FEATURE

By Orhan Cagri Imer, Sonia Compans, Tamer Başar, and R. Srikant

Available Bit Rate Congestion Control in ATM Networks

Developing Explicit Rate Control Algorithms

ATM (asynchronous transfer mode) is the underlying technology enabling B-ISDN (broadband integrated services digital network). B-ISDN was introduced as the successor to narrowband ISDN after the latter fell short of meeting the high demand for bandwidth required by emerging applications such as real-time video and high definition TV (HDTV). B-ISDN envisions the transmission of fixed-size packets (cells) over digital virtual circuits at rates exceeding 150 Mb/s.

ATM is basically a packet-switching technology with 53-byte-long cells. The small cell size makes it possible to build switches that can accept and switch a large batch of cells. ATM is asynchronous in that it has no requirements that cells rigidly alternate among the various sources (i.e., cells arrive randomly from different sources).

An international nonprofit organization, the *ATM Forum*, was established several years ago, with the primary objective of promoting and extending the use of ATM products and services. The ATM standards set by the Forum define the user-network interface; that is, the way a computer owned by a private user can connect to the network and communicate through it. For that, five different services are available and are used for different types of communication [1]:

- *Constant Bit Rate (CBR):* This service category is used by connections that request a static amount of bandwidth that is continuously available during the connection lifetime. Telephone and television use this service.
- *Variable Bit Rate (VBR):* The cell rate is variable and is mainly intended for bursty sources. This service can

Imer, Başar (tbasar@decision.csl.uiuc.edu), and Srikant are with the Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana, IL 61801-2307, U.S.A. Compans is with Alcatel Business Systems, 92707 Colombes Cedex, France.

0272-1708/01/$10.00©2001IEEE

be real-time VBR (video conferencing) or nonreal-time VBR (multimedia e-mail).

- *Available Bit Rate (ABR):* In this service category, the cell rate depends on the availability of the network. It is basically designed for bursty traffic whose bandwidth range is known roughly. A rate control mechanism is specified by the ATM Forum. Examples of this service category include browsing the Web and e-mail.
- *Unspecified Bit Rate (UBR):* This category uses the leftover network capacity. UBR service does not specify traffic-related service guarantees. Background file transfer uses this service.
- *Guaranteed Frame Rate (GFR):* This service category is intended to support non-real-time applications requiring a minimum rate guarantee. It does not require adherence to a rate control protocol. An example application is frame relay interworking.

In principle, any UBR or GFR application can take advantage of the ABR flow control protocol to achieve a low cell loss ratio. Thus, the network traffic generated by various sources can be thought of as composed of CBR, VBR, and ABR connections only.

When a virtual circuit (VC) is established between a source and a destination, both the customer (source) and the carrier (switch) must agree on a contract defining the quality of service (QoS). The ATM Forum defines three QoS parameters:

- cell delay variation (CDV),
- maximum cell transfer delay (maxCTD),
- cell loss ratio (CLR).

In CBR and VBR services, a traffic contract specifying the QoS guarantees is negotiated during the virtual circuit setup phase and is maintained for the duration of the connection. The ABR service, on the other hand, does not require bounding the delay or the delay variation experienced by a given connection. Therefore, with CBR and VBR traffic, it is generally not possible for the source to decrease its rate, even when an intermediate node becomes congested, because of the QoS guarantees made at VC setup time. However, ABR sources might adjust their rates to the level of available service at times of congestion. Thus, ABR traffic can be used to control congestion in the network.

About four years ago, the ATM Forum adopted a rate-based congestion control scheme for the ABR service [1]. In this scheme, explicit rate control messages are sent from intermediate nodes to the sources using special cells called resource management (RM) cells. The goal of this congestion control mechanism is to fairly share the bandwidth left over from high-priority traffic (CBR and VBR) among the ABR sources while making sure that the links throughout the network are fully utilized.

An ATM network consists of several nodes (switches) interconnected via bidirectional links. We say that a switch is *bottlenecked* if the incoming ABR cell rate at any one of its output ports is larger than the available rate to serve the ABR cells. Clearly, at a given instant, there may be multiple bottlenecks in the network, as in Fig. 1.

Although the rate-based congestion control schemes are standardized, developing good explicit rate computation algorithms is still an open issue. As the link speeds continue to rise, the delay-bandwidth product (i.e., the product of the round-trip propagation delay and the link capacity) increases. An issue of importance that arises in this context is how to deal with action delays, which is the time from the moment control information is sent to a source, until an action is taken by it, and until subsequently that action affects the state of the switch that initiated that command. In this article, we consider a control-based mathematical model that helps us address this problem.

Our initial modeling assumption is that of a single bottleneck switch shared by several ABR sources. Although in a real network topology the sources may be interconnected in several ways, resulting in multiple bottlenecks (Fig. 1), the single bottleneck assumption admits theoretical as well as experimental justification [2] and provides a good starting point for the derivation of effective rate controllers. More precisely, we present three different congestion control algorithms for ABR control and study their performance in a simulated network environment. The mathematical model uses idealized linear queue dynamics, already introduced in [3]-[5], but the simulation model takes the saturation nonlinearities into account. Particularly, in a real network, queue length at a switch must lie between zero and the size of the buffer, and this is taken into account in the simulation model. Two of the algorithms we present formulate the congestion control problem as a stochastic team where players are users sharing the bottleneck switch. In this formulation, the approach involves a model for the available bandwidth as an autoregressive (AR) process, driven by an arbitrary, independently distributed random sequence, and minimization of an objective functional through which most of the design criteria are reflected [6]-[7]. The third algorithm we present is based on a deterministic model of the

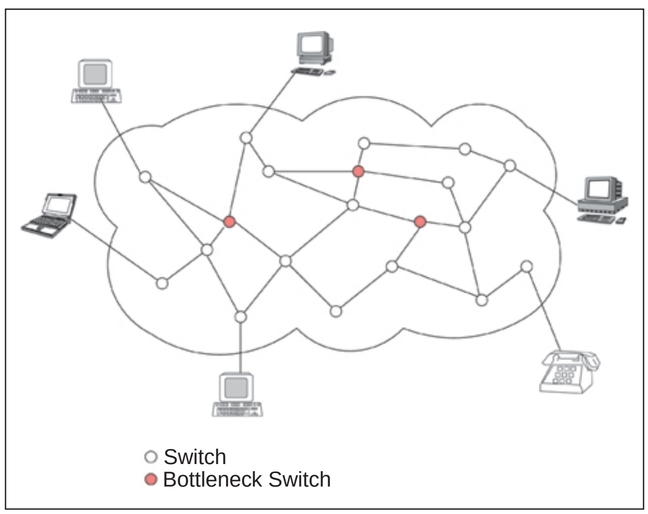

Figure 1. *A generic ATM network with multiple bottlenecks.*

available bandwidth. In this scheme, the emphasis is more on robustness against the variations in round-trip delays and estimating the number of sources sharing the bottleneck switch.

Several other types of ABR congestion control designs have been considered. We briefly summarize these here to compare and contrast them with our approach. The simplest feedback control mechanism is called *rate matching.* In rate matching, the node measures the average rate available to ABR sources at periodic intervals and simply divides a fraction of this capacity equally among the various users.

In the ABR service, the source adapts its rate to changing network conditions.

This is the basic approach used in [8], although several modifications are used in the actual implementation. The main advantage of this scheme is its simplicity, but it is difficult to control queue length optimally to avoid buffer overflows. This scheme, however, is stable (i.e., the queue length remains bounded in an appropriate stochastic sense [9]). Queue length information is not used in the basic algorithm, although [8] allows one to incorporate queue length information in an ad hoc manner. Alternatively, this problem can be viewed as a feedback control problem where queue length is used for explicit feedback. This approach is used in [10]-[11] to study this problem using classical control techniques or using a state-space approach. As in rate matching, the primary goal is not optimality, but simply queue length stability. In these approaches, the available bandwidth to ABR sources is treated as an unmodeled disturbance. Thus, the algorithms in [10]-[11] ensure stability in the presence of this disturbance, but do not address the issue of performance. In a recently published work [12], a closed-loop proportional-derivative controller is proposed, which achieves max-min fairness plus queue length stability, but the design falls short of addressing the issue of robustness against uncertainty in delays.

ABR Service

Flow Control Model for ABR

In the ABR service, the source adapts its rate to changing network conditions. As previously mentioned, information about the state of the network, such as bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special control cells called RM cells. ABR flow control occurs between a source and a destination, which are connected via bidirectional links. The forward direction is the direction from the source to the destination, and the backward direction is the direction from the destination to the source.

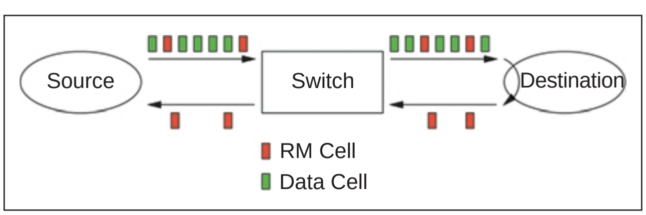

Figure 2. *ABR traffic management model.*

A source generates forward RM cells every Nrm data cells, where Nrm is generally taken to be 32. These cells travel along the same path as the data cells but are treated specially by the switches along the way (Fig. 2). The switch may:

- Directly insert feedback control information into RM cells by using the explicit rate (ER) field of RM cells.
- Provide binary feedback by marking the congestion indication (CI) or no increase (NI) bit in the RM cells.
- Indirectly inform the source about congestion by setting the explicit forward congestion indication (EFCI) bit in the data cell header, and rely on the destination to convey congestion information back to the source by marking the CI bit in the backward RM cells it generates.
- Spontaneously generate backward RM cells and ship them back to the source.

Note that Fig. 2 is only a generic representation of the control loop, as there may be more than one switch between the source and the destination. On the establishment of an ABR connection, the source specifies to the network both a maximum required bandwidth and a minimum usable bandwidth. These are designated as peak cell rate (PCR) and the minimum cell rate (MCR), respectively. The MCR may be specified as zero. The bandwidth available from the network may vary, but should not become less than MCR. Each ABR source has a current cell rate, allowed cell rate (ACR), which it must modify upon receiving feedback from the network via RM cells. The ACR always falls somewhere between MCR and PCR. When a source sends out a forward RM cell, it sets the ER field of the RM cell to the rate at which it would currently like to transmit. As the RM cell passes through the various switches on the way to the destination and back to the source, those that are congested may reduce the ER. When the source receives the RM cell back, it takes one of the following actions depending on the settings of the ER field and CI and NI bits.

- When there is no congestion (both CI and NI bits are not set), ACR can be increased (but not above PCR) by a quantity RIF×PCR, where RIF is the rate increase factor. ACR cannot be increased, however, above the explicit rate specified in the ER field.
- When a source receives a backward RM cell with CI bit set, it decreases its ACR (but not below MCR) by a quantity RDF×PCR, where RDF is the rate decrease factor. However, again the ACR of the source cannot be larger than the explicit rate specified in the ER

field. Both RIF and RDF are negotiated in the VC setup phase.

- Finally, when the source receives a backward RM cell with only the NI bit set, it sets its rate to the minimum of ACR and ER.

be equal to $(\mu - r^u)/(N - M^u)$. Here μ is the available service rate for ABR sources at a particular switch, N is the number of active sessions at that particular switch, r^u is the total rate of connections that are bottlenecked elsewhere or are limited by their PCRs, and M^u is the number of such connections.

Performance Criteria

Numerous algorithms have been and are being developed for ATM ABR congestion control. To be able to study and compare these algorithms on equal grounds, we need to set some performance measures independently of the particular algorithm under investigation. The main goal of ABR control is to provide fairness among all VCs with a minimal cell loss ratio and maximal utilization of network resources. The latter two of these objectives can be achieved by regulating the queue length at bottleneck nodes around a desirable level. Tracking such a nominal queue length (whose exact value is determined based on QoS requirements) is desirable to avoid losses due to overflow and waste of the buffer capacity due to underflow.

The goal of ABR control: provide fairness among all virtual circuits with a minimal cell loss ratio and maximal utilization of network resources.

