Optimal Structured Feedback Policies for ABR Flow Control Using Two-Timescale SPSA

Shalabh Bhatnagar, Michael C. Fu, Steven I. Marcus, *Fellow, IEEE*, and Pedram J. Fard

***Abstract*—Optimal structured feedback control policies for rate-based flow control of available bit rate service in asynchronous transfer mode networks are obtained in the presence of information and propagation delays, using a numerically efficient two-timescale simultaneous perturbation stochastic approximation algorithm. Models comprising both a single bottleneck node and a network with multiple bottleneck nodes are considered. A convergence analysis of the algorithm is presented. Numerical experiments demonstrate fast convergence even in the presence of significant delays. We also illustrate performance comparisons with the well-known Explicit Rate Indication for Congestion Avoidance (ERICA) algorithm and describe another algorithm (based on ERICA) that does not require estimating available bandwidth (as in ERICA).**

***Index Terms*—Network of nodes, optimal structured feedback policies, rate-based ABR flow control, single bottleneck node, two-timescale SPSA.**

I. Introduction

THE AVAILABLE bit rate (ABR) service in asynchronous transfer mode (ATM) networks is used primarily for data traffic. Bandwidth allocation for ABR service is done after the higher priority services such as constant bit rate (CBR) and variable bit rate (VBR) have been allocated bandwidth. The available bandwidth is a time-varying quantity, and for proper utilization, the network requires the ABR sources to control their individual traffic flows. The proposals discussed by the ATM forum for flow control in ABR service [26] were broadly classified into two categories: rate-based and credit-based schemes. The rate-based scheme [8] was finally accepted by the ATM forum primarily because of the higher hardware complexity and costs involved in the latter [26]. Several algorithms for rate-based flow control have since been proposed at the ATM forum. We discuss these in detail below since we use a rate-based scheme in this paper.

Initial proposals for rate-based ABR flow control were based on single bit feedback schemes [15], [23]. Using some of these schemes, the system was either found to become unstable or suffer from the beat-down problem [26]. In general, single-bit feedback schemes did not work well for ATM networks, and as a result, explicit rate feedback schemes [9] were proposed. The idea in these was that the switches would compute the ABR source rates (based on the level of congestion) and feedback these directly to the sources. These schemes were found to substantially improve system performance and offer more flexibility to switch designers. Several explicit-rate feedback schemes have been proposed in recent years; amongst these are Enhanced PRCA (EPRCA) [22], the Dynamic Max Rate Control Algorithm (DMRCA) [10], and the well-known Explicit Rate Indication for Congestion Avoidance (ERICA) [16]. We shall consider the last algorithm above in detail for comparison purposes with our algorithm. In [3], the controller computes the input rate as a linear function of past rates and queue levels, and the algorithm requires a complicated tuning of control parameters in order to ensure stability. In [19] and [18], the Smith principle is used with a simple controller. However, the feedback law necessarily requires that the queue length information be fed back with rate computation done at the source end of the system, making the scheme incompatible with ATM forum standards. In [1], the problem is formulated as a stochastic control problem but with linearized queue dynamics, where the queue length may become negative. In [25], a continuous-time queueing model is studied and stability conditions for various controllers obtained. However, performance analysis is done only under the assumptions of no delays and the system being continuously observed, which seem unrealistic in the ABR context.

In this paper, we consider both a single bottleneck node model (as in [25]) and its extension to a network of such nodes for explicit rate-based flow control. The traffic from ABR sources (to be controlled) is modeled using controlled Poisson processes. The uncontrolled traffic (representing all the other traffic in the network) is modeled using appropriate Markov-modulated Poisson processes (MMPP). Our models include delays in receiving rate information and in packet transmissions. We consider structured feedback policies that have several levels of control. We develop a simultaneous perturbation stochastic approximation (SPSA) [27], [14] variant of a two-timescale stochastic approximation algorithm in [5] to obtain the optimal policy with this structure. The

Manuscript received March 5, 1999; revised January 17, 2000 and January 25, 2001; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. H. Low. This work was supported by the National Science Foundation under Grant DMI-9713720, by the Semiconductor Research Corporations under Grant 97-FJ-491, and by DoD under Contract MDA90497C3015.

S. Bhatnagar was with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA and the Division of Mathematics and Computer Science, Free University, Amsterdam, The Netherlands. He is now with the Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi, India (e-mail: shalabh@cse.iitd.ernet.in).

M. C. Fu is with the Robert H. Smith School of Business and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: mfu@rhsmith.umd.edu).

S. I. Marcus is with the Department of Electrical and Computer Engineering, and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: marcus@eng.umd.edu).

P. J. Fard was with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA. He is now with Hughes Network Systems, Germantown, MD 20876 USA (e-mail: pfard@hns.com).

Publisher Item Identifier S 1063-6692(01)06849-2.

two-timescale stochastic approximation algorithm developed in [5] (see also [2]) for simulation-based parametric optimization had the advantage that it updates the parameter at deterministic instants obtained using two different step sizes (time scales), without the need for regeneration, as is typically the case with traditional perturbation analysis schemes [11], [12]. On the other hand, like other finite-difference schemes, it requires $(N+1)$ simulations for any N-vector parameter to obtain the gradient estimate. In [4], this algorithm was also used for a single bottleneck node ABR problem. As a result of slow convergence, only three-level parameterized policies were considered in the numerical experiments. Here, we consider not just the single bottleneck node case with policies that have more levels of control, but also a network of multiple nodes and multiple ABR sources or virtual connections (VCs). Our two-timescale SPSA algorithm requires only two parallel simulations for any N-vector parameter, with all the N-components of the parameter vector updated simultaneously. For the multiple bottleneck node case, we compare the performance of our algorithm using structured feedback policies with the well-known congestion-control scheme ERICA for different settings. Furthermore, using our techniques, we also obtain an "optimal" ERICA-type policy which, however, does not require an estimate of available bandwidth like ERICA. We observe that even though ERICA is "fairer" in some cases than our policies, in all cases that we studied, our algorithm shows a variation in the queue length process that is many times lower than ERICA. For the single bottleneck node, our numerical experiments also highlight the substantial performance gains obtained by employing structured feedback policies that we propose over optimal open loop policies.

The rest of the paper is organized as follows. In Section II, we describe both the single bottleneck node and the network models, and formulate the optimization problem. In Section III, we describe our two-timescale SPSA scheme for obtaining the optimal structured policy, briefly compare it with two previously proposed two-timescale stochastic approximation algorithms (cf. [5]) and present our main result. The convergence analysis is briefly presented in the Appendix, with detailed proofs provided in [6]. In Section IV, we present numerical experiments with both types of models. Finally, Section V provides concluding remarks and extensions.

II. The Optimization Problem

In this section, we present two models: the basic single bottleneck node model and its extension to a network of bottleneck nodes with multiple ABR sources and/or VCs.

A. Single Bottleneck Node Model

Our basic model, shown in Fig. 1, is a bottleneck node with two input streams, one controlled (representing the traffic from the ABR source), and the other uncontrolled (representing all the other traffic in the network passing through this node). The ABR stream is modeled as a controlled Poisson process with instantaneous intensity specified by a feedback-control law defined below. The uncontrolled stream is modeled as an MMPP. Note that the uncontrolled stream represents traffic from CBR, VBR, and other ABR sources, and is thus very bursty in nature. An MMPP model is appropriate for this kind of traffic. The MMPP rates are set in the following manner: Let $\{X(t),\, t \geq 0\}$ be a finite state, irreducible, aperiodic Markov process with state–space S_u. When $X(t) = i \in S_u$, the instantaneous rate of the uncontrolled stream is $\lambda_{u,\,i}$. For given $T > 0$ fixed, let $X_n \triangleq X(nT)$ represent the state of the modulating chain of the uncontrolled MMPP at time nT. Let $p(i;\;j),\, i,\, j \in S_u$ represent the transition probabilities of $\{X_n\}$. In the following, we assume that we observe the queue length and use that information for obtaining the ABR rate. This assumption is quite common in the ABR literature, see for instance, EPRCA [22], DMRCA [10], and also [26]. The size of the bottleneck buffer is B and could be large (e.g., in the simulation experiments that we illustrate in Section IV, B is taken as 5×10^5). Let $q_n,\, n \geq 0$, represent the queue length observed at times $nT,\, n \geq 0$. We assume that the ABR rate is held fixed in the time intervals $[nT,\, (n+1)T),\, n \geq 0$, with $\lambda_c(n)$ representing this ABR rate (in the nth interval) and is computed using the queue length (q_n) observed at the node. The new rate is then fed back to the source. Thus, the scheme that we propose is essentially an explicit rate feedback scheme. The above information however reaches the ABR source with a delay D_b, whereupon the ABR source starts sending packets with the new rate. Further, there is a propagation delay D_f in the path of packets arriving at the bottleneck node from the source. We assume throughout that the quantities T, D_b, and D_f are constants. Let $S = \{0,\, 1,\, \ldots,\, B\}$ be the set of possible queue length values. Let θ represent the parameter to be optimized that takes values in a compact set $C \subset \mathcal{R}^N$. We assume, in particular, that C is of the form $\prod_{i=1}^{N} [\lambda_{i,\,\min},\, \lambda_{i,\,\max}]$, with $\lambda_{i,\,\min} > 0$, for all $i = 1,\, \ldots,\, N$. Let $h\colon S \rightarrow \mathcal{R}^+$ be a given bounded and nonnegative cost function ($\mathcal{R}^+$ is the space of nonnegative reals). Our aim is to find a θ that minimizes the long-run average cost

