) shows the performance of the controller when the operating point is located in a very non-linear region (utilisation = 0.9). The reference value is now set to 60 cells. The controller can be seen to maintain the QoS quite close to the desired value. Since the gain for service rate increases is small (figure 2 right), the time taken to lower the percentile can be large. Therefore, we modified the controller to avoid an overshoot. This accounts for the larger delay in satisfactory control.

To test the performance of the controller due to parameter variations, we change the average arrival rate to 135 cells/sec at time = 100 seconds, moving the utilisation to 0.9. We can see that the controller copes quite nicely with this disturbance (figure 4 right).

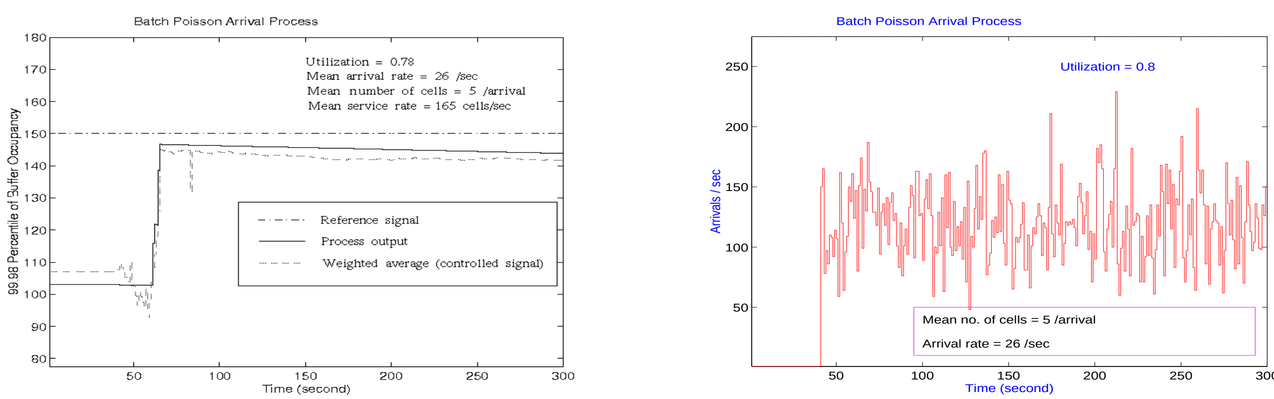

Figure 5. Closed loop control (left)   Number of arrivals (right)

Finally, we would like to demonstrate the performance of the controller on aggregate traffic with greater burstiness. As mentioned in [13], the batch-Poisson process is the limiting case for the aggregate traffic due to a number of multiplexed On/Off bursty sources. Therefore, we chose a batch-Poisson process with mean number of arrival instances = 26 /sec. At each arrival instance, the number of arrivals is determined according to an geometric distribution with mean 5 arrivals. The average rate for the aggregate traffic is then 130 cells/sec. The service rate is chosen to be 165 cells/sec (utilisation = 0.78). The reference value for the QoS is chosen as $< 150\, cells, 99.98\, percentile >$. Figure 5 (right) shows that the traffic is much more bursty than in the case of Poisson traffic (figure 3 right). The controller still does quite well in regulating the QoS (figure 5 left), the achieved QoS being in error by only about 6%.

4 Conclusions

We have described a cell scheduler for ATM networks that regulates the QoS of aggregate bursty traffic, at high utilisation. The scheduler uses adaptive feedback control to maintain the QoS at the desired value. We have demonstrated its performance in the face of traffic fluctuations and traffic parameter variations.

As a by-product, an online model of the queueing system has been identified, which can be used for performance monitoring, e.g., to detect instability and predict congestion. We have also demonstrated the effectiveness of service rate modulation over flow control in providing quick and low-overhead control. The benefit of this scheme is that it requires very little knowledge of the arriving traffic. Even if the bandwidth initially allocated by the call admission control is in error, the controller can still provide the traffic with the desired QoS.

Acknowledgments

The authors would like to thank Dr. S. L. Shah, Process Control Group, Dept. of Chemical Engineering at the University of Alberta for his valuable help. Dr. D. Liu supplied the Matlab code for the minimum-bias controller and helped in the testing and debugging of our simulations. Mr. Rohit Patwardhan provided a great deal of help along the way. Special thanks are due to Mr. S. Lakshminarayanan, for the effort spent in trying out various controllers and for the very helpful suggestions he gave us during the course of this work.

References

[1] R. Addie and M. Zukerman. An approximation for performance evaluation of stationary single server queues. *IEEE Transactions on Communications*, 42(12):3150–3160, 1994.

[2] R. Addie and M. Zukerman. A Gaussian characterization of correlated ATM multiplexed traffic and related queueing studies. *Proceedings of ICC '93*, Geneva, May 1993.

[3] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of data–handling system with multiple sources. *The Bell Technical Journal*, 61(8):1871, 1982.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking Research and Experience*, pages 3–26, October 1990.

[5] S. Golestani. A stop-and-go queueing framework for congestion management. *Proceedings of ACM SIGCOMM '90*, pages 8–18, September 1990.

[6] R. Guerin, H. Ahmadi, and M. Naghsineh. Equivalent capacity and its application to bandwidth allocation in high–speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968, 1991.

[7] D. Y. Liu, S. L. Shah, D. G. Fisher, and X. Liu. Multimodel-based minimum bias control of a benchmark paper machine process. *Canadian Journal of Chemical Engineering, to appear*, 1997.

[8] T. Murase, H. Suzuki, S. Sato, and T. Takeuchi. A call admission control scheme for ATM networks using a simple quality estimate. *IEEE Journal on Selected Areas in Communications*, 9(9):1461, 1991.

[9] S. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, February 1992.

[10] A. Pitsillides and J. Lambert. Adaptive connection admission and flow control : quality of service with high utilisation. *INFOCOM '94*, pages 1083–1091, 1994.

[11] K. M. Rege. Equivalent bandwidth and related admission criteria for ATM systems — a performance study. *International Journal of Communication Systems*, 7:181, 1994.

[12] E. D. Sykas, K. M. Vlakos, K. P. Tsoukatos, and E. N. Protonotarios. Congestion control — effective bandwidth allocation in ATM networks. *IFIP Transactions C [Communication Systems]*, C–14:65, 1993.

[13] N. Yin and M. Hluchyj. Simple models for statistically multiplexed data traffic in cell relay networks. *GLOBECOM '93*, 2:824–829, 1993.

[14] L. Zhang. Virtual clock : a new traffic control algorithm for packet switching networks. *Proceedings of ACM SIGCOMM '90*, pages 19–29, September 1990.