Fairness is an issue that requires more discussion, as it may be hard to visualize what is meant by a *fair* allocation in a large network with multiple bottlenecks. The most widely accepted notion of fairness is the *max-min fairness* criterion [15]. Under this criterion, the fair share of each connection contending for a given link bandwidth should

Basic Model of an ATM Switch

Generally, an ATM switch has several input and output lines. The number of input lines is almost always the same as the number of output lines, because the links are bidirectional. Cells arrive on the input lines asynchronously, but the switching is done synchronously with the help of a master clock. Each input line is connected to a common bus, through which the incoming cells are directed to their corresponding output ports. Most of the commercial ATM switches use output queueing to prevent high cell loss rates. In output queueing each output line has a finite buffer, where the incoming cells are served on a FIFO (first in, first out) basis. Associated with each

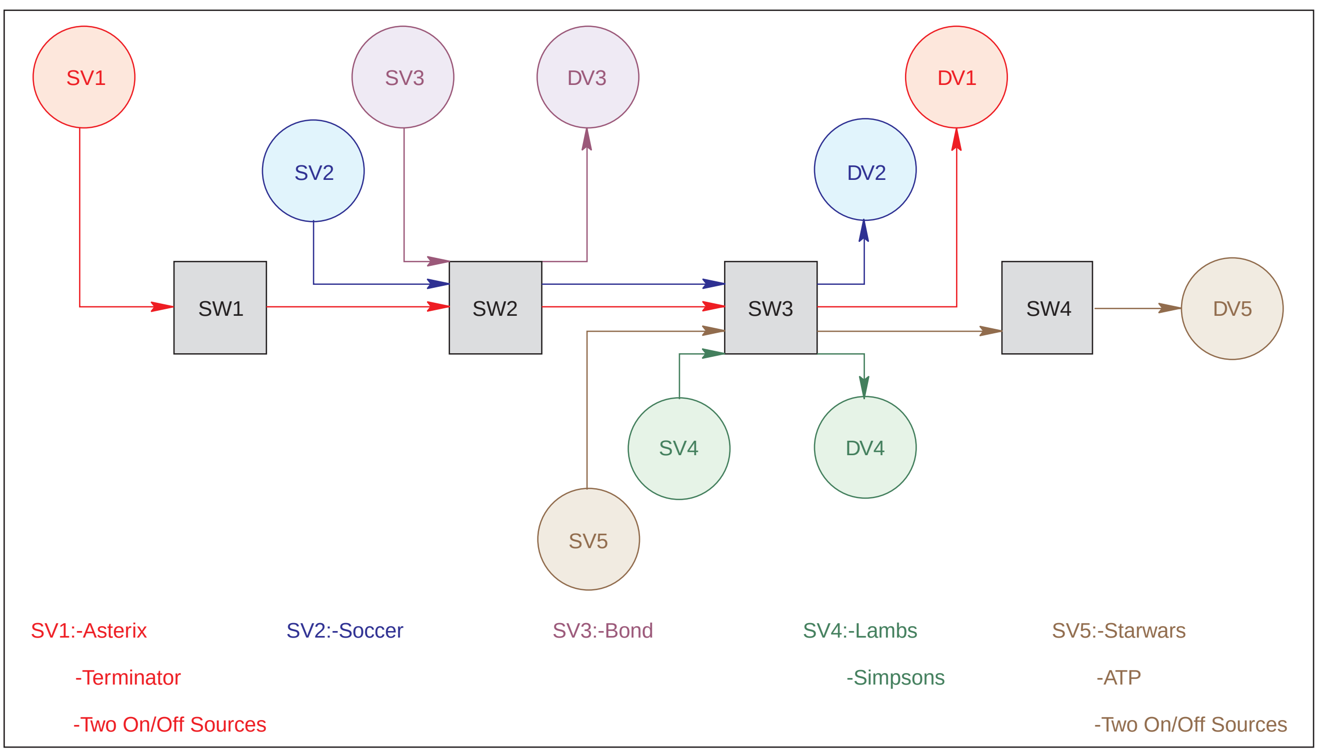

Figure 3. *Four-node configuration: VBR sources.*

output line is an ER controller that suggests a data rate for each ABR VC.

In what follows, we focus on a particular output line of an ATM switch. Other output lines can be treated in a similar manner. The model we adopt here is a discrete-time model, where a time unit corresponds to the interval over which the rate available to ABR sources is determined (that is, the interval over which measurements are made). Further, this measurement interval is assumed to be long enough for the switch to be able to process several cells-a reasonable as - sumption if the link speeds are high and packet sizes are small. This allows us to ignore the cell-level dynamics and model the ABR traffic as a fluid.

For a controlled connection, the PCR constraint never becomes active, as this would invalidate the assumption that the source is controlled.

Let r_n denote the total number of cells that arrive in one of the output buffers of an ATM switch in the interval $[n, n+1)$, and let μ_n denote the number of cells that depart from this buffer in the same time interval. Note that μ_n represents the available bandwidth (unused by higher priority traffic, particularly CBR and VBR). Denoting the queue length at time n by q_n and ignoring the boundary effects on the queue dynamics as in [16], we have the evolution

$$q_{n+1} = q_n + r_n - \mu_n. \tag{1}$$

Let there be a total number of N connections (sources) switched through the output line under study, and the number of cells that arrive from source m during time-slot $[n, n+1)$ be denoted by r_{mn}. Clearly

$$r_n = \sum_{m=1}^{N} r_{mn}.$$

In general, r_n has two components: 1) The number of cells that arrive from *uncontrolled* sources, i.e., those sources which are bottlenecked elsewhere in the network or are limited by their PCR constraints. The ER controller at this switch has no control over these sources. In other words, the ER field of an uncontrolled source is either overwritten at some other switch or is replaced by its PCR value at the source. We denote this component of r_n by r_n^u. 2) The number of cells that arrive from *controlled* sources, which are bottlenecked at this switch. The ER fields of RM cells of this type of sources are modified to achieve several traffic-related service guarantees. We assume that each source, either controlled or uncontrolled, has an MCR $= 0$. If the MCRs are positive, we can reserve the minimum cell rate for each user and make the assumption that a source will attempt to send at a rate no smaller than its reserved MCR.

Let there be a total number of M controlled sources and the number of cells that arrive from controlled source m in the interval $[n, n+1)$ be denoted by r_{mn}^c. Then, we have the following relation between r_n, r_{mm}^c, and r_n^u:

$$r_n = \sum_{m=1}^{M} r_{mn}^c + r_n^u.$$

In the analysis to follow, we assume that the switch exercises ER ABR congestion control, which amounts to setting RIF $= 1$, RDF $= 0$. Note that for a controlled connection, the PCR

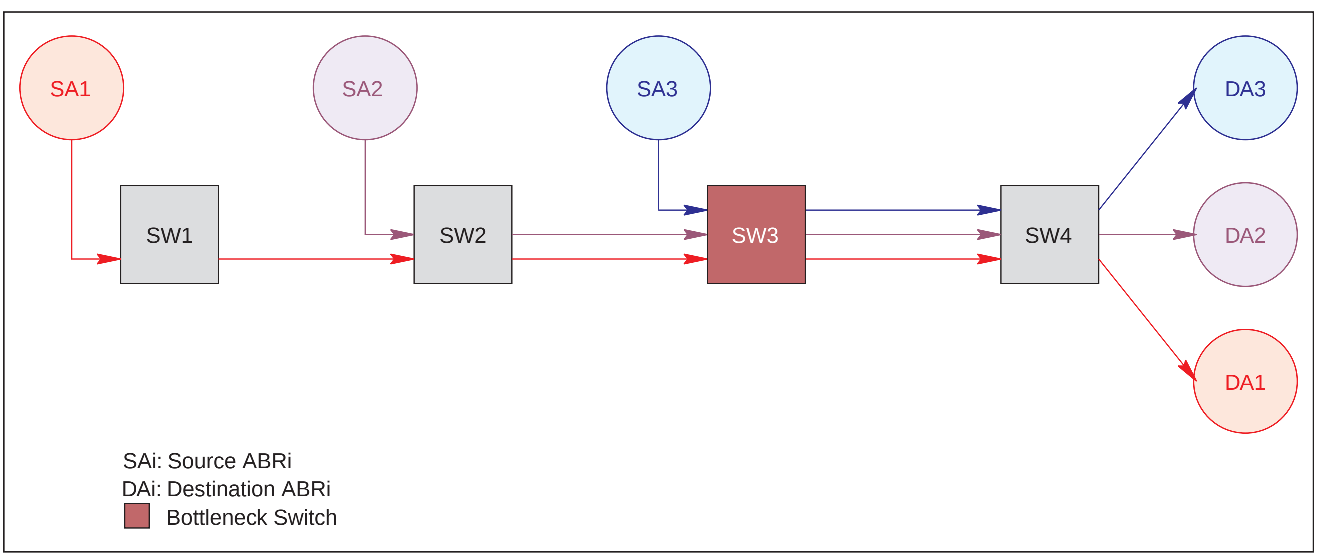

Figure 4. *Fairness configuration: ABR sources.*

Table 1. Action delays for the fairness model.

	SW1	SW2	SW3	SW4
ABR1	0	3	6	10
ABR2		0	3	6
ABR3			0	3

constraint never becomes active, as this would invalidate the assumption that the source is controlled. Also note that both r_n and μ_n represent cell rates with the unit of time taken as the sampling interval over which the measurements are made.

As indicated earlier, the decision regarding the rate at which each source should transmit is made by the switch, and the type of information accessible to the switch affects this decision. Since ATM technology relies on the high-speed switching of packets, the amount of overhead caused by the ER controller should be kept at a minimal level. At time $n+1$, when the decision on the rate of each controlled source has to be made, three pieces of information are available to the switch without any delay: queue length, q_n, available bandwidth, μ_n, and the total number of cells arrived in the interval $[n, n+1)$. Extracting these numbers does not require any elaborate measurements, as q_n can be measured by looking at how many cells in the buffer are waiting to be served, while μ_n and r_n can be determined by counting the number of cells left and arrived in the interval $[n, n+1)$, respectively. In general, to be able to fairly divide the bandwidth among all sources, the switch needs to know how many controlled sources are being served at a given time. Calculating the number of VCs requires the switch to look at the header of every single RM cell and tell what source it originated from and what destination it is routed to. Even though theoretically possible, this task brings in a computational overhead slowing down the operation of the ATM switch. Moreover, the number of VCs, N, calculated this way may not be equal to the actual number of controlled sessions, M, as some of the sources might be limited by their PCRs, bottlenecked at some other switch in the network, or just too bursty to be exercised any control over. Therefore, it is desirable to develop an ER control algorithm that relies only on the knowledge of the three easily accessible quantities described above.

As mentioned earlier, it takes time from the moment the ER decision is made by the switch until an action is taken by a source, and until subsequently that action affects the state of the node that initiated the action. Thus, the cell rate of source m at time n, r_{mn}, is actually an outcome of an action taken d_m time units earlier, where d_m represents the action delay for source m and is taken to be independent of time n. Without any loss of generality, we assume that the d_ms are ordered such that