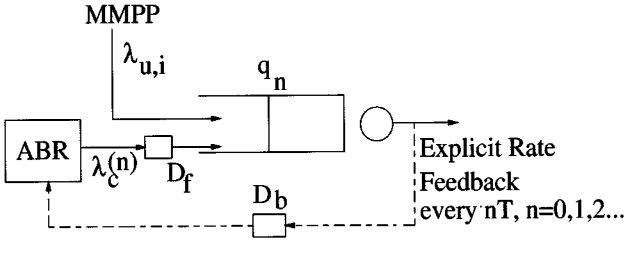

Fig. 1. Single bottleneck node.

$$J(\theta) \triangleq \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^{n} h(q_i). \tag{1}$$

The feedback policy that governs the rate $\lambda_c(n)$ is given as follows. Let $a_i,\, i = 0,\, 1,\, \ldots,\, N$ be integers such that $-1 = a_0 < a_1 < a_2 < \cdots < a_{N-1} < a_N = B$. Then the sets $S_i = \{a_{i-1}+1,\, \ldots,\, a_i\},\, i = 1,\, \ldots,\, N$, form a partition of S. For $i = 1,\, \ldots,\, N$, let

$$\lambda_c(n) = \lambda_i \qquad \text{if } q_n \in S_i. \tag{2}$$

In the above, $\theta \triangleq (\lambda_1,\, \lambda_2,\, \ldots,\, \lambda_N)^T$ is the controllable parameter. Markov decision processes (MDP) [21] represent a general

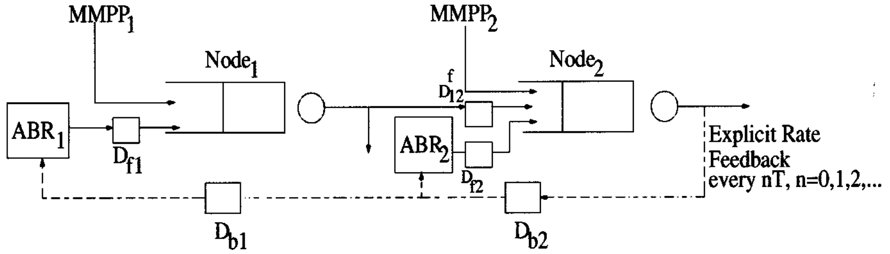

Fig. 2. Two-node network.

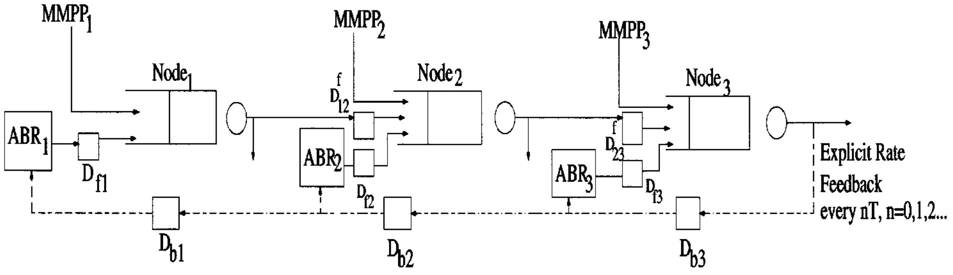

Fig. 3. Three-node network.

framework for dealing with dynamic decision making problems under uncertainty (for example when $N = B$ and $S_i = \{i\}$ above). However, a numerical solution based on standard MDP solution techniques [21] faces the "curse of dimensionality" for large state spaces. Our numerical experiments in Section IV, for instance, have $B = 5 \times 10^5$. In addition, MDP solution techniques also require explicit specification of the transition probabilities of the continuous time chain at instants T, $2T$, etc. In the presence of nonzero delays D_b and D_f, these techniques become computationally prohibitive. Our two-timescale SPSA algorithm is very effective in solving such problems. Next, we illustrate the extension of the above model to a network with multiple bottleneck nodes.

B. Network with Multiple Nodes

Here, we will give a precise mathematical model considering zero delays, since the notation for the delayed case becomes extremely cumbersome. The delayed case will, however, be explained in detail. Consider, for instance, a network with several nodes and ABR sources (Figs. 2 and 3 are examples of networks of bottleneck nodes with multiple sources that we consider in the numerical experiments). Thus, we consider several VCs that use any given node. There are various feedforward and feedback delays on the network paths (e.g., Figs. 2 and 3). Note that in the previous model (of a single bottleneck node), we consider only one VC for optimization purposes, with the rest of the traffic (ABR and otherwise) captured in the uncontrolled MMPP stream. We, however, still assume (for the network case) that there is uncontrolled MMPP traffic that passes through any given bottleneck node. Also, a certain fraction of these packets randomly enters the next bottleneck node with the remaining uncontrolled packets dropped after service. Our model is thus very general, since there could be some CBR, VBR, and "other" ABR sources comprising an MMPP stream that have more than one bottleneck node on their paths. All external MMPP streams entering individual bottleneck nodes are assumed independent of one another and have possibly different parameters. Thus, in this model, we are concerned about finding optimal ABR rates of several connections having (possibly) different virtual paths simultaneously. We again assume (as before) that the queue length at each node or switch in the network is observed every T units of time. Based on the queue length information, each switch computes an ABR rate for each individual VC that uses the node. This information is then transmitted with a delay from the current node to either the next switch on the backward path of the VC or the ABR source itself (if there is no other node in between). Thus, in particular, rate computation for individual VCs at any given node involves computing the minimum of the rate computed by the current node and the latest available (delayed) rate set for that VC by its successor node on the forward path. We present here a simple feedback policy. Our techniques allow for various interesting modifications of this policy. In Section IV, we also present a policy based on ERICA that uses $h_j(n)$ (see below) in place of an estimate of available bandwidth (that ERICA requires for rate allocation) and use our techniques to obtain an optimal such policy, that we call ERICA$_{\text{opt}}$. If $q_{j,n}$ is now the queue length at instant nT of bottleneck node j (this will be different for different nodes) and $S_{j,i}, i = 1, \ldots, N_j$, are subsets of S as in (2) corresponding to node j, we let

$$h_j(n) = \lambda_{j,i} \qquad \text{if } q_{j,n} \in S_{j,i} \tag{3}$$

where $h_j(n)$ is some auxiliary rate computed by node j. Note that the number (N_j) of subsets $(S_{j,i})$ of S corresponding to node j can be different for different nodes. Now, if there are (say) L VCs that have node j on their virtual path, then the ABR rate set by this node for any given VC is the minimum of $h_j(n)/L$ and the (latest known) ABR rate set by the successor node of the given VC, from the current node, on the forward path of the VC. The parameter to be tuned (now) can be written as $\theta = (\theta_1, \ldots, \theta_M)^T$, where M is the number of nodes and where each $\theta_j = (\lambda_{j,1}, \ldots, \lambda_{j,N_j})^T$ is the vector of rates for the particular node j. The objective is to minimize the average cost [as in (1)], now defined by

$$J(\theta) \overset{\Delta}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \hat{h}(q_{1,i}, \ldots, q_{M,i}) \tag{4}$$

with appropriately defined real valued cost function $\hat{h}$. In the next section, we present a computationally efficient stochastic approximation algorithm for finding optimum θ.

III. Two-Timescale SPSA Algorithm

We present the algorithms here for the single bottleneck node model. Thus, $\theta \overset{\Delta}{=} (\lambda_1, \ldots, \lambda_N)^T$ is the parameter vector to be tuned in order to minimize $J(\theta)$ in (1). The corresponding algorithms for the network case are similar to these. We first briefly motivate the use of stochastic approximation, i.e., when $J(\theta)$ is not available analytically but must be estimated by simulation (see [13], [17]). A stochastic approximation algorithm recursively updates θ using gradient descent with decreasing step-sizes and an appropriate estimate of $\nabla J(\theta)$. The efficiency of the algorithm usually depends on the quality and computational requirements of the gradient estimate.

The advantages of using the SPSA approach are best appreciated after first presenting the original two-timescale algorithm of [5]. Let $\delta > 0$ be a small fixed constant. Define positive real-valued sequences $\{a(n)\}$ and $\{b(n)\}$ as follows: $a(0) = \hat{a}$, $b(0) = \hat{b}$, $a(i) = \hat{a}/i$, $b(i) = \hat{b}/i^{\alpha}$, $1/2 < \alpha < 1$, $\hat{a}, \hat{b} > 0$, $i \geq 1$. Then

$$\frac{a(n+1)}{a(n)}, \frac{b(n+1)}{b(n)} \to 1, \qquad \text{as } n \to \infty \tag{5}$$

$$\sum_n a(n) = \sum_n b(n) = \infty, \qquad \sum_n a^2(n), \sum_n b^2(n) < \infty \tag{6}$$

$$a(n) = o(b(n)). \tag{7}$$

Define $\{n_m, m \geq 0\}$ as follows: $n_0 = 1$ and $n_{m+1} = \min\{j > n_m \mid \sum_{i=n_m+1}^{j} a(i) \geq b(m)\}$, $m \geq 1$. Let $\theta(m)$ represent the mth update of θ. Let $\pi_i(\lambda)$ denote the point closest to $\lambda \in \mathcal{R}$ in the interval $[\lambda_{i,\min}, \lambda_{i,\max}] \subset \mathcal{R}$ (defined earlier) and $\pi(\theta)$ be defined by $\pi(\theta) = (\pi_1(\lambda_1), \pi_2(\lambda_2), \ldots, \pi_N(\lambda_N))^T$. Let $\lambda_i(m)$ be the mth update of λ_i. The $(N+1)$ parallel simulations are obtained as follows. The first simulation generates $\{(q_j, X_j)\}$ (recall that q_j denotes the queue length and X_j the state of the MMPP, both at instants jT), governed by $\{\tilde{\theta}_j\}$, where $\tilde{\theta}_j = \theta(m)$, for $n_m \leq j < n_{m+1}$. The remaining N parallel simulations generate the corresponding processes $\{(\overline{q}_j^i, \overline{X}_j^i)\}$, $i = 1, \ldots, N$, governed by $\{\tilde{\theta}_j^i\}$, $i = 1, \ldots, N$, where $\tilde{\theta}_j^i = \pi(\theta(m) + \delta e_i)$, $i = 1, \ldots, N$, $n_m \leq j < n_{m+1}$, and e_i is the unit vector in the ith direction. Then the algorithm is as follows: For $i = 1, \ldots, N$

$$\lambda_i(m+1) = \pi_i\left(\lambda_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j)\left(\frac{h(q_j) - h(\overline{q}_j^i)}{\delta}\right)\right). \tag{8}$$