$$0 \le d_1 \le \cdots \le d_M \le \bar{d}$$

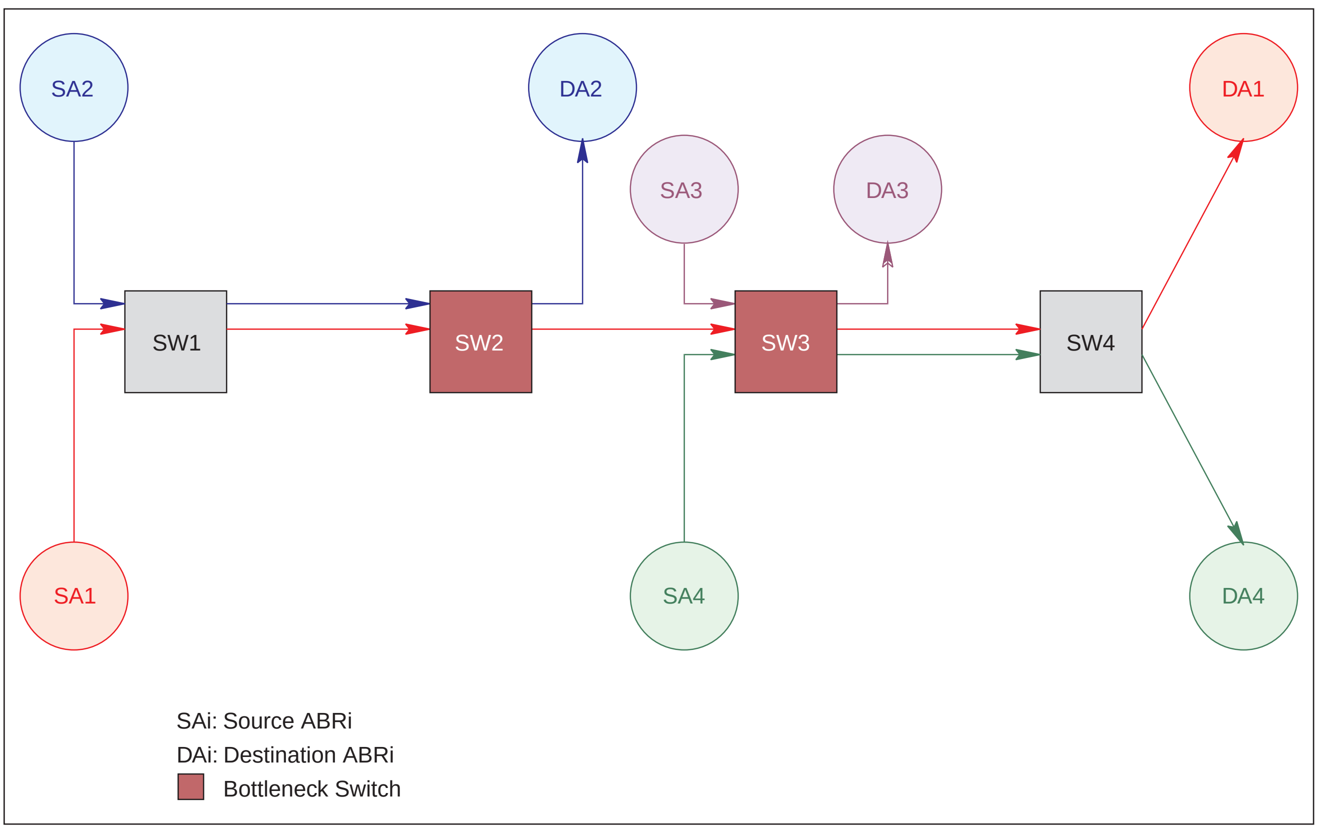

Figure 5. *Max-min fairness configuration: ABR sources.*

where $\bar{d}$ corresponds to the maximum round-trip network delay.

Congestion Control Algorithms

Controller 1: A Certainty Equivalent Controller

Basic Assumptions

The available bandwidth for ABR, μ_n, may change in an unpredictable way since the transmission rate of VBR traffic is time varying. The bandwidth available to the controlled sources, on the other hand, is subject to further uncertainty due to the variations in the uncontrolled traffic, r_n^u. These observations motivate us to consider the CBR, the VBR, and the uncontrolled ABR traffic collectively as interference, modeled by an AR process, which is stable:

$$\gamma_n := \mu_n - r_n^u = \mu - r^u + \xi_n \qquad (2)$$

$$\xi_n = \sum_{i=1}^{p} \alpha_i \xi_{n-i} + \phi_{n-1}. \qquad (3)$$

Table 2. Action delays for the max-min fairness model.

	SW1	SW2	SW3	SW4
ABR1	0	3	6	10
ABR2	0	3		
ABR3			0	
ABR4			0	3

Here μ is the constant nominal service rate, r^u is the nominal rate of uncontrolled sessions, p is the order of the AR process, $\alpha_i, i = 1,\ldots,p$ are the parameters characterizing the process, and the driving term $\{\phi_n\}_{n\geq 1}$ is a zero-mean i.i.d. Gaussian sequence with variance k^2.

Given the dynamics of the AR process, the source rates, r_{mn}^c, are determined by minimizing a cost function that quantifies the tradeoff between two partially conflicting goals: steady queue length around a nominal value and fair allocation and actual sharing of the available bandwidth. The cost function adopted for this purpose is

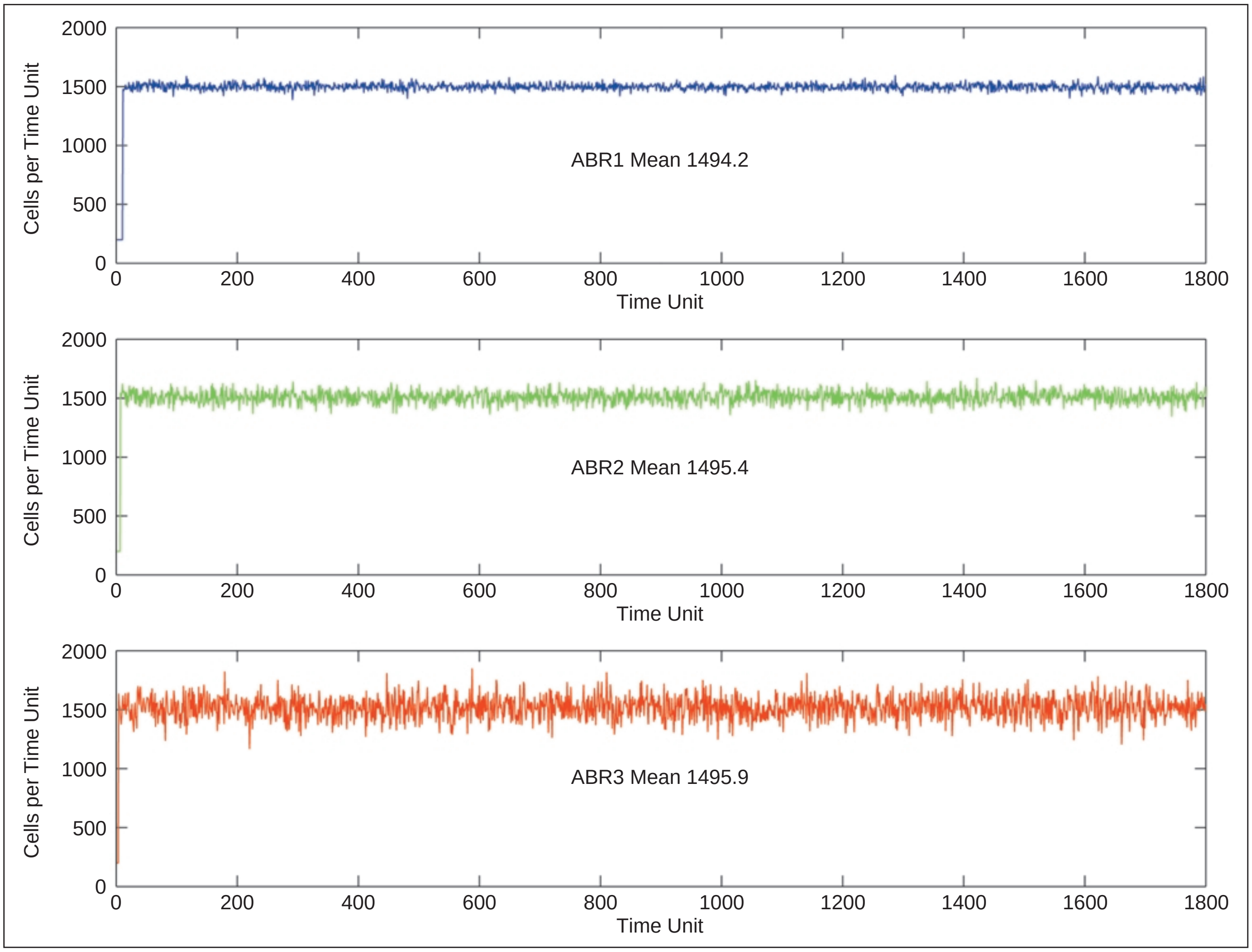

Figure 6. *Fairness model: Source rates under Controller 1.*

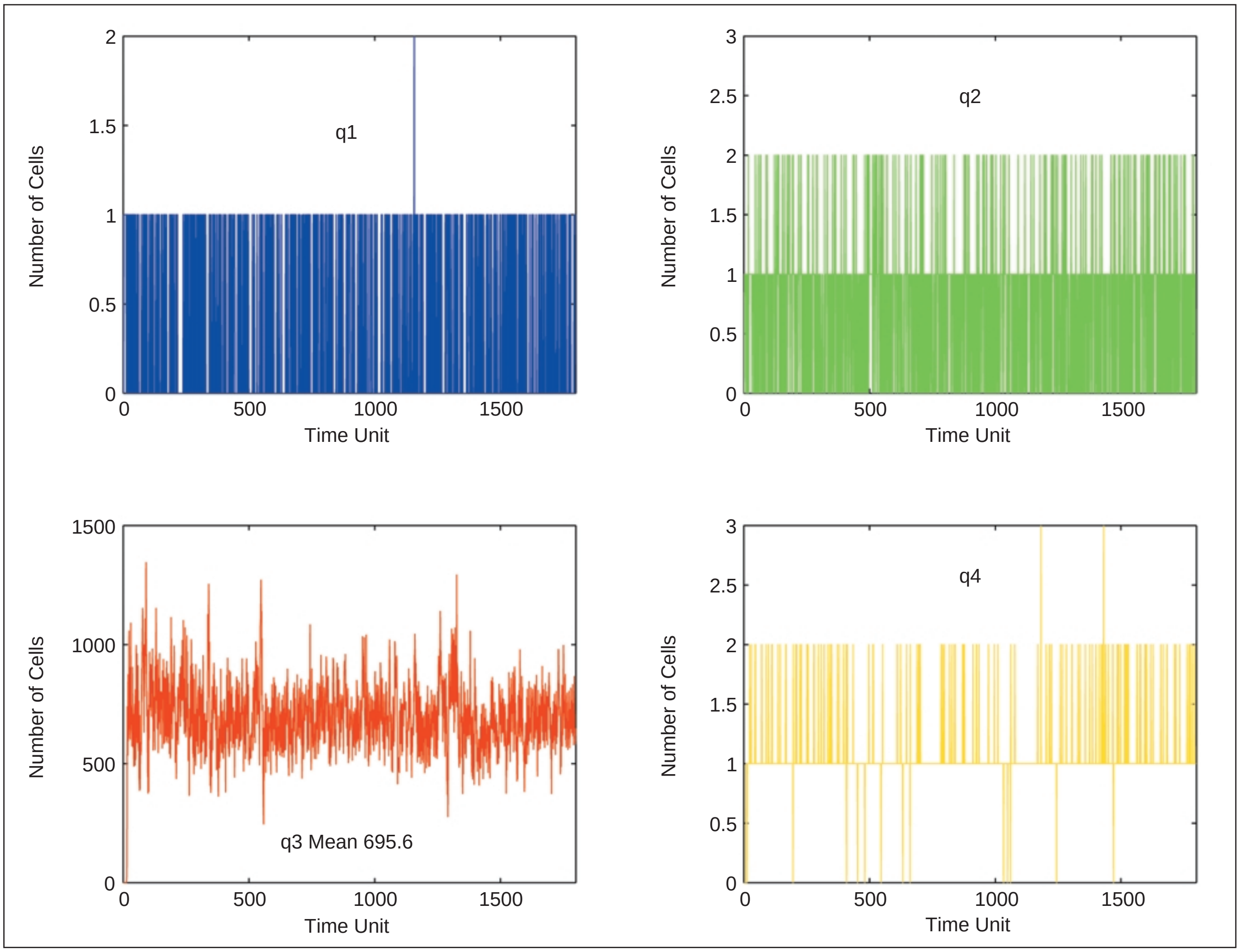

Figure 7. *Fairness model: Queue lengths under Controller 1.*

$$J = \limsup_{N\to\infty} \frac{1}{N} E\left\{\sum_{n=1}^{N}\left[(q_n - Q)^2 + \sum_{m=1}^{M}\frac{1}{k_m^2}\left(r_{mn}^c - a_m\gamma_n\right)^2\right]\right\} \quad (4)$$

where Q is the target queue length, k_ms are positive constants, $\sum_{m=1}^{M} a_m = 1$, and $E\{\cdot\}$ denotes expectation over the statistics of the AR process. Exact fair sharing implies $a_m = 1/M$, which will be the initial value for each a_m. Note that this choice of a_m requires the knowledge of the number of controlled VCs, M.

The first term in (4) represents the penalty for deviating from the desired queue length, Q, while the second term represents a penalty (for each source) for deviating from the authorized transmission rate, which we have taken to be a fraction of the available bandwidth for each source to achieve max-min fairness. The k_ms quantify the relative priority given to each source-the larger k_m is, the lower the priority.