An alternative (proposed and used in [5]) to using $(N+1)$ parallel simulations is to move the algorithm in cycles during each of which only two simulations are used as follows. The first simulation corresponds to $\{(q_j, X_j)\}$ and is governed by $\{\tilde{\theta}_j\}$, defined as earlier, and the second simulation is represented as $\{(\overline{q}_j, \overline{X}_j)\}$, which is governed by $\{\hat{\theta}_j\}$ defined by $\hat{\theta}_j = \pi(\theta(m)+\delta e_i)$ for $j = n_{Nm+i-1}, n_{Nm+i-1}+1, \ldots, n_{Nm+i}-1$, $i = 1, \ldots, N$, $m \geq 0$. The algorithm of [5] is

$$\lambda_i(m+1) = \pi_i\left(\lambda_i(m) + \sum_{j=r_{i-1}+1}^{r_i} a(j)\left(\frac{h(q_j) - h(\overline{q}_j)}{\delta}\right)\right). \tag{9}$$

Here, $r_i = n_{Nm+i}$ and $r_{i-1} = n_{Nm+i-1}$ (used above for notational simplicity). Thus instead of all components being updated every n_m steps, $m \geq 1$, as in (8), only one component is updated now every n_m steps and the algorithm thus moves in bigger loops or cycles of n_{Nm} with all components updated once at the end of the bigger loop. It is clear that one needs only two simulations in this manner but there is a tradeoff with speed of convergence. We return to this issue after we present our two-timescale SPSA algorithm next.

For any $m \geq 0$, let $\Delta(m) \in \mathcal{R}^N$ be a vector of mutually independent and mean zero random variables $\{\Delta_{m,1}, \ldots, \Delta_{m,N}\}$, [*viz.*, $\Delta(m) \overset{\Delta}{=} (\Delta_{m,1}, \ldots, \Delta_{m,N})^T$] taking values in a compact set $E \subset \mathcal{R}^N$ and having a common distribution. We assume that these random variables satisfy Condition A) below.

Condition A): There exists a constant $\overline{K} < \infty$, such that for any $m \geq 0$, and $i \in \{1, \ldots, N\}$, $E[\Delta_{m,i}^{-2}] \leq \overline{K}$.

We further assume that $\{\Delta(m)\}$ is a mutually independent sequence with $\Delta(m)$ independent of $\sigma(\theta(l), l \leq m)$, the latter being the filtration generated by the sequence of parameter updates upto instant m. Condition A) is a standard condition in SPSA algorithms [27]. Define parallel processes $\{(q_j^1, X_j^1)\}$ and $\{(q_j^2, X_j^2)\}$ such that for $n_m < j \leq n_{m+1}$, $\{(q_j^1, X_j^1)\}$ is governed by $\pi(\theta(m) - \delta\Delta(m)) \overset{\Delta}{=} (\pi_1(\lambda_1(m) - \delta\Delta_{m,1}), \ldots, \pi_N(\lambda_N(m) - \delta\Delta_{m,N}))^T$. Similarly, $\{(q_j^2, X_j^2)\}$ is governed by $\pi(\theta(m) + \delta\Delta(m))$ defined analogously. In the above, $\theta(m)$ is governed by the following recursion equations. For $i = 1, \ldots, N$

$$\lambda_i(m+1) = \pi_i\left(\lambda_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j)\left(\frac{h(q_j^1) - h(q_j^2)}{2\delta\Delta_{m,i}}\right)\right). \tag{10}$$

Note that we choose $\delta > 0$ to be a fixed small constant in this algorithm (as in the previous algorithms above), unlike the usual Kiefer–Wolfowitz algorithms that require δ to go to zero. This has been done to ensure that the variance does not blow up. However, one could modify the algorithm to allow δ to go to zero slowly enough so that the variance does not increase rapidly in the beginning of the algorithm. We observed in the numerical experiments that a small enough δ chosen arbitrarily works well. In [7], another two timescale SPSA algorithm that updates the parameter after every fixed number of epochs has recently been developed and, that along with algorithm (10), have been applied for optimizing parameters in uncountable state hidden Markov models under a Liapunov stability assumption.

We now discuss the reasons for the SPSA scheme in (10) to be computationally more efficient than both schemes (8) and (9). We begin with (9) first. It was shown in [5] that the scheme (9) tracks trajectories of an ordinary differential equation (o.d.e.) similar to (11) below, but with a factor of $1/N$ multiplying the RHS of it. Also, the scheme (8) tracks trajectories of (11) as is. This means that even though the qualitative behavior of the algorithm (9) is the same as that of (8) and (10) (as shown in Appendix), the factor of $1/N$ on the RHS of (11) essentially serves to slow down its rate of convergence. Hence, Theorem 3.1 (below) indicates that we no longer need $(N+1)$ parallel simulations for an N-vector parameter as (8) would require, while at the same time we do not compromise on the speed of convergence. The convergence analysis proceeds through a sequence of steps and is given in the Appendix. However, due to lack of space, all the details in the proofs have not been provided here, but can be found in [6]. We state now our main result, the proof of which is sketched in the Appendix.

The o.d.e. technique is commonly used to prove convergence of stochastic approximation algorithms. Here, we show that the algorithm (10) asymptotically converges to the stable points of the o.d.e. (11) below. Let $\tilde{Z}(t) \stackrel{\Delta}{=} (\tilde{Z}_1(t), \ldots, \tilde{Z}_N(t)) \in \mathcal{R}^N$, where $\tilde{Z}_i(t), i = 1, \ldots, N$, satisfy the o.d.e.

$$\dot{\tilde{Z}}_i(t) = \tilde{\pi}_i\left(-\nabla_i J\left(\tilde{Z}(t)\right)\right), \qquad t \geq 0, \qquad \tilde{Z}(0) \in C \tag{11}$$

where, for any bounded, continuous, real-valued function $v(\cdot)$

$$\tilde{\pi}_i(v(y)) = \lim_{0<\Delta\to 0}\left(\frac{\pi_i(y+\Delta v(y)) - \pi_i(y)}{\Delta}\right).$$

For $x = (x_1, \ldots, x_N)^T$, let $\tilde{\pi}(x) = (\tilde{\pi}_1(x_1), \ldots, \tilde{\pi}_N(x_N))^T$. The role played by the operator $\tilde{\pi}(\cdot)$ is, in some sense, to force the o.d.e. (11) to evolve within the constraint set C. Let $K = \{\theta \in C \,|\, \tilde{\pi}(\nabla J(\theta)) = 0\}$ represent the set of local optima. Also, for $\eta > 0$, let $K^\eta = \{\theta \in C \,|\, \exists\, \theta' \in K \text{ s.t. } ||\theta - \theta'|| \leq \eta\}$ represent the set of points within η of local optima.

Theorem 3.1: Given $\eta > 0$, $\exists\, \overline{\delta} > 0$ such that for any $\delta \in (0, \overline{\delta}]$, the algorithm (10) converges to K^η almost surely.

Note that the above theorem only gives the existence of a $\overline{\delta} > 0$ such that, for any $\delta \leq \overline{\delta}$, the algorithm will almost surely converge to a "small" neighborhood of the optimal point. However, as explained earlier, we found in the numerical experiments that any small enough δ chosen arbitrarily seems to work well.

IV. Numerical Results

We divide this section into two subsections. In the first of these, we provide numerical results for the single bottleneck node case, wherein we show experiments with various parameter settings using our two-timescale SPSA scheme on structured feedback policies (2) and compare their performance with optimal open loop policies. Next, we show numerical experiments with two and three node networks shown in Figs. 2 and 3, respectively. Here, we first use our SPSA algorithm on the type of policies described in Section II-B and compare the performance of the optimal policy within that class with the well-known ERICA algorithm. We also provide an algorithm (based on ERICA) that does not require estimating the available bandwidth (as ERICA does) and use our algorithm (10) to obtain an optimal policy within the class of ERICA-type policies.