Derivation of Controller 1

To arrive at a somewhat simpler configuration, we first introduce the shifted variables

$$x_n := q_n - Q$$
$$u_{mn} := r_{mn}^c - a_m(\mu - r^u)$$

where x_n stands for the state and u_{mn} for the control. Then the state dynamics (1)-(3) become

$$x_{n+1} = x_n + \sum_{m=1}^{M} u_{mn} - \xi_n$$
$$\xi_{n+1} = \sum_{i=1}^{p} \alpha_i\, \xi_{n+1-i} + \phi_n$$

and the cost function is

$$J = \limsup_{N\to\infty} \frac{1}{N} E\left\{\sum_{n=1}^{N}\left[(x_n)^2 + \sum_{m=1}^{M}\frac{1}{k_m^2}(u_{mn} - a_m\xi_n)^2\right]\right\}.$$

Let

$$\tilde{u}_{mn} := u_{mn} - a_m\xi_n, \quad \tilde{u}_n := (\tilde{u}_{1n}, \ldots, \tilde{u}_{Mn})'.$$

Then the cost function can be written as

$$J = \limsup_{N\to\infty} \frac{1}{N} E\left\{\sum_{n=1}^{N}\left[(x_n)^2 + \sum_{m=1}^{M}\frac{1}{k_m^2}(\tilde{u}_{mn})^2\right]\right\}.$$

Also, in terms of $\tilde{u}$, the dynamics for x become

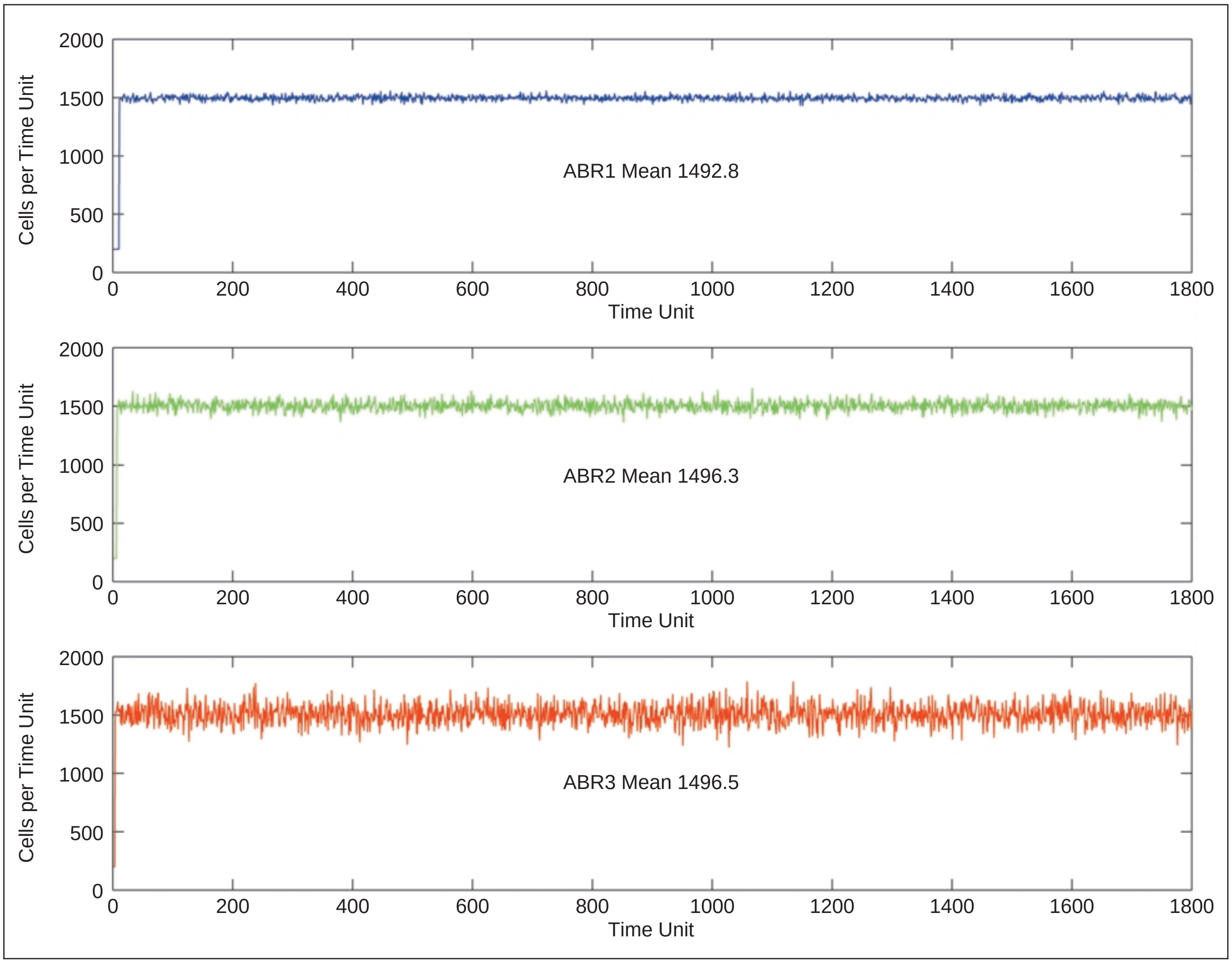

Figure 8. *Fairness model: Source rates under Controller 2.*

$$x_{n+1} = x_n + \sum_{m=1}^{M} \tilde{u}_{mn}.$$

If there was no delay (i.e., $d_1 = d_2 = \cdots = d_M = 0$), then the standard discrete-time linear regulator theory [17] would apply, yielding the unique solution

$$\tilde{u}_n = -[R + bb's]^{-1} bs\ x_n$$

where

$$b' := (1 \quad 1 \quad \ldots \quad 1)_{1\times M}$$

$$R := \operatorname{diag}\left(\frac{1}{k_1^2}, \ldots, \frac{1}{k_M^2}\right)$$

and s is the unique positive root of

$$s = 1 + s\left[1 - b'(R + bb's)^{-1} bs\right].$$

This can be solved explicitly to yield

$$s = \frac{1 + \sqrt{1 + 4\zeta^2}}{2}; \qquad \zeta^2 := \left(\sum_{m=1}^{M} k_m^2\right)^{-1}.$$

The original controller would then be (with zero delay)

$$u_n = \tilde{u}_n + a_m \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \xi_n. \tag{5}$$

Since the impact of u_{mn} on the network would be felt only after d_m time units, here we invoke certainty equivalence, where, in the solution (5) for each controller u_{mn}, $m = 1, 2, \ldots, M$, we replace the queue length and bandwidth by their estimates d_m time units later. These considerations lead to the following certainty equivalent controller, which we will refer to as Controller 1:

$$u^*_{m,n} = -p_m\, \hat{x}_{n+d_m|n} + a_m\, \hat{\xi}_{n+d_m|n}, \qquad m = 1, \ldots, M. \tag{6}$$

Here p_m is the mth component of $[R + bb's]^{-1} bs$ and $\hat{x}_{n+d_m|n}$, $\hat{\xi}_{n+d_m|n}$ are the predicted values of x_{n+d_m} and ξ_{n+d_m}, respectively, based on the what is known at time n and given that all other controllers are also in the form (6). These predictors are generated by

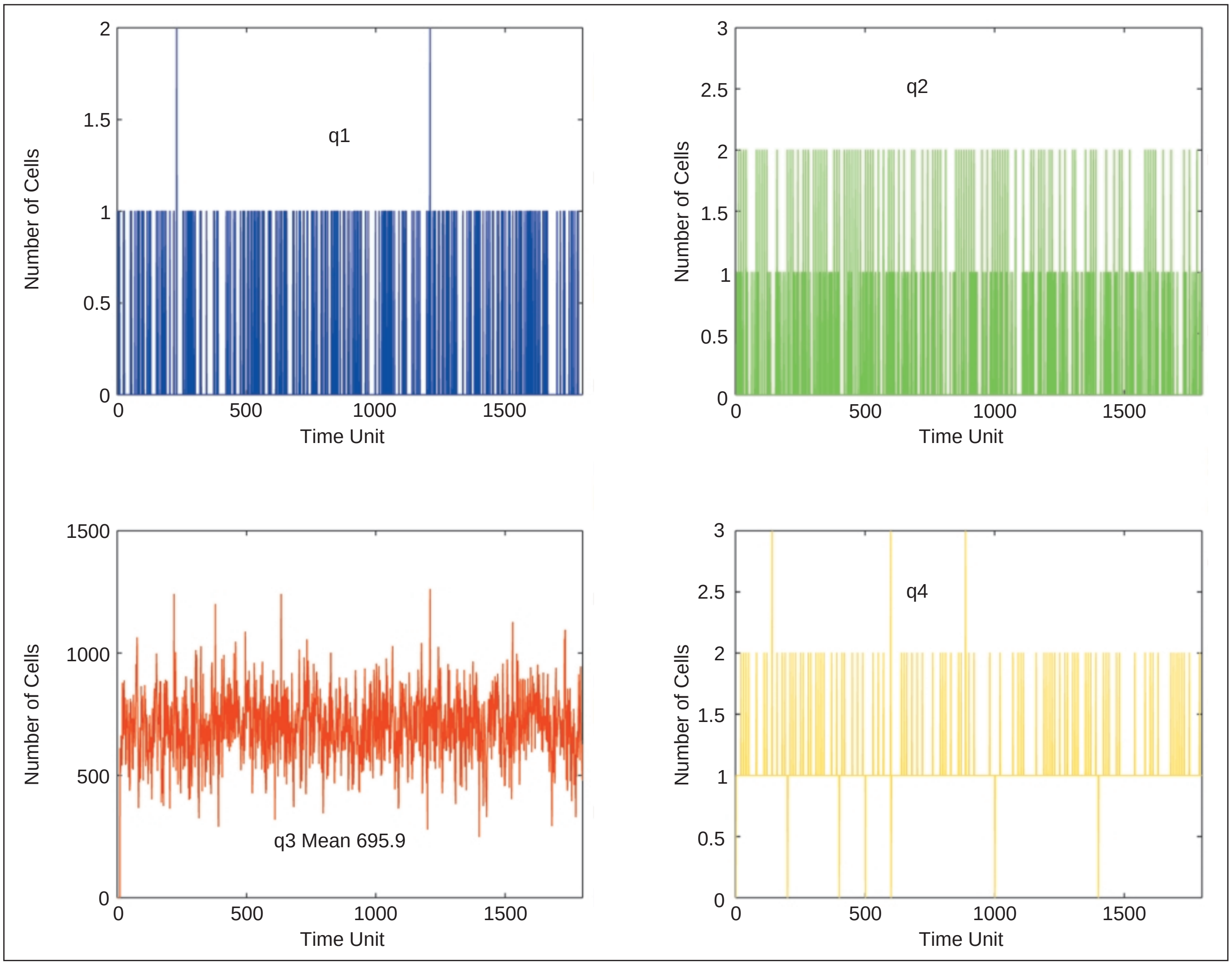

Figure 9. *Fairness model: Queue lengths under Controller 2.*

$$\hat{x}_{n+j|n} = \hat{x}_{n+j-1|n} + \sum_{m=1}^{M} \hat{u}_{m,n+j-1-d_m|n} - \hat{\xi}_{n+j-1|n}, \quad j \geq 1;$$
$$\hat{x}_{n|n} = x_n;$$
$$\hat{\xi}_{n+j|n} = \sum_{i=1}^{p} \alpha_i \hat{\xi}_{n+j-i|n}, \quad j \geq 1; \quad \hat{\xi}_{n-k|n} = \xi_{n-k}, \quad k \geq 0;$$

and

$$\hat{u}_{m,n+j-1-d_m|n} := \begin{cases} -p_m \hat{x}_{n+j-1|n} + a_m \hat{\xi}_{n+j-1|n} & \text{if } j \geq d_m + 1 \\ u_{m,n+j-1-d_m} & \text{if } j < d_m + 1. \end{cases}$$

These are the recursive equations generating the predictors for the queue length and rate information at a future time, where the future time is the current time plus the action delay for the corresponding source. For example, $\hat{\xi}_{n+j|n}$ denotes the predicted value at time n of the value of ξ at some future time $n+j$, based on the information available at time n. A similar interpretation holds for $\hat{x}_{n+j|n}$.