In implementing stochastic approximation algorithms, the convergence rate in practice often depends heavily on the choices of the step size sequence parameters $\hat{a}$, $\hat{b}$ and α. In our experiments, we choose $\hat{a} = \hat{b} = 1$, and $\alpha = 2/3$, as these values worked well in the experiments reported in [5], and appear to work well in all of the cases considered here. However, in general, adjustments might need to be made for particular problems. For example, if the algorithm appears to be moving too slowly, then larger values of $\hat{a}$ and $\hat{b}$ might be required; similarly, an overly oscillating iterate sequence would probably call for scaling these parameters downwards.

A. Single Bottleneck Node

Flow control in ABR service requires balancing various conflicting performance criteria, such as mean and variance of delay and throughput. Often, this is addressed by minimizing the distance of stationary mean queue length from a given fixed constant N_0 [28], [25], [4]. We adopt a similar approach here and choose $h(x) = |x - N_0|$, to be our cost function with N_0 assumed given. In the concluding section, we also indicate ways to obtain an optimal such N_0. We compare the performance of optimal structured closed loop feedback policies of type (2) obtained by applying the two-timescale SPSA algorithm (10), with the optimal open-loop policy, defined by setting $\lambda_c(n) = \lambda^*$ for all n, where λ^* is obtained by applying the two-timescale algorithm (8) for the scalar case ($N = 1$). Note that the optimal open-loop policy has a fixed rate, and thus does not adapt to observed queue lengths.

For the closed-loop policies, we perform experiments with policies that have 5- and 11-parameter levels, respectively. We do not show here the details of the experiments with 11-level policies due to space considerations and also since the observations there are similar to those of 5-level policies. However, these can be found in [6]. We assume throughout that both D_b and D_f are integral multiples of T. The form of the five-level policies for obtaining $\lambda_c(n)$ is as follows:

$$\lambda_c(n) = \begin{cases} \lambda_1^*, & \text{if } q_n < N_0 - 2\epsilon \\ \lambda_2^*, & \text{if } N_0 - 2\epsilon \leq q_n < N_0 - \epsilon \\ \lambda_3^*, & \text{if } N_0 - \epsilon \leq q_n \leq N_0 + \epsilon \\ \lambda_4^*, & \text{if } N_0 + \epsilon < q_n \leq N_0 + 2\epsilon \\ \lambda_5^*, & \text{if } q_n > N_0 + 2\epsilon. \end{cases} \tag{12}$$

ϵ (above) is a given fixed constant. We actually consider a generalization of the model in Fig. 1, with rate feedback done at instants nF_b, $n \geq 1$, for F_b a fixed multiple of T. This gives us added flexibility in studying the effect of changes in F_b in addition to those in T. The role played by F_b is, in some sense, that of an additional delay. The sequence of events is thus as follows. The ABR rate $\lambda_c(\cdot)$ is computed at times nT, $n \geq 1$, at the node using feedback policies above. These rates are fed back to the source every F_b units of time. The source receives this rate information with a delay D_b, and upon receiving it, immediately starts sending packets with the new rate. The packets arrive at the node with a propagation delay D_f. The uncontrolled process is an MMPP with the underlying Markov chain chosen for simplicity to be an irreducible two-state chain. To simplify the simulation code, we assume that the underlying chain undergoes state transitions every T units of time. The buffer size B is 5×10^5. We tested our SA scheme on various combinations of the parameters D_b, D_f, N_0, ϵ, T, F_b, $\lambda_{u,i}$, $p(i;\ j)$. We also show experiments with two controlled sources feeding into the same bottleneck node, but with rate information ($\lambda_c(n)$) fed back with different delays (almost without any delay to the first and with a significant delay to the second). We observe that the bandwidth is shared equally by the two sources using our closed-loop policies (2).

Let θ^* denote the parameter value for the corresponding optimal policy, i.e., $\theta^* = (\lambda_1^*, \ldots, \lambda_5^*)^T$ for the closed-loop policy and $\theta^* = \lambda^*$ for the open-loop policy. In the following, subscript θ^* is used in the definition of various performance measures to indicate θ^*-parameterized stationary distributions of the various quantities. Thus, $\text{Var}_{\theta^*}(q_n)$ represents the stationary variance of $\{q_n\}$ parameterized by θ^*. Let B_d represent the segment or band (of queue length values) $[N_0 - \epsilon, N_0 + \epsilon]$. We compare performance in terms of parameters of queue length distributions and throughput rate: $\overline{q}$, P_{band}, σ_q^2, $\overline{\lambda}_c$, P_{idle} and $J(\theta^*)$. These quantities and their estimates are defined as follows:

$$\overline{q} \triangleq E_{\theta^*}[q_n] \approx \frac{1}{L_a} \sum_{i=1}^{L_a} q_i$$

$$\sigma_q^2 \triangleq \text{Var}_{\theta^*}(q_n) \approx \left(\frac{1}{L_a} \sum_{i=1}^{L_a} q_i^2 \right) - (\overline{q})^2$$

$$P_{band} \triangleq P_{\theta^*}(q_n \in B_d) \approx \frac{1}{L_a} \sum_{i=1}^{L_a} I\{q_i \in B_d\}$$

$$P_{\text{idle}} \triangleq P_{\theta^*}(q_n = 0) \approx \frac{1}{L_a} \sum_{i=1}^{L_a} I\{q_i = 0\}$$

$$\overline{\lambda}_c \triangleq E_{\theta^*}[\lambda_c(n)] \approx \frac{1}{L_a} \sum_{i=1}^{L_a} \lambda_c(i)$$

$$J(\theta^*) \approx \frac{1}{L_a} \sum_{i=1}^{L_a} |q_i - N_0|$$

where L_a is taken as 10^5 in our experiments. Note that all these performance estimates are computed after the SPSA algorithm converges. The last performance measure is the one that the algorithm seeks to minimize, but clearly the others are closely related. One desires P_{band} to be high in order to satisfy various other performance criteria. The measure P_{idle} gives the stationary probability of the server lying idle and should be close to zero. The average ABR throughput rate $\overline{\lambda_c}$ is often considered the most important measure of performance in ABR, because it is this measure which tells us whether the available bandwidth has been properly utilized or not. Also, clearly any good scheme should provide a low variance σ_q^2.

In the simulations, we choose $\lambda_1, \ldots, \lambda_4 \in [0.10, 3.0]$ and $\lambda_5 \in [0.10, 0.90]$. We choose λ_5 in a smaller range merely to speed up convergence, since we intuitively know that λ_5^* should be the lowest rate because it is applied when the queue length is in the "highest" region. We observed that any other choice for the range of λ_5 works well, too, as long as the range is smaller than that of $\lambda_1, \ldots, \lambda_4$; for instance, $\lambda_5 \in [0.1, 1.5]$ works as well. Moreover, the choice of this range does not depend on the number of ABR connections. We choose the same range as above for our experiments with two ABR sources feeding into the same bottleneck node in Table V and also for our experiments with networks of bottleneck nodes described in Section IV-B. If we, however, choose $\lambda_5 \in [0.10, 3.0]$ (the same range as $\lambda_1, \ldots, \lambda_4$), our algorithm takes longer to converge particularly when delays are high because of initial oscillations. We found that λ_5^* is higher when delays are high than when they are low. Thus, the range of λ_5 should not be made too small eitherl.

We choose the service time process to be i.i.d., exponential with rate $\mu = 1.0$. In [4], more general service time processes have been considered for a similar model with a Poisson uncontrolled stream, and similar results as for exponential service were obtained. The $\Delta_{m,i}$, $i = 1, \ldots, N$, in $\{\Delta(m)\}$ in our algorithm (10) are chosen as i.i.d. symmetric Bernoulli distributed with $\Delta_{m,i} = \pm 1$ w.p. 1/2. We arbitrarily choose $\delta = 0.12$ in the experiments. In the following, we consider two settings for the uncontrolled traffic: (a) $\lambda_{u,1} = 0.05$, $\lambda_{u,2} = 0.15$, $p(1;\ 1) = p(1;\ 2) = p(2;\ 1) = p(2;\ 2) = 0.5$, and (b) $\lambda_{u,1} = 0.2$, $\lambda_{u,2} = 0.4$, $p(1;\ 1) = p(2;\ 1) = 0.6$, $p(1;\ 2) = p(2;\ 2) = 0.4$. In Tables I–V, the two settings are summarized by the value of $\overline{\lambda}_u$ (the mean rate of the uncontrolled MMPP), which is 0.10 and 0.28 in cases (a) and (b), respectively.

We show experiments for the $D_b = D_f = 0$ and $D_b, D_f > 0$ cases, respectively, under nonzero T and F_b. Throughout, "O.L." represents the optimal open-loop policy. For $D_b = D_f = 0$, we perform experiments with fixed N_0, ϵ and uncontrolled MMPP parameters (Tables I and II), and varying T and F_b. Also in Table IV, we choose T and F_b fixed along with N_0, ϵ and the uncontrolled MMPP parameters, and vary D_b and D_f. In Table III, we study the effect of varying N_0 with all other parameters fixed. For small D_b, D_f, T and F_b, our algorithm converges in about 130–150 iterations, whereas for large D_b, D_f, T and F_b, it takes about 200–250 iterations. On a Sun Sparc Ultra10 work station, our algorithm takes less than 10 min in most cases to converge. The same is also true of the network case (cf. Section IV-B). We also ran the algorithm (9) of [5] for 5-level policies with no delays ($D_b = D_f = 0$). It did not converge even after 350 iterations after running close to 200 minutes, indicating that our algorithm is orders of magnitude faster than the one of [5]. We discuss our results for the single bottleneck node case in detail below.