The above algorithm is relatively easy to implement. The estimator algorithms are simple scalar operations, and the scalar solution of the Riccati equation, s, has already been obtained explicitly. In summary, an easily implementable version of Controller 1 is given below in the form of pseudocode:

Pseudocode for the node's computation at time *n* using Controller 1

for $j = 1$ to d_M do
  for $m = 1$ to M do
    if $(n + j - 1 - d_m \geq n)$
      $\hat{u}_{m,n+j-1-d_m|n} = -p_m \hat{x}_{n+j-1} + a_m \hat{\xi}_{n+j-1}$
    endif
  endfor
  $\hat{x}_{n+j} = \hat{x}_{n+j-1} + \sum_{m=1}^{M} \hat{u}_{m,n+j-1+d_m|n} - \hat{\xi}_{n+j-1}$
endfor
$u_{mn} = -p_m \hat{x}_{n+d_m} + a_m \hat{\xi}_{n+d_m}$

Controller 2: Optimal Controller

Recall that in the derivation of Controller 1, we first assumed zero delays, solved the simplified linear quadratic regulator (LQR) problem, and then incorporated the delays into controllers through estimators using the certainty-equivalence principle. An alternative approach, which leads to the opti-

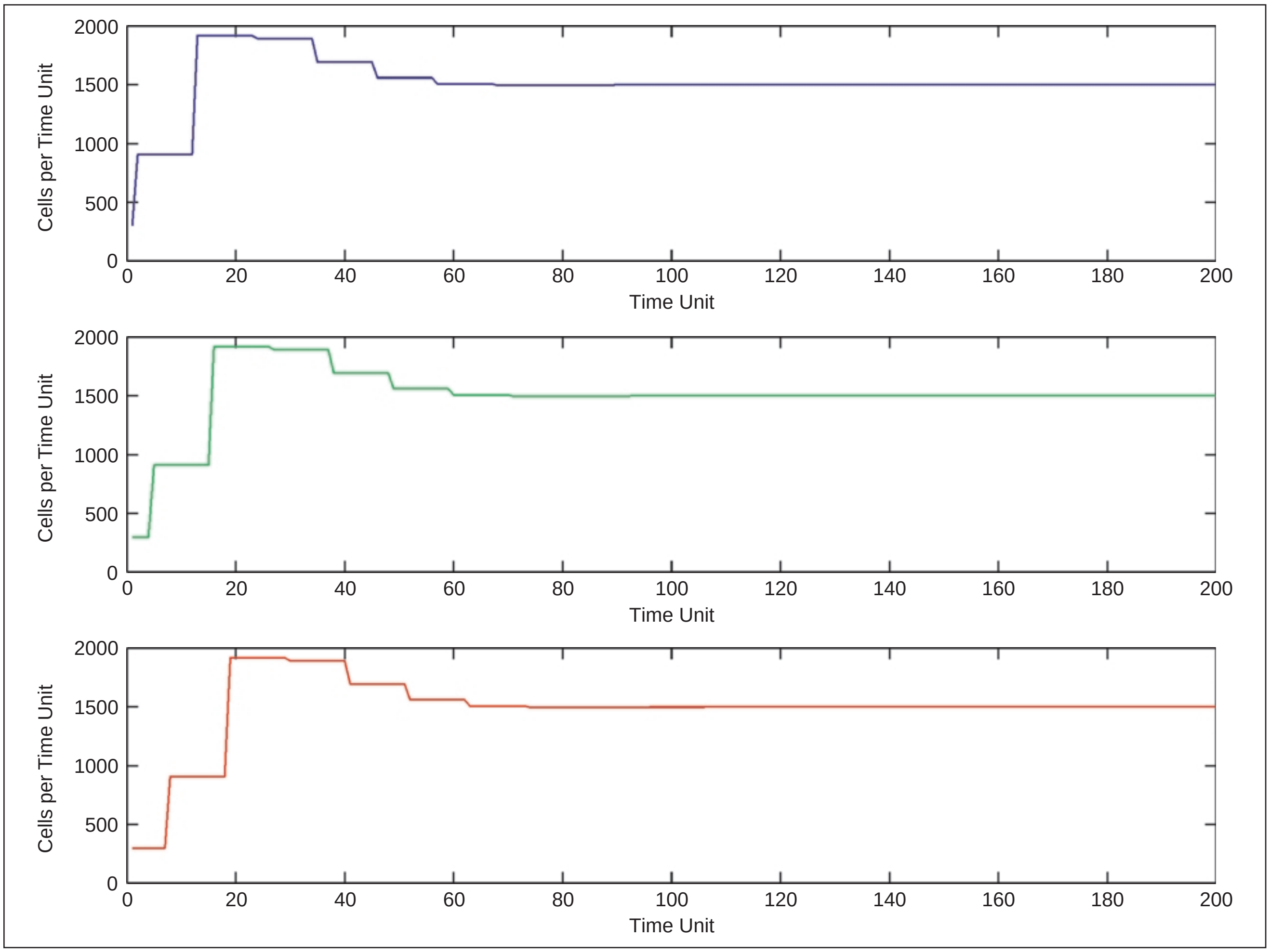

Figure 10. *Fairness model: Source rates under Controller 3.*

mal solution of the problem, is to augment the state space in an appropriate way. Basically, one must introduce new state variables for those sources that have nonzero delays. Since this is a tedious process with many algebraic details, we do not include the derivation of the optimal controller here, but instead refer the interested reader to [18]. The optimal solution is characterized in terms of the solution of a discrete-time algebraic Riccati equation (DARE), whose dimension is determined by the magnitude of the largest delay and the order of the AR process describing the available capacity.

Controller 3: A Robust Adaptive Controller Under Uncertainty

Basic Assumptions

In our earlier paper [6], by considering a one-node example with perfect delay information and instantaneous feedback, we showed that delay is an important factor in any design of rate flow controllers and hence must be explicitly taken into account in any realistic model of high-speed networks, as we have done above. In a realistic network, however, there are several deviations from this ideal model.

Both Controllers 1 and 2 assume complete knowledge of delays in the network. Although the end-to-end round-trip delay may be known to the switch as part of the fixed round-trip time (FRTT) computation performed at VC setup time, we can still have small errors in delay estimates. One source of error is the assumption that the delay is an integer multiple of the time unit (measurement interval), which may not be true. Further, there is variability in the delays due to queues in the virtual circuits. Finally, RM cells are generated only every 32 data cells and hence feedback is not instantaneous.

In the derivations of Controllers 1 and 2, we also assumed that the switch has access to the information about the number of controlled sources, M. Recall that we need this number to determine the fair share of each source. As mentioned earlier, in a realistic environment with bursty sources, determination of M may not be an easy task. Hence, the switch must have a method of estimating this parameter.

Motivated by these observations, we want to develop a controller that is robust to uncertainty in delays, and at the same time adaptive to the number of sources. Thus, in deriving what we call Controller 3, we only require the knowledge

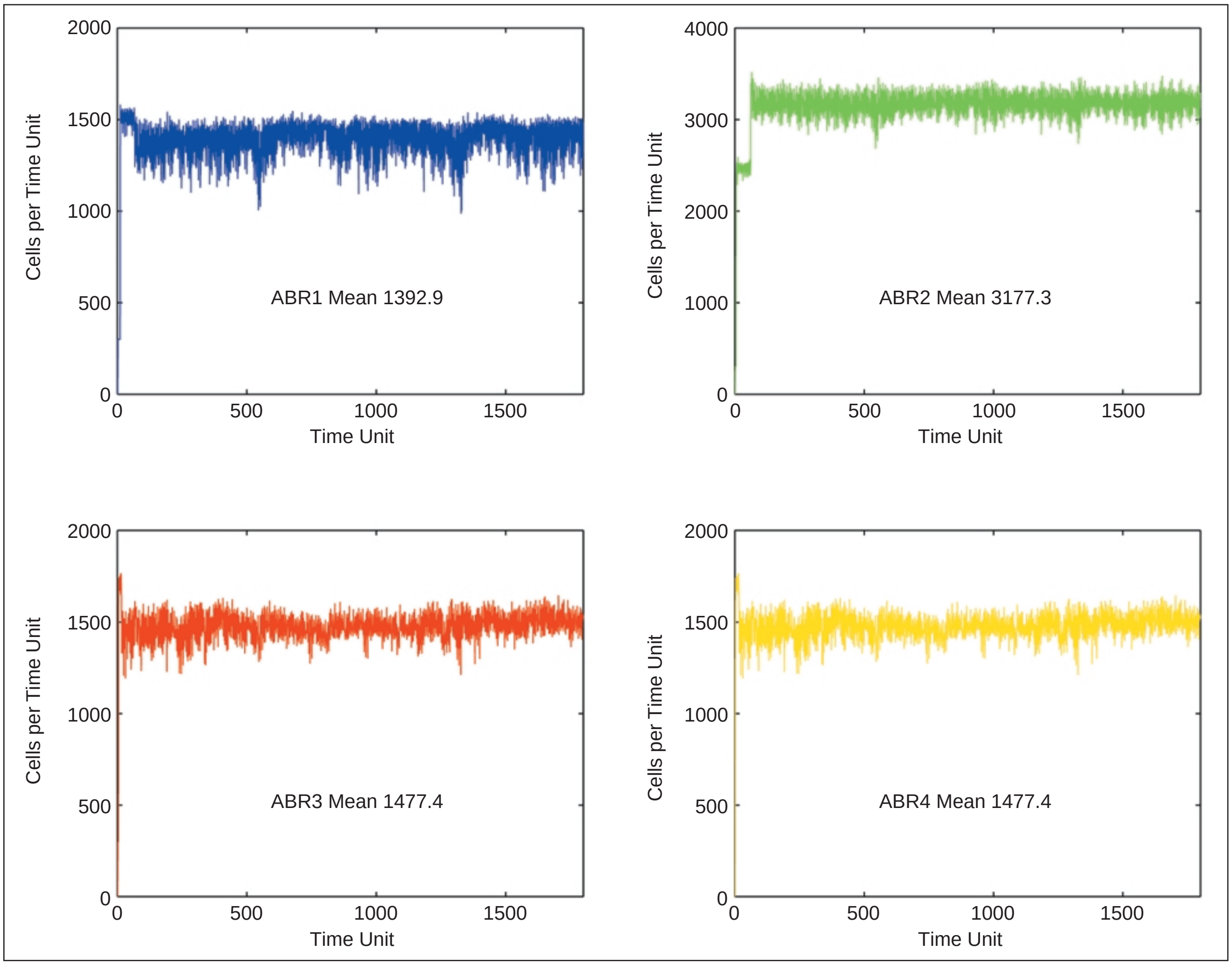

Figure 11. *Max-min fairness model: Source rates under Controller 1.*

of $\bar{d}$ and $\bar{M}$, which denote the upper bounds on the round-trip delay and the number of controlled sessions, respectively.

Although the available bandwidth for ABR, μ_n, and the rate of uncontrolled sessions, r_n^u, may change with time, as in the previous section, we assume here that both μ_n and r_n^u are constants denoted by μ and r^u, respectively. These assumptions are justified if μ_n and r_n^u vary slowly compared to the time constant of the closed-loop system. With these simplifying assumptions, the state equation (1) becomes

$$q_{n+1} = q_n + \sum_{m=1}^{M} r_{mn}^c + r^u - \mu.$$

Since the rate of the controlled source m at time n is actually an outcome of an action taken by the switch d_m time units earlier, we have

$$r_{mn}^c = c_{n-d_m}$$

where c_n denotes the command issued by the switch at time n.

Derivation of Controller 3

As the actual number of controlled sources, M, is not known to the switch, we start our analysis by proposing a method to estimate this figure. Let us first consider the case when $r^u = 0$ (i.e., there are no uncontrolled sessions). Our method relies on a simple observation: If the switch sends out the same command, say c, for $(\bar{d}+1)$ time units, at the end of the $(\bar{d}+1)$st step, all of the controlled sources in the system will be transmitting at rate c. Hence, at the end of the $(\bar{d}+1)$st time step, if we divide the incoming cell rate, r_n, by the assigned rate c, we obtain the actual number of controlled sources, M, at the switch. To have a running estimate of this figure, one can construct an estimator, $\hat{M}_n$, and update it every $(\bar{d}+1)$ time units using this scheme. We note that this algorithm converges to the exact value of M in only a finite number of steps. Having determined M in this manner, one can set the command c at the next time slot to be equal to μ / M and achieve fairness, which in this particular case also corresponds to max-min fairness, as we took $r^u = 0$.