The closed-loop solution utilizes almost the entire bandwidth ($\overline{\lambda}_c + \overline{\lambda}_u \approx \mu$) even when D_b, D_f, T, and F_b are sufficiently high. The performance degrades when the delays D_b and D_f increase, but remains better than the optimal open-loop case even when D_b and D_f become significantly high. In Table IV, performance can be seen to be better than the optimal open-loop

TABLE I
$N_0 = 10, \epsilon = 1, \overline{\lambda}_u = 0.1$

T	F_b	$\bar{q}$	P_{band}	σ_q^2	λ_c	P_{idle}	$J(\theta^*)$
1	2	10.1	0.45	7.3	0.90	0	2.0
1	5	9.5	0.34	12.8	0.90	0	2.7
1	10	10.0	0.26	21.3	0.89	0.01	3.5
1	O.L.	5.3	0.08	32.3	0.76	0.15	6.4
5	10	9.8	0.26	21.6	0.89	0.02	3.6
5	25	8.2	0.17	34.1	0.82	0.07	4.8
5	50	6.4	0.12	38.3	0.75	0.16	6.0
5	O.L.	5.1	0.09	31.9	0.75	0.16	6.4

TABLE II
$N_0 = 10, \epsilon = 1, \overline{\lambda}_u = 0.28$

T	F_b	$\bar{q}$	P_{band}	σ_q^2	λ_c	P_{idle}	$J(\theta^*)$
1	2	10.2	0.44	7.9	0.72	0	2.2
1	5	10.4	0.33	13.2	0.72	0	2.7
1	10	9.9	0.26	20.5	0.71	0.01	3.5
1	O.L.	5.3	0.08	33.2	0.60	0.13	6.5
5	10	9.2	0.26	20.5	0.70	0.02	3.6
5	25	7.3	0.16	27.9	0.64	0.07	4.8
5	50	7.3	0.12	39.4	0.59	0.14	5.6
5	O.L.	5.3	0.09	31.1	0.58	0.15	6.4

TABLE III
$T = 1, F_b = 2, \epsilon = 1, \overline{\lambda}_u = 0.1$

N_0	$\bar{q}$	P_{band}	σ_q^2	λ_c	P_{idle}	$J(\theta^*)$
2	1.5	0.55	2.3	0.55	0.34	1.3
3	2.7	0.51	4.0	0.74	0.14	1.6
4	3.8	0.48	4.9	0.83	0.07	1.8
5	4.9	0.46	5.8	0.85	0.04	1.9
6	5.7	0.46	6.4	0.87	0.02	2.0
7	6.9	0.46	6.5	0.89	0.01	2.0
8	8.2	0.44	6.9	0.89	0.01	2.0
9	8.9	0.45	7.2	0.90	0	2.0
10	10.1	0.45	7.3	0.90	0	2.0
11	11.4	0.43	7.4	0.90	0	2.1

TABLE IV
$T = 1, F_b = 2, N_0 = 10, \epsilon = 1, \overline{\lambda}_u = 0.1$

D_b	D_f	$\bar{q}$	P_{band}	σ_q^2	λ_c	P_{idle}	$J(\theta^*)$
1	0	10.1	0.38	9.7	0.89	0	2.4
1	1	9.8	0.38	9.7	0.88	0	2.4
5	5	9.7	0.26	20.0	0.88	0.01	3.6
10	10	8.8	0.20	28.0	0.87	0.03	4.5
20	10	8.4	0.14	39.5	0.84	0.06	5.2
20	40	8.2	0.13	44.0	0.80	0.06	5.4
30	20	8.1	0.13	48.0	0.79	0.08	5.7
40	10	6.4	0.12	52.7	0.78	0.12	5.8
40	30	6.9	0.12	56.4	0.78	0.12	5.9
50	50	7.1	0.12	65.2	0.78	0.10	6.0
50	100	6.7	0.11	59.5	0.77	0.13	6.1
O.L.	-	5.3	0.08	32.3	0.76	0.15	6.4

TABLE V
$T = 1, F_b = 2, N_0 = 10, \epsilon = 1, \overline{\lambda}_u = 0.1$

D_{b1}	D_{b2}	D_{f1}	D_{f2}	λ_{c1}	λ_{c2}	$J(\theta^*)$
1	10	4	10	0.44	0.45	3.6
1	30	4	10	0.44	0.44	3.9
1	50	4	20	0.42	0.43	4.4
1	80	4	100	0.41	0.42	4.7

case even for $D_b + D_f = 150$ for all performance measures except σ_q^2. The variance σ_q^2 for the open-loop case is lower possibly because of its mean queue length $\overline{q}$ being much lower than the desired mean queue length of 10. Also as expected, we get the best performance for lower values of T and F_b (see Table I and II). When the settings of the uncontrolled MMPP stream are changed such that the mean rate $\overline{\lambda}_u$ of the stream is increased, the performance degrades as expected (cf. Table I versus Table II). We also consider the case of two controllers feeding arrivals into the same bottleneck node in addition to the uncontrolled MMPP stream (Table V). Explicit rate information is fed back to the two sources with different delays D_{b1} and D_{b2}. Furthermore, there are different delays D_{f1} and D_{f2} in packets arriving to the bottleneck node from the two sources. We observe that the stationary mean rates $\overline{\lambda}_{c1}$ and $\overline{\lambda}_{c2}$ for the two sources are almost the same even when the difference in delays is significantly high. In Table III, we vary N_0 and fix other parameters with $D_b = D_f = 0$ to see the effect on performance. As expected for small N_0, $\overline{\lambda}_c$ is low, and subsequently P_{idle} is high. But as N_0 increases, $\overline{\lambda}_c$ becomes high, and P_{idle} becomes close to zero. Also, as expected, the variance σ_q^2 also increases with N_0. In the light of these observations, we discuss in the concluding section a method of finding an optimal N_0. We now discuss our experiments for networks of bottleneck nodes.

B. Network of Nodes

Here we consider two cases—a two-node network in Fig. 2 and a three-node network in Fig. 3, respectively. There are two VCs in the two-node network case (Fig. 2); the virtual path of the first ABR source passes through Nodes 1 and 2, while that of the second passes only through Node 2. Both VCs have the same destination. Similarly, in the three-node network case (Fig. 3), there are three VCs with a common destination. There are uncontrolled MMPP streams with different parameters passing through each node in both cases. We again assume, for simplicity (as in the case of a single bottleneck node), that the respective underlying Markov chain corresponding to each of these streams switches states every T units of time. In the two-node network case (Fig. 2), we assume that 40% of randomly selected uncontrolled packets from the first MMPP stream (MMPP$_1$) enter Node 2 with the remaining uncontrolled packets dropped after service at Node 1. Thus, we assume that on an average 40% of the uncontrolled traffic from Node 1 also passes through Node 2. We also make this assumption in the three-node network case (Fig. 3). We further assume, in the latter case, that 40% of randomly selected uncontrolled packets (comprising those from MMPP$_2$, as well as the ones from MMPP$_1$ that pass through Node 2), enter Node 3 after service at Node 2, while the remaining uncontrolled packets are dropped after service at Node 2. Thus, we consider very general settings here. We consider three possible Poisson rates for each MMPP stream for both the two-node and three-node network cases, respectively. Our main feedback policy (described in Section II-B) is as follows. For $i = 1, 2,$ for the two-node case (respectively, for $i = 1, 2, 3$, for the three-node case), let

$$h_i(n) = \begin{cases} \lambda^*_{i,1}, & \text{if } q_{i,n} < N_i - 2\epsilon_i \\ \lambda^*_{i,2}, & \text{if } N_i - 2\epsilon_i \le q_{i,n} < N_i - \epsilon_i \\ \lambda^*_{i,3}, & \text{if } N_i - \epsilon_i \le q_{i,n} \le N_i + \epsilon_i \\ \lambda^*_{i,4}, & \text{if } N_i + \epsilon_i < q_{i,n} \le N_i + 2\epsilon_i \\ \lambda^*_{i,5}, & \text{if } q_{i,n} > N_i + 2\epsilon_i. \end{cases} \qquad (13)$$

In the above, $q_{i,n}$ is the queue length at Node i. Also, in the above, N_i and ϵ_i are similar to N_0 and ϵ, respectively, for the single bottleneck case. In all our experiments here, we let $\epsilon_i = 1$ (for simplicity) for all nodes. Let $\lambda_{c,i}(n)$ represent the ABR rate for source i. Then, for the two-node case, we set $\lambda_{c,2}(n) = h_2(n)/2$, $\lambda_{c,1}(n) = \min(h_1(n), h_2(n)/2)$. The ABR rates in the three-node network case are determined as follows: $\lambda_{c,3}(n) = h_3(n)/3$, $\lambda_{c,2}(n) = \min(h_2(n)/2, h_3(n)/3)$, $\lambda_{c,1}(n) = \min(h_1(n), \lambda_{c,2}(n))$. In what follows, we refer to our main policy simply as Policy (13) for convenience. Next, let us briefly explain the ERICA algorithm [16]. ERICA computes the available bandwidth ABR$_{BW}$ according to ABR$_{BW}$ = Target utilization × (Link Capacity − (VBR + CBR)$_{BW}$). Target utilization is typically set at 0.90–0.95. We choose this quantity to be 0.95 in our experiments. If there are L VCs using a link at any time, ERICA computes the Fairshare of each VC according to Fairshare = ABR$_{BW}/L$. Next, ERICA computes VC$_{\text{share}}$ = CCR$/z$, where CCR is the source current cell rate stored in the resource management (RM) cells and z is the load factor computed as z = ABR$_{\text{Inputrate}}$/ABR$_{BW}$. The optimal operating point is $z = 1$. Finally, the explicit rate (ER) marked by the switch for the VC is calculated as follows: ER$_{\text{calculated}}$ $\leftarrow$ $\min(\max($Fairshare, VC$_{\text{share}})$, ABR$_{BW}$, ER$_{\text{in RM cell}})$.