Now if the rate of the uncontrolled connections, r^u, is not zero, the above scheme fails to converge to the actual number

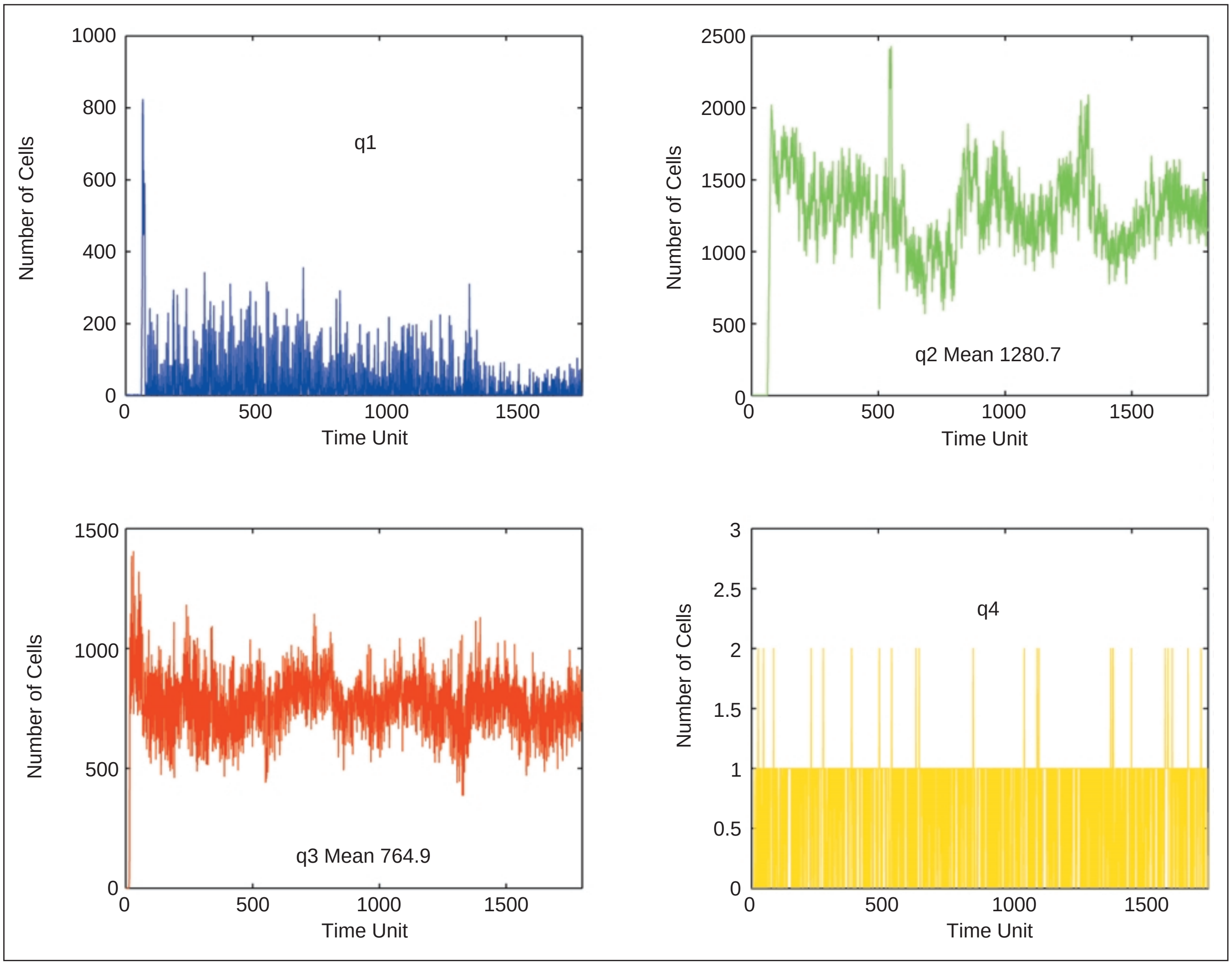

Figure 12. *Max-min fairness model: Queue lengths under Controller 1.*

of bottlenecked connections, *M*. It still does converge, however, but to a different value. In fact, as we will show shortly, in calculating the share of each controlled connection, if one uses the number to which the algorithm converges, then the resulting bandwidth allocation is max-min fair.

Before proceeding further, we note that the choice of the rate *c* above is completely arbitrary, because it does not play any role in the estimation process. Thus, we are free to pick any rate larger than zero for the algorithm to work. Mathematically speaking, letting $\hat{M}_n$ denote the estimate of *M* at time *n*, we propose the following estimator:

$$c_{n(\bar{d}+1)} = c_{n(\bar{d}+1)+1} = \cdots = c_{(n+1)(\bar{d}+1)-1},$$
$$\hat{M}_{n(\bar{d}+1)} = \hat{M}_{n(\bar{d}+1)+1} = \cdots = \hat{M}_{(n+1)(\bar{d}+1)-1} = \frac{r_{n(\bar{d}+1)}}{c_{(n-1)(\bar{d}+1)}}$$

where the first equation sets c_n to the same value for $\bar{d}+1$ steps, while the second equation is used to estimate *M* every $\bar{d}+1$ time units. For ease of notation, let us introduce the following subsequences:

$$q_n^s := q_{n(\bar{d}+1)},\quad c_n^s := c_{n(\bar{d}+1)},\quad r_n^s := r_{n(\bar{d}+1)},\quad \hat{M}_n^s := \frac{r_n^s}{c_{n-1}^s} = M + \frac{r^u}{c_{n-1}^s} \tag{7}$$

with $\hat{M}_0^s = \bar{M}$. To complete the design of Controller 3, we need to specify how c_n^s should be selected to achieve the dual goal of max-min fairness and queue length stability. We propose the following design for c_n^s:

$$c_n^s = \max\left\{\frac{\mu}{\hat{M}_n^s} - \beta(q_n^s - Q), 0\right\} \tag{8}$$

where $\beta > 0$ is the gain to be selected to ensure stability, and the max function is introduced to ensure that the switch asks the source to transmit at a positive rate in excess of MCR =0, as required by the QoS specifications. In (8), the term $-\beta(q_n^s - Q)$ is introduced to drive the queue length, q_n, to the desired set point, *Q*, by providing negative feedback in the closed-loop system dynamics.

Note that if q_n converges to *Q*, the command of the node for controlled sources, c_n^s, converges to the solution of

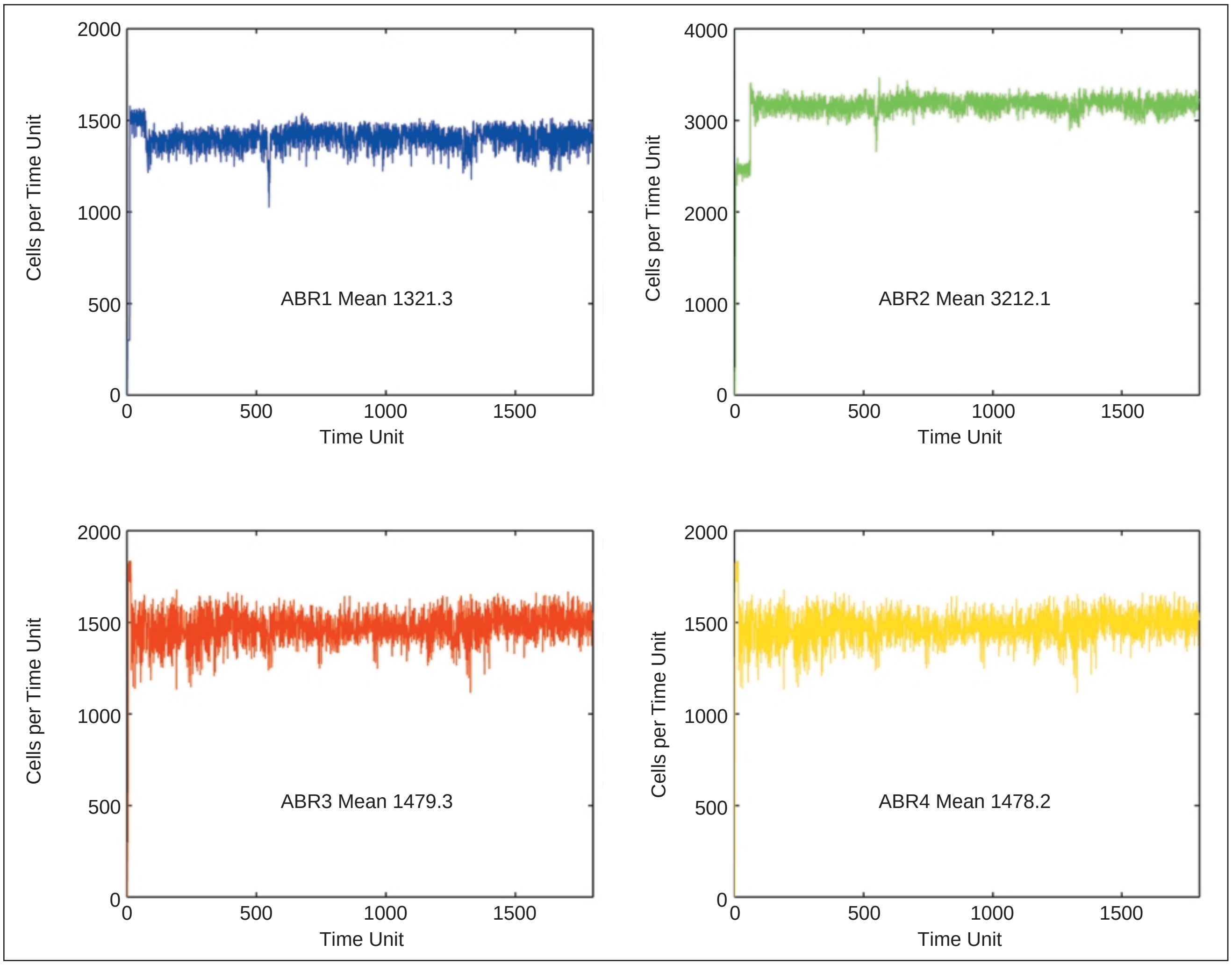

Figure 13. *Max-min fairness model: Source rates under Controller 2.*

$$c_\infty^s = \max\left\{\frac{\mu}{M + \frac{r^u}{c_\infty^s}},\ 0\right\},$$

which is given by

$$c_\infty^s = \max\left\{\frac{\mu - r^u}{M},\ 0\right\} = \frac{\mu - r^u}{M}$$

where the second equality follows from the obvious constraint $\mu - r^u \geq 0$. Hence, if we can show that the closed-loop queue dynamics, q_n, converge to Q, then the rate of controlled sessions converges to the max-min fair share of the available bandwidth. In fact, it can be shown that if the controller gain, β, is picked such that

$$0 < \beta < \frac{1}{\left(\bar{d} + 1\right)\bar{M}}\left(1 - \frac{r^u}{\mu}\right),$$

then the robust control policy, given by (7)-(8), achieves queue length control plus fair share of the available bandwidth [20]. The proof of this fact is rather tedious and is hence omitted in this article. Note that, as with Controllers 1 and 2, the algorithm we propose here does not suffer from computational complexity, because there is a single design parameter, namely β, to be tuned, and to determine the ER the switch has to perform only two divisions, one multiplication, and two additions per output line, every $(\bar{d} + 1)$ time units. Moreover, the information the switch needs to perform these calculations, $\{r_n^s, q_n^s, \mu\}$, is locally available, and just a few memory elements per output port are required, as there are only four numbers, $\{\bar{d}, \bar{M}, Q, \beta\}$, to be stored.

Simulations

Simulation Model

In simulations, we consider a four-node high-speed network where the link speed is the speed of light, the service rate offered by every link is 1 Gb/s, and the distance between adjacent nodes is constant and equal to 1000 Km. We take the time unit to be the time required to serve 5000 cells (5000 ×

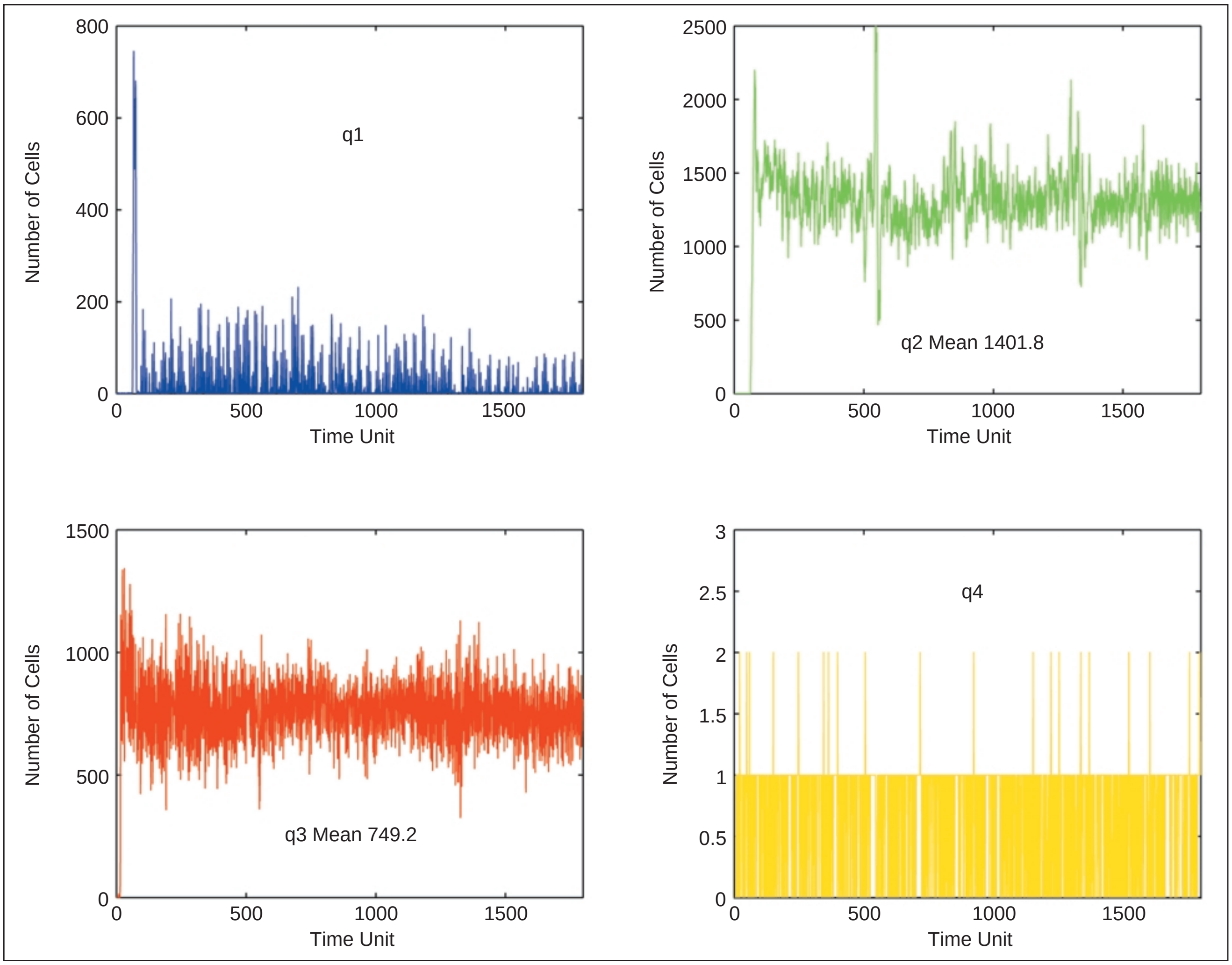

Figure 14. *Max-min fairness model: Queue lengths under Controller 2.*

53×8 bits). Then, the propagation delay for a cell (i.e., the time required for a cell to traverse a link) is

$$t = \frac{1\times10^6}{3\times10^8} \times \frac{10^9}{53\times8} \times \frac{1}{5000} \approx 1.6 \text{ time units},$$

where we have again taken the speed of propagation to equal the speed of light. Accordingly, we consider here the four-node network depicted in Fig. 3. In the figure, *SVi* represents a set of VBR sources as detailed under the figure and *DVi* is the destination of *SVi*. Moreover, the VBR sources considered are of two types: video traces obtained from [22]-[24] and simulated ON-OFF sources. The latter are bursty sources simulated by us, which alternate between ON and OFF states according to a Markov chain, and when in the ON state, cells arrive at a constant rate.

Superimposed on the model in Fig. 3, which depicts only VBR sources, we consider two configurations for the ABR sources:

- *Parking lot model*: This model will allow us to evaluate the fairness of our algorithms when there is only one bottleneck node. This network has a "parking lot" configuration [25] where three ABR sources with different delays (see Fig. 4) are bottlenecked at the third node. The one-way propagation delay from one switch to another is 1.6 time units. Since the delays used in the control algorithms are integers, we will take the nearest integers to the actual delays. For example, the actual action delay between source ABR1 and the second switch is $2 \times 1.6 = 3.2$ time units. The corresponding delay used in the mathematical model is then three. Table 1 specifies the action delays (the ones used by the control algorithms) between the sources and the switches. Recall that the action delay characterizing a source with respect to a node is defined as the time taken from the moment the node sends control information to a source (via a backward RM cell) until that source takes the corresponding action, and until subsequently that action affects the state of the node. For the adaptive control algorithm,

we take $\bar{d} = 10$, which is the largest possible network delay in this configuration. Note that to implement the adaptive robust controller, we do not need the delay information given in Table 1.
- *Max-min model*: The parking lot model has only one bottleneck node. This does not allow us to fully evaluate the max-min fairness properties of our algorithms. Hence we consider the model depicted in Fig. 5. Three of the ABR sources are bottlenecked at the third node. The remaining ABR source (SA2) will then use the remaining capacity at the second switch (according to the max-min criterion [15]), which then becomes bottlenecked. The network eventually ends up with two bottleneck nodes. Table 2 specifies the action delays used in the mathematical model.

For the simulation examples, we used the following parameter values:
- Time unit: Time required to serve 5000 cells
- Target queue length: $Q = 700$
- ICR = 300, MCR = 0, PCR = 4500
- RIF = 1, RDF = 0
- Weights: $k_m = 1\ \forall m$ (Controllers 1 and 2)
- Maximum number of sessions: $\bar{M} = 5$ (Controller 3)
- Maximum network delay: $\bar{d} = 10$ (Controller 3)
- Controller gain: $\beta = 1/(1.2 \times \bar{M} \times (\bar{d}+1)) \simeq 0.0152$ (Controller 3).

For Controllers 1 and 2, the nominal service rate μ and the AR process parameters α_i are estimated online using the Yule-Walker algorithm [26], assuming that the order of the AR process is eight. This is discussed in the next subsection. For Controller 3, we take μ to be equal to 4500 cells per time unit. The propagation delay from one node to the next is 1.6 time units. Note that the actual delay is variable since the cells go through node buffers, which leads to additional queueing delays.

Following the current ATM standards, the feedback mechanism has been implemented using RM cells that are generated by the sources every 32 data cells. Even though the mathematical model for the controllers can only use integer values for delay, the actual one-way propagation delay for the RM cells is 1.6 in the simulations. ATM standards allow for measurement of propagation delays in the signaling protocol. Queueing delay, however, remains variable and unknown. Thus, one of the goals of our simulation is to show that the impact of these two types of approximations (to the actual delays) on the performance of Controllers 1 and 2 is negligible.

The AR Process

Two of our algorithms are based on modeling the available capacity (i.e., total capacity minus the capacity used by the VBR and uncontrolled ABR sources) as an AR process (see (1)-(2)). The order p, the parameters α_i, and the variance of the noise have to be determined. We wish to emphasize that the variance of the noise is determined as part of the estimation of the AR parameters but is never actually used in the control algorithms, as it is not needed.

Tuning the parameters of the AR process and finding the best possible value of p is a challenging task in general. Several methods exist to calculate the parameters, α_i, once p is fixed. We use the Yule-Walker algorithm [26] to determine p and α_is from the data that is observed online.

Let T be the time interval over which we attempt to fit an

The ABR service plays a central role in regulating the network traffic.

AR model to the available capacity. Then one criterion to determine the best order for an AR process is the final prediction error (FPE), defined by

$$\text{FPE} = \hat{\sigma}^2 \frac{T+p}{T-p}$$

where $\hat{\sigma}$ is the estimate of the variance of the noise caused by the cross traffic. The order p of the process that minimizes the FPE is the best order [26]. The time interval in our simulations was $T = 200$, and using this, we determined that $p = 8$ gives good results. For the sake of brevity, we do not include the details here.

Simulation Results

In simulations, all rates are expressed in cells per time unit.

Fairness

First, by considering the "parking lot" configuration (Fig. 4), we study the fairness of our designs when there is only one bottleneck node (SW3). In a single-bottleneck node scenario, fairness is the capability of the algorithm to fairly distribute its available capacity among the various sources despite the presence of different delays.
- *Controllers 1 and 2*: When VBR sources are present, the mean available bandwidth at SW3 is around 4500 cells per time unit. In addition, recall that sources ABR1, ABR2, and ABR3 have delays of six, three, and zero, respectively. The simulations for both controllers show that the capacity is fairly shared among the three sources (Figs. 6 and 8) with a mean rate of around 1495. Although the sources have different delays, the design manages to provide fairness. The queue at the third node is regulated around 700, as expected (Figs. 7 and 9).
- *Controller 3*: When there are no VBR sources, the available service rate at SW3 is exactly 4500 cells per time unit. As can be seen from Fig. 10, our algorithm

achieves fair share. Also the queue at SW3 converges to the desired value of 700. In addition, the algorithm finds the actual number of sources at SW3 in $\bar{d}+1=11$ time units, as expected. The rate of convergence of source rates as well as the queue length depend on the controller gain β. A smaller value of β results in a smaller overshoot but a larger settling time.

Max-Min Fairness

The configuration under consideration is depicted in Fig. 5. The main bottleneck node is the third one, where the capacity should be equally distributed among the sources ABR1, ABR3, and ABR4. Then, since ABR1 does not use its fair share at node 2, ABR2 should use the remaining capacity and increase its cell rate to make up for the difference.

- *Controllers 1 and 2*: With VBR sources present, the mean available capacities are around 4870, 4720, 4480, and 4860 cells per time unit for switches 1, 2, 3, and 4, respectively. If the capacity at SW3 is equally distributed among the sources ABR1, ABR3, and ABR4, each one of these sources should transmit at around 1500 cells per time unit. Then, ABR2 should use the unused capacity at SW2 and increase its cell rate to around 3200. Nevertheless, our basic algorithm sets the weights to the exact fair share ($1/M$ for M sources), and the drop observed in the queue length at switch 2 is not substantial enough to increase the rate of ABR2. Therefore, max-min fairness is not achieved with the original controllers. An adaptive weight algorithm needs to be implemented to adapt the weights to a max-min fairness configuration. This problem has been addressed in [27], where the authors determine the actual activity of each source and deduce from that a max-min fair share. We adopt a similar idea to choose the weights a_m, as described below. Consider a switch with M sources going through it. We first evaluate the mean cell rate of each source m, say $meanCCR(m)$, by using the CCR (current cell rate) field in the RM cells. If a source uses less than $1/M$ times the capacity of the switch, then its a_m is reduced to the fraction that it actually uses. Such sources are called *underloading* sources. As a result, the remaining capacity is fairly allocated to the rest of the sources.