We also provide here an alternative policy (based on ERICA) that uses quantities $h_i(n)$ in (13) (for bottleneck node i) in place of ABR$_{BW}$ with the rest of the algorithm the same as in ERICA. Note that a key advantage in so doing is that one does not require estimation of ABR$_{BW}$ (as ERICA does and which is very difficult to obtain in practice), and instead, one directly observes the queue lengths at the nodes as in Policy (13). Moreover, we use algorithm (10) [as in Policy (13)] to find the optimum parameter in the class of ERICA-type policies as well. We call the resulting optimal policy (in this class) ERICA$_{\text{opt}}$.

For the case of two-node (respectively, three-node) networks, we assume a similar parameter constraint region as for the single bottleneck node case, *viz.* $\lambda_{i,1}, \ldots, \lambda_{i,4} \in$ [0.10, 3.0] and $\lambda_{i,5} \in$ [0.10, 0.90], $i = 1, 2$, for two-node (resp. $i = 1, 2, 3$, for three-node) networks. We assume for simplicity in the ERICA implementation that the MMPP streams comprise only CBR and VBR traffic (and not ABR from other sources). Note that ERICA does not require thresholds N_i. For the two-node case, using Policy (13) and ERICA$_{\text{opt}}$, we perform experiments for two different cases: $N_1 = 7$, $N_2 = 30$ (Tables VI and VIII) and $N_1 = 10$, $N_2 = 25$ (Tables VII and IX), respectively. We also set the service times as $\mu_1 = 1.0$ and $\mu_2 = 1.5$, respectively, for the two nodes in the two-node case. The MMPP stream parameters are selected as follows. Let $\Lambda_1 \triangleq (\lambda^1_1, \lambda^1_2, \lambda^1_3)^T = (0.2, 0.4, 0.6)^T$ and $\Lambda_2 \triangleq (\lambda^2_1, \lambda^2_2, \lambda^2_3)^T = (0.1, 0.2, 0.3)^T$ represent two different vectors of uncontrolled rates. Let $\pi = (0.3, 0.4, 0.3)^T$ represent the vector of stationary probabilities. When MMPP stream (Λ_1, π) is used, the mean uncontrolled rate is 0.4 and when the other stream (Λ_2, π) is used, the same is 0.2. In Tables VI–X, we use various combinations of these streams in the two nodes. We represent the uncontrolled traffic simply by the mean rates $\overline{\lambda}_{u1}$ and $\overline{\lambda}_{u2}$ at Node 1 and Node 2, respectively, in Fig. 2. We arbitrarily set the delays in the two-node case as follows: $D_{b1} = 10$, $D_{b2} = 1$, $D_{f1} = 20$, $D_{f2} = 10$ and $D^{12}_f = 10$ (see Fig. 2). For the three-node case, in Tables XI–XIII, we let both MMPP$_1$ and MMPP$_2$ be (Λ_1, π), and for MMPP$_3$ we consider two different scenarios (Λ_3, π^1) and (Λ_4, π^2), respectively, where $\Lambda_3 \triangleq (\lambda^3_1, \lambda^3_2, \lambda^3_3)^T = (0.1, 0.3, 0.65)^T$, $\pi^1 = (0.2, 0.3, 0.5)^T$, and $\Lambda_4 \triangleq (\lambda^4_1, \lambda^4_2, \lambda^4_3)^T = (0.05, 0.2, 0.4)^T$, $\pi^2 = (0.3, 0.5, 0.2)^T$. Thus, $\overline{\lambda}_{u3}$ is either 0.435 or 0.195. We arbitrarily set $N_1 = 7$, $N_2 = 10$ and $N_3 = 25$ for the three-node case. We also vary the service rate for Node 3 in Tables XI–XIII

TABLE VI

[POLICY (13)]: $T = 1, N_1 = 7, N_2 = 30$

λ_{u1}	λ_{u2}	λ_{c1}	λ_{c2}	MV
0.2	0.2	0.44	0.86	9.27
0.2	0.4	0.50	0.80	12.71
0.4	0.4	0.50	0.55	21.66
0.4	0.2	0.50	0.80	12.72

TABLE VII

[POLICY (13)]: $T = 1, N_1 = 10, N_2 = 25$

λ_{u1}	λ_{u2}	λ_{c1}	λ_{c2}	MV
0.2	0.2	0.62	0.67›	12.12
0.2	0.4	0.39	0.71	9.19
0.4	0.4	0.37	0.73	9.90
0.4	0.2	0.52	0.78	12.53

TABLE VIII

(ERICA$_{\text{opt}}$): $T = 1, N_1 = 7, N_2 = 30$

λ_{u1}	λ_{u2}	λ_{c1}	λ_{c2}	MV
0.2	0.2	0.55	0.74	11.58
0.2	0.4	0.46	0.64	10.91
0.4	0.4	0.33	0.77	8.89
0.4	0.2	0.35	0.95	8.94

TABLE IX

(ERICA$_{\text{opt}}$): $T = 1, N_1 = 10, N_2 = 25$

λ_{u1}	λ_{u2}	λ_{c1}	λ_{c2}	MV
0.2	0.2	0.50	0.79	10.0
0.2	0.4	0.40	0.49	11.0
0.4	0.4	0.44	0.66	11.40
0.4	0.2	0.50	0.80	12.37

TABLE X

(ERICA): $T = 1$

λ_{u1}	λ_{u2}	λ_{c1}	λ_{c2}	MV
0.2	0.2	0.62	0.63	31.26
0.2	0.4	0.55	0.55	35.96
0.4	0.4	0.47	0.59	38.18
0.4	0.2	0.52	0.72	26.11

TABLE XI
[POLICY (13)]: $T = 1$

λ_{u3}	μ_3	λ_{c1}	λ_{c2}	λ_{c3}	MV
0.435	1.8	0.39	0.42	0.54	6.86
0.435	2.0	0.41	0.50	0.65	7.42
0.435	2.2	0.44	0.54	0.78	7.81
0.195	1.8	0.38	0.48	0.74	6.32
0.195	2.0	0.25	0.67	0.88	8.66
0.195	2.2	0.47	0.55	0.99	8.12

TABLE XII
(ERICA$_{\text{opt}}$): $T = 1$

λ_{u3}	μ_3	λ_{c1}	λ_{c2}	λ_{c3}	MV
0.435	1.8	0.36	0.45	0.55	6.80
0.435	2.0	0.40	0.47	0.68	8.30
0.435	2.2	0.41	0.48	0.87	7.11
0.195	1.8	0.42	0.51	0.67	8.12
0.195	2.0	0.44	0.54	0.82	7.70
0.195	2.2	0.43	0.58	0.99	7.51

TABLE XIII
(ERICA): $T = 1$

λ_{u3}	μ_3	λ_{c1}	λ_{c2}	λ_{c3}	MV
0.435	1.8	0.40	0.43	0.48	28.83
0.435	2.0	0.43	0.48	0.58	27.55
0.435	2.2	0.45	0.51	0.71	26.04
0.195	1.8	0.44	0.49	0.60	24.76
0.195	2.0	0.45	0.53	0.73	25.52
0.195	2.2	0.46	0.56	0.89	28.39

TABLE XIV
[POLICY (13)]: $T = 1$

Delays	λ_{c1}	λ_{c2}	λ_{c3}	MV
0	0.46	0.48	0.62	5.92
1.0	0.47	0.50	0.59	8.01
2.0	0.47	0.49	0.60	9.70
3.0	0.46	0.48	0.62	10.39
4.0	0.47	0.50	0.59	12.04
5.0	0.45	0.54	0.58	13.02

TABLE XV
(ERICA$_{\text{opt}}$): $T = 1$

Delays	λ_{c1}	λ_{c2}	λ_{c3}	MV
0	0.48	0.50	0.58	7.14
1.0	0.51	0.48	0.57	9.33
2.0	0.46	0.46	0.65	9.60
3.0	0.46	0.46	0.64	10.44
4.0	0.48	0.49	0.59	12.00
5.0	0.45	0.51	0.60	12.88

TABLE XVI
(ERICA): $T = 1$

Delays	λ_{c1}	λ_{c2}	λ_{c3}	MV
0	0.49	0.50	0.52	41.57
1.0	0.52	0.50	0.51	42.34
2.0	0.52	0.52	0.52	147.56
3.0	0.52	0.52	0.52	239.07
4.0	0.52	0.52	0.52	183.81
5.0	0.52	0.52	0.52	248.60