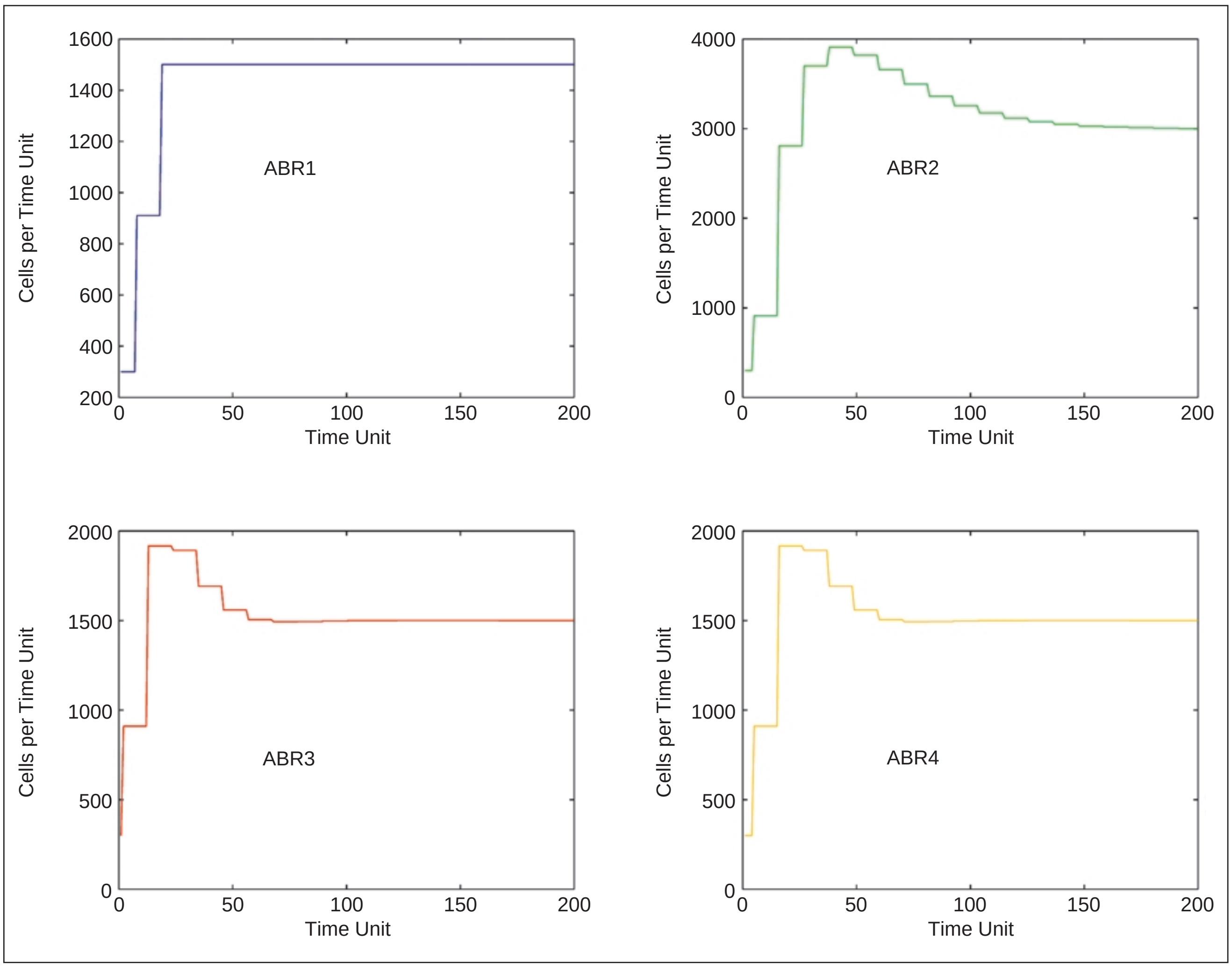

Figure 15. *Max-min fairness model: Source rates under Controller 3.*

To account for variability, we actually compare a source's bandwidth utilization to a number slightly smaller than $1/M$, say $0.85/M$.

The switch then computes the source optimal rates using these new weights. For example, consider the three sources ABR1, ABR2, and ABR3 at the switch. The total available capacity is 3000 cells per time unit. Source ABR1 is bottlenecked somewhere else at 500 cells per time unit. Then the new distribution of the weights is $a_{\text{ABR1}} = 1/6$, $a_{\text{ABR2}} = 5/12$, and $a_{\text{ABR3}} = 5/12$. Therefore, the optimal rates at the present switch will be 500 cells per time unit for ABR1 and 1250 cells per time unit for ABR2 and ABR3. These rates will be used as feedback information.

The simulations done with the adaptive weight algorithm indicate that max-min fairness is indeed achieved (Figs. 11, 12, 13, and 14). Initially (with all a_ms set to the fair share), the bottleneck node is the third one and sources ABR1, ABR3, and ABR4 share fairly the capacity at this switch (around 1500 cells per time unit). Meanwhile, source ABR2 only uses half of the capacity of switch 2. Actually its cell rate is slightly higher than the fair share. Because of the small queue length, the design appears to increase the allowed cell rate slightly. Once a_ms at each switch are adapted, source ABR2 should increase its rate to 3220 cells per time unit while the other sources still transmit at 1500 cells per time unit. The simulations show that source ABR2 effectively catches up (Figs. 11 and 13).

- *Controller 3*: One advantage of using Controller 3 is that it achieves max-min fair bandwidth allocation without necessitating any sort of adjustment of parameters, assuming of course that the available service rate remains constant. We simulate the configuration depicted in Fig. 5 taking the constant service rate to be 4500 cells per time unit for each switch. As the simulation results indicate (Fig. 15), the algorithm converges to the max-min fair allocation rather fast while stabilizing the queue around the desired set point, $Q = 700$ at SW3.

Conclusions

In this article, we have presented a control-theoretic approach to designing ABR congestion control algorithms. ATM networks deal with different types of traffic. Among the several services offered by ATM, the ABR service plays a central role in regulating the network traffic, as it is the only service category that uses explicit feedback from the network. We have presented several algorithms for ABR congestion control and have shown that they perform well under various criteria such as basic fairness and max-min fairness while achieving high bandwidth utilization. In addition, excessive cell loss is avoided by controlling the queue length. In our designs, the network delay is explicitly taken into account, which we believe is useful in modeling network behavior over links with high delays, such as satellite ATM networks, or IP over ATM. Furthermore, we have shown that one of our algorithms is robust to uncertainty in round-trip delays, and at the same time it can adapt to the variations in the number of sources sharing a switch. These properties make this algorithm easier to apply in volatile networks, where both the network delay and the number of active sessions are not known or cannot be predicted accurately beforehand. In summary, we strongly believe that control theoretic design tools can be effectively used in designing high-performance algorithms in the context of ATM ABR congestion control.

Acknowledgment

The research reported here was supported by NSF Grant NSF ANI 98-13710, NSF CAREER Award NCR 9701525, AFOSR MURI Grant AF DC 5-36128, and a grant from Rome Labs (Air Force Contract F30602-96C0156) through Scientific Systems Company, Inc.

References

[1] The ATM Forum, Technical Committee, "Traffic management, Version 4.1," pp. 43-55, af-tm-0121.000, Mar. 1999.

[2] J-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proc. ACM SIGCOMM'93*, San Francisco, CA, Sept. 1993, pp. 289-298.

[3] J-C. Bolot, "A self-tuning regulator for overload control in communication networks," in *Proc. 31st IEEE Conf. Decision and Control*, Tucson, AZ, Dec. 1992, pp. 1022-1023.

[4] J-C. Bolot and A.U. Shankar, "Analysis of a fluid approximation to flow control dynamics," in *IEEE INFOCOM'92*, Florence, Italy, May 1992, pp. 2398-2407.

[5] S. Keshav, "A control-theoretic approach to flow control," in *Proc. SIGCOMM'91*, Zurich, Switzerland, Sept. 1991, pp. 3-15.

[6] E. Altman, T. Başar, and R. Srikant, "Robust rate control for ABR sources," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1998.

[7] E. Altman, T. Başar, and R. Srikant, "Congestion control as a stochastic control problem with action delays," *Automatica*, vol. 35, pp. 1937-1950, Dec. 1999.

[8] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, "The ERICA switch algorithm for ABR traffic management in ATM networks," *IEEE/ACM Trans. Networking*, vol. 8, no. 1, pp. 87-98, Feb. 2000.

[9] O. Ait-Hellal, E. Altman, and T. Başar, "Rate-based flow control with bandwidth information," *Eur. Trans. Telecommunications*, vol. 8, no. 1, pp. 55-65, 1997.

[10] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: The case of a single congested node," *IEEE/ACM Trans. Networking*, vol. 1, no. 6, pp. 693-707, 1993.

[11] L. Benmohamed and Y. T. Wang, "A control-theoretic ABR explicit rate algorithm for ATM switches with per-VC queueing," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1998, pp. 183-190.

[12] A. Kolarov and G. Ramamurthy, "A control-theoretic approach to the design of an explicit rate controller for ABR service," *IEEE Trans. Networking*, vol. 7, pp. 741-753, Oct. 1999.

[13] A. Kolarov and G. Ramamurthy, "A control theoretic approach to the design of closed loop rate based flow control for high speed ATM networks," in *Proc. IEEE INFOCOM*, 1997.

[14] S. Mascolo, D. Cavendish, and M. Gerla, "ATM rate based congestion control using a Smith predictor: An EPRCA implementation," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 569-576.

[15] D.P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[16] E. Altman and T. Başar, "Optimal rate control for high speed telecommunication networks," in *Proc. 34th IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 1995, pp. 1389-1394.

[17] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[18] O.C. Imer and T. Basar, "Optimal solution to a team problem with information delays: An application in flow control for communication networks," in *Proc. 38th IEEE Conf. Decision and Control*, Phoenix, AZ, Dec. 1999, pp. 2697-2702.

[19] C. Fulton, S.-Q. Li, and C.S. Lim, "UT: ABR feedback control with tracking," in *Proc. IEEE INFOCOM*, 1997, pp. 806-815.

[20] O.C. Imer, T. Başar, and R. Srikant, "A robust adaptive control algorithm for ABR service in ATM networks," in *Proc. IEEE ICCCN*, Las Vegas, NV, Oct. 2000, pp. 48-53.

[21] M.W. Garrett. "Contributions toward real-time services on packet switched networks," Ph.D. dissertation, Columbia Univ., New York, May 1993.

[22] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proc. 20th Annual Conf. Local Computer Networks*, Minneapolis, MN, 1995, pp. 397-406.

[23] "MPEG-1 frame size traces," [Online]. Available FTP: ftp-info3.informatik.uni-wuerzburg.de, /pub/MPEG/

[24] "Video traces,"1992, Bell Communications Research, Inc. [Online]. Available FTP: thumper.bellcore.com, pub/vbr.video.trace

[25] R. Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey," *Computer Networks ISDN Syst. (Netherlands)*, vol. 28, no. 13, pp. 1723-38, Oct. 1996.

[26] P.J. Brockwell and R.A. Davis, *Times Series: Theory and Methods*. New York: Springer-Verlag, 1991.

[27] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, and B. Vandalore, "On determining the fair bandwidth share for ABR connections in ATM networks," in *Proc. IEEE Int. Conf. Communications (ICC) 1998*, June 1998, pp. 1435-1451.

Orhan Cagri Imer received his B.S. degree from Middle East Technical University, Ankara, Turkey, in 1997 and his M.S. degree from the University of Illinois at Urbana-Champaign in 2000, both in electrical engineering. Currently, he is a Research Assistant at the Coordinated Science Laboratory at the University of Illinois. His research interests include control and discrete-event simulation of communication networks and market-based resource allocation in agent systems.

Sonia Compans received the Baccalaureate from Academie de Toulouse, France, in 1992, the B.S. degree from the National Polytechnic Institute of Lorraine, Ecole Nationale Superieure d'Electricitè et de Mecanique, France, in 1997, and the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1998. Since then she has been with Alcatel in France, working in the telecommunications field.

Tamer Başar received the B.S.E.E. degree from Robert College, Istanbul, in 1969, and the M.S., M.Phil, and Ph.D. degrees in engineering and applied science from Yale University, in 1970, 1971, and 1972, respectively. Since 1981, he has been with the University of Illinois at Urbana-Champaign, where he is currently the Fredric G. and Elizabeth H. Nearing Professor of Electrical and Computer Engineering and Research Professor at the Coordinated Science Laboratory. His current research interests are robust nonlinear and adaptive control, control of and over communication networks, risk-sensitive estimation and control, and robust identification. He is a member of the National Academy of Engineering, a Fellow of the IEEE, Past President of the IEEE Control Systems Society, Deputy Editor-in-Chief of *Automatica*, and a member of editorial and advisory boards of a number of other international journals. In 1993, he received the Medal of Science of Turkey and the Distinguished Member Award of the IEEE Control Systems Society, and in 1995 the Axelby Outstanding Paper Award of the same society.

R. Srikant received the B.Tech. degree from the Indian Institute of Technology, Madras, India, in 1985 and the M.S. and Ph.D. degrees from the University of Illinois in 1988 and 1991, respectively, all in electrical engineering. He was a Member of Technical Staff in AT&T Bell Laboratories from 1991 to 1995. Since August 1995, he has been with the University of Illinois where he is currently an Associate Professor in the Department of General Engineering and the Coordinated Science Laboratory and an Affiliate in the Department of Electrical and Computer Engineering. His research interests include communication networks, queueing theory, and stochastic control. He is currently an Associate Editor of *Automatica*. He received an NSF CAREER award in 1997.