(with all delays zero) between three different values *viz.* $\mu_3 =$ 1.8, 2.0 and 2.2, respectively. In Tables XIV–XVI, we vary delays between 1.0 and 5.0 (here a delay of x implies that each of the delays D_{f1}, D_{b1}, D_f^{12}, etc. are equal to x), when sampling time in all cases is $T = 1$. Also, here we let the uncontrolled streams MMPP$_1$ and MMPP$_2$ be (both) (Λ_2, π) with a mean rate $\overline{\lambda}_{u1} = \overline{\lambda}_{u2} = 0.2$, and MMPP$_3$ be (Λ_3, π^1) (with $\overline{\lambda}_{u3} = 0.435$). We compare the performance of Policy (13) and ERICA$_{\text{opt}}$ using our algorithm, with that of ERICA in terms of ABR rate allocations of each scheme and the variation in the resulting queue length processes as follows. Let $\overline{\lambda}_{ck}$, $k = 1, 2$, in Fig. 2 (resp. $k = 1, 2, 3$, in Fig. 3) represent the mean ABR rate of the kth ABR source in Fig. 2 (resp. Fig. 3). These are defined in exactly the same manner as $\overline{\lambda}_c$ for the single bottleneck node case in Section IV-A. Also, let $\sigma_{q,\,i}^2$ represent the variance of queue length in ith queue defined in the same manner as σ_q^2 in Section IV-A. We define the measure of variation (MV) in queue lengths as follows: MV $= (\sigma_{q,\,1}^2 + \sigma_{q,\,2}^2)^{1/2}$ for the system in Fig. 2, and MV $= (\sigma_{q,\,1}^2 + \sigma_{q,\,2}^2 + \sigma_{q,\,3}^2)^{1/2}$ for the system in Fig. 3. We select the cost function now as $\hat{h}(q_{1,\,n}, q_{2,\,n}) = (|q_{1,\,n} - N_1| + |q_{2,\,n} - N_2|)/2$ for the two-node case and $\hat{h}(q_{1,\,n}, q_{2,\,n}, q_{3,\,n}) = (|q_{1,\,n} - N_1| + |q_{2,\,n} - N_2| + |q_{3,\,n} - N_3|)/3$ for the three-node case. We choose $\delta = 0.12$ as earlier. The step-sizes $\{a(n)\}$ and $\{b(n)\}$ are as before. The random variables $\Delta_{m,\,i}$ are assumed to be symmetric Bernoulli distributed as in the single bottleneck node case.

For the two-node network (Fig. 2), for most cases, the mean ABR rate for the first source ($\overline{\lambda}_{c1}$) using both Policy (13) and ERICA$_{\text{opt}}$ is lower than that for ERICA, while that for the second source ($\overline{\lambda}_{c2}$) for these policies is higher than that for ERICA. For the three-node network case, a similar behavior is observed, except that the difference between corresponding source rates using both Policy (13) and ERICA$_{\text{opt}}$, with that of ERICA is much lower now. However, our key observation here is that the measure of variation of the queue length processes for all two-node and three-node cases, using Policy (13) and ERICA$_{\text{opt}}$ is many times lower (almost an order of magnitude in some cases) than that using ERICA. We observe in Tables XIV–XVI that, in the presence of delays, the measure of variation using our policies does not degrade as much as ERICA does. ERICA is known to exhibit highly oscillatory behavior and our experiments seem to confirm this fact. Our algorithm, on the other hand, is much more robust to delays and variations in other parameters. This fact is further illustrated in Figs. 4 and 5. We observe in the experiments for the three-node network shown in Tables XIV–XVI that all three policies [Policy (13), ERICA$_{\text{opt}}$, and ERICA] have a similar mean queue length in the first two nodes. In the first node, the mean queue length for all three policies is in the range 2.1–3.5. In the second node, the same is in the range 3.6–5.6. In the third node, the mean queue length using Policy (13) and ERICA$_{\text{opt}}$ is in the range 24–27, while that using ERICA is 43.5 with zero delays and becomes 245 when all delays are equal to five. A similar deterioration in the variance performance for the third queue is observed using ERICA, which is the reason for the high MV using this scheme.

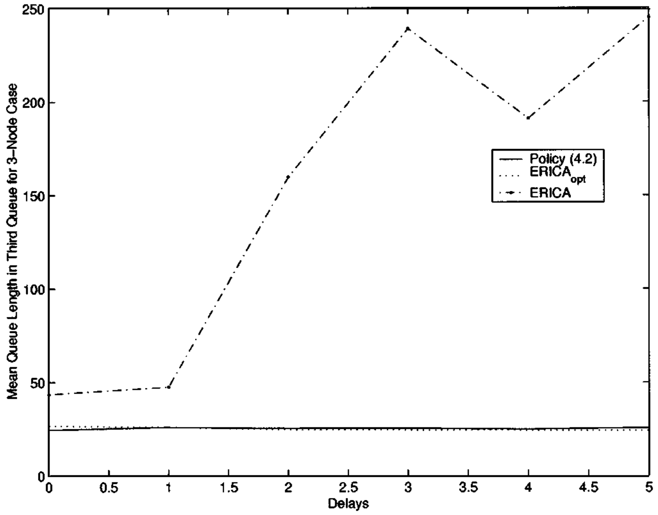

Fig. 4. Comparison of mean queue lengths using all Policies in the third node in the network in Fig. 3.

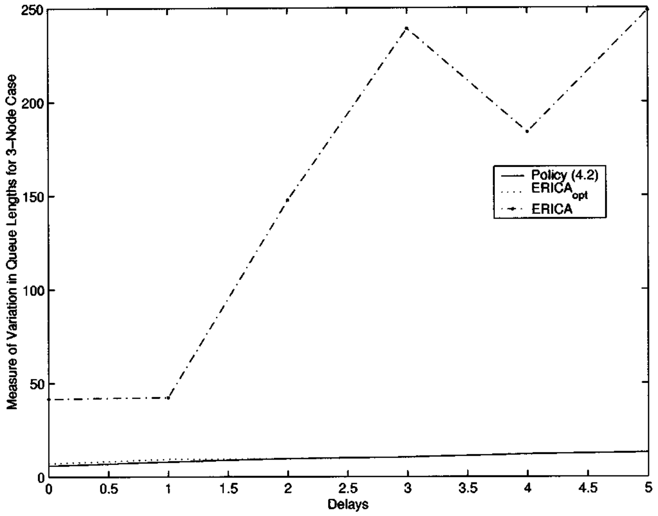

Fig. 5. Comparison of measure of variation performance using all Policies in the third node in the network in Fig. 3.

V. Conclusion and Extensions

We studied the problem of ABR rate-based flow control in the presence of information and propagation delays, by developing a numerically efficient two-timescale SPSA algorithm. In particular, we studied two models, the single bottleneck node and a network of such nodes. The convergence of this algorithm was theoretically proven, and numerical experiments were conducted to investigate the performance of the structured feedback policies. For the case of a single bottleneck node, multilevel closed-loop feedback policies were compared with optimal open-loop policies and were found to perform better than the latter even in the presence of significant delays. We observed that our algorithm converges orders of magnitude faster than the algorithm of [5]. We also performed experiments with two and three node networks. We compared the performance of our structured policy [Policy (13)] with ERICA. Further, using our techniques, we also obtained another "optimal" structured policy (ERICA$_{\rm opt}$) that is based on ERICA, which has the advantage (over ERICA) that it does not require an estimate of available bandwidth using it. We observed that even though ERICA is "fairer" in some cases than Policy (13) and ERICA$_{\rm opt}$; however, in all the cases we considered, both Policy (13) and ERICA$_{\rm opt}$ have a measure of variation in queue length processes that is many times lower than ERICA.

One natural extension of this work is to apply similar methods for selecting thresholds N_0 and N_i. For instance, for the single bottleneck node case, by incorporating N_0 into the parameter vector θ, i.e., θ is now represented as $\theta = (\lambda_1, \ldots, \lambda_N, N_0)^T$, an optimal N_0 can be determined by optimizing θ as earlier. However, note that if we continue with the same form of the cost function $h(q_n) = |q_n - N_0|$, then this would in fact give rise to a family of parameterized cost functions (parameterized by N_0). Table III in our numerical experiments suggests choosing a band $[a, b]$ (depending upon acceptable levels of performance) within which one can expect N_0 to lie. One can then select a cost function that takes (say) value zero on $[a, b]$ and increases sharply outside. The two-timescale SPSA algorithm (10) applied to suitable parameterized policies similar to (2) with the parameter $\theta = (\lambda_1, \ldots, \lambda_N, N_0)^T$ can then give rise to an optimal N_0 within that class of policies. The only problem is with the fact that N_0 is continuous-valued now. However, one can use the integral part of N_0-updates in the policy as an approximation. A similar approach can also be used for finding optimal N_i for the case of network of nodes.

Finally, we mention an open problem here. The problem is to prove Theorem A.1 (see the Appendix) for the system in Fig. 1 with i.i.d. general service times (we assumed exponential distribution) and with a finite or an infinite buffer. The rest of the convergence analysis for such a system can be shown as remarked at the end of Theorem A.1.

Appendix
Convergence Analysis

We show the convergence analysis for the single bottleneck node case. The analysis for the network case follows in a similar manner with minor changes in the proof of Theorem A.1. However, due to considerations of space, some of the proofs have been shortened, with the details found in [6]. We assume for simplicity here that the service time process is i.i.d. with exponential distribution. This assumption is however only required in the proof of Theorem A.1 (below), which along with Corollary A.1 establishes the preliminary hypotheses for convergence of algorithm (10). The remark at the end of Theorem A.1 explains the difficulty with the general service time case.

When $D_b = D_f = 0$, the rate $\lambda_c(n)$ becomes effective in the time interval $[nT, (n+1)T)$. Then, under the type of policies (2), it is clear that $\{(q_n, X_n)\}$, $n \geq 0$, is a Markov chain. When D_b, D_f are nonzero, we will assume for simplicity that $D_b + D_f = MT$ for some integer $M > 0$. In this case, the ABR rate $\lambda_c(n)$ computed at time nT at the node is in fact effective in the time interval $[(n+M)T, (n+M+1)T)$. Thus, in the interval $[nT, (n+1)T)$, packets from the ABR source that arrive at the node were in fact sent from the source with rate $\lambda_c(n-M)$ computed at time $(n-M)T$ at the node. For such a system, it can be seen that the joint process $\{(q_n, X_n, q_{n-1}, X_{n-1}, \ldots, q_{n-M}, X_{n-M})\}$, $n \geq 0$ is a

Markov chain. In a related paper [1], it is shown that a system as in Fig. 1 with (say) $D_b + D_f = MT$ is equivalent to one with $D_f = 0$ and $D_b = MT$. We define a Markov process that is aperiodic, irreducible and positive recurrent to be ergodic. It is easy to see that, since we have a finite buffer system and because $\{X_n\}$ is ergodic, for $D_b = D_f = 0$, the joint process $\{(q_n, X_n)\}$, under policies (2) is ergodic. Similarly for the delayed case (when $D_b + D_f = MT$), the joint process $\{(q_n, X_n, q_{n-1}, X_{n-1}, \ldots, q_{n-M}, X_{n-M})\}$ under policies (2) is ergodic as well. For ease of exposition, we consider the case $D_b = D_f = 0$ in detail from now on and explain the changes necessary for nonzero D_b, D_f as we proceed. Thus, for any given θ, $\{(q_n, X_n)\}$ is ergodic Markov with $\{\lambda_c(n)\}$ as in (2). Let $\mu_\theta(q, x)$ be the stationary distribution of this Markov chain on $S \times S_u$, for given $\theta \in C$. Let $\nu_\theta(q)$ be the marginal of $\mu_\theta(q, x)$ on S that corresponds to the stationary distribution of $\{q_n\}$ alone. Thus, $\nu_\theta(q) = \sum_{x \in S_u} \mu_\theta(q, x)$. The average cost $J(\theta)$ in (1) can now be written as $J(\theta) = \sum_{i \in S} h(i)\nu_\theta(i) = \sum_{i \in S} \sum_{x \in S_u} h(i)\mu_\theta(i, x)$.

Next, we establish some preliminary hypotheses necessary to prove Theorem 3.1. Let $p_\theta(i, x;\; i', x')$, $i, i' \in S$, $x, x' \in S_u$ represent the transition probabilities for the Markov chain $\{(q_n, X_n)\}$ for given θ. Let D_n denote the number of departures from the queue in the time interval $[nT, (n+1)T)$, A_n^c denote the number of arrivals from the controlled source in $[nT, (n+1)T)$ and A_n^u be the number of arrivals from the uncontrolled stream during the same time interval.

Theorem A.1: Under all policies of type (2), $J(\theta)$ is continuously differentiable in θ.

Proof: When $D_b = D_f = 0$, for $J(\theta)$ to be continuously differentiable, it is enough to show that $\mu_\theta(\cdot, \cdot)$ is continuously differentiable in θ. For ease of exposition, let us consider for the moment that θ is a scalar. Writing in matrix notation, let for fixed θ, $P(\theta) := [[p_\theta(i, x;\; j, y)]]$ be the transition probability matrix of $\{(q_n, X_n)\}$ and $\mu(\theta) := [\mu_\theta(i, x)]$ denote the vector of stationary probabilities. Also let $Z(\theta) := [I - P(\theta) - P^\infty(\theta)]^{-1}$, where I is the identity matrix and $P^\infty(\theta) = \lim_{m\to\infty}(P(\theta) + \cdots + P^m(\theta))/m$. It is shown in [24] that $P^\infty(\theta)$ and $Z(\theta)$ exist for all finite-dimensional Markov chains. Also, from [24, Th. 2], we can write

$$\mu(\theta + h) = \mu(\theta)(I + (P(\theta + h) - P(\theta))Z(\theta) + o(h)). \quad (14)$$

Thus, $\mu'(\theta) = \mu(\theta)P'(\theta)Z(\theta)$. The proof of existence and continuity of the derivative $P'(\theta)$ (of $P(\theta)$) is straightforward (see [6]). Thus, the derivative $\mu'(\theta)$ [of $\mu(\theta)$] exists. We also have

$$\begin{aligned}|\mu'(\theta + h) - \mu'(\theta)| &\\ \le |\mu(\theta + h)P'(\theta + h)Z(\theta + h) - \mu(\theta)P'(\theta + h)Z(\theta + h)| &\\ - |\mu(\theta)P'(\theta + h)Z(\theta + h) - \mu(\theta)P'(\theta)Z(\theta + h)| &\\ - |\mu(\theta)P'(\theta)Z(\theta + h) - \mu(\theta)P'(\theta)Z(\theta)|. &\end{aligned}$$

Now, from [24, Th. 2], we can write $Z(\theta + h)$ as $Z(\theta + h) = Z(\theta)H(\theta, \theta + h) - P^\infty(\theta)H(\theta, \theta + h)U(\theta, \theta + h)Z(\theta)H(\theta, \theta + h)$, where, $H(\theta, \theta + h) = [I - (P(\theta + h) - P(\theta))]^{-1} \to I$ as $|h| \to 0$ and $U(\theta, \theta + h) = (P(\theta + h) - P(\theta))Z(\theta) \to \bar{0}$ as $|h| \to 0$. In the above $\bar{0}$ is the matrix (of appropriate dimension) with all zero elements. It thus follows that $Z(\theta + h) \to Z(\theta)$ as $|h| \to 0$. Moreover from (14), $\mu(\theta)$ is continuous. Thus from above, $\mu'(\theta)$ is continuous in θ and the claim follows. For vector θ, a similar proof as above verifies the claim. The proof for the delayed case $D_b, D_f \neq 0$ follows in a similar manner. □

Remark: Let V_n represent the work load at instant nT. For a system as in Fig. 1 (but) with general i.i.d. service times, under policies (2), $\{(V_n, X_n)\}$ for $D_b = D_f = 0$, is ergodic Markov. Similarly, $\{(V_n, X_n, V_{n-1}, X_{n-1}, \ldots, V_{n-M}, X_{n-M})\}$ is ergodic Markov for $D_b + D_f = MT$. The above Markov chains are however uncountable and [24, Th. 2] (which holds for a finite state system) is no longer valid. However, Corollary A.1 below can be shown easily for this system using sample path arguments [4]. Moreover, the remainder of the analysis can be shown in a manner similar to that below. Thus, the problem of proving Theorem A.1 for i.i.d. general service times remains an open problem. As a first step, one can try to prove this result for more general distributions, for instance, mixture of sums of exponentials (in a similar manner as above). Moreover, as mentioned in Section IV-A, in [4] more general service time distributions were tried on a similar model with uncontrolled arrival stream Poisson and similar experimental results as for exponential service times were obtained. We conjecture that Theorem A.1 holds for general service distributions as well.

Corollary A.1: Under all policies (2), the map $\theta \to p_\theta(i, x_1;\; j, x_2)$ is continuous in θ.

Proof: The above map is differentiable in θ (see above), and hence continuous. □

We now proceed with the rest of the convergence analysis. Let $\mathcal{F}_n \overset{\Delta}{=} \sigma(q_j^1, q_j^2, X_j^1, X_j^2, \tilde{\theta}_j, \tilde{\Delta}_j, 1 \le j \le n)$ represent the σ-algebra associated with information up to period nT and where $\tilde{\theta}_j = \theta(m)$ and $\tilde{\Delta}_j = \Delta(m)$, for $n_m \le j < n_{m+1}$. We consider the undelayed case ($D_b = D_f = 0$) here. The delayed case is treated in detail in [6]. For any sets $A \subset S$ and $D \subset S_u$, define sequences $\{M_{1,n}(A \times D)\}$ and $\{M_{2,n}(A \times D)\}$ as follows. For $k = 1, 2$

$$\begin{aligned}M_{k,n}(A \times D) &= \sum_{m=0}^{n-1} b(m)^{-1} \left[\sum_{j=n_m+1}^{n_{m+1}} a(j)(I\{q_j^k \in A, X_j^k \in D\} \right.\\ &\qquad \left. - E[I\{q_j^k \in A, X_j^k \in D\} \,|\, \mathcal{F}_{j-1}]) \right]\end{aligned}$$

and $n \ge 1$, where $I\{\cdot\}$ is the indicator or characteristic function. One can then proceed to show (see [6] for details) that $\{M_{1,n}(A \times D)\}$ and $\{M_{2,n}(A \times D)\}$ are zero mean, square integrable martingale sequences with a.s. convergent quadratic variation processes. Thus, we have the following.

Lemma A.1: Given $A \subset S$ and $D \subset S_u$, $\{M_{1,n}(A \times D)\}$, and $\{M_{2,n}(A \times D)\}$ converge a.s.

Proof: Follows from [20, Proposition VII.2.3(c), pp. 149–150]. □

Now for $m \geq 1$, define random variables $\{\mu_m^1\}$ and $\{\mu_m^2\}$ as follows: Let $A \subset S$ and $D \subset S_u$ be any two sets. Then,

$$\mu_m^k(A \times D) = \frac{\sum_{j=n_m+1}^{n_{m+1}} a(j) I\{q_j^k \in A,\, X_j^k \in D\}}{\sum_{j=n_m+1}^{n_{m+1}} a(j)}, \quad k = 1,\, 2.$$

Recall that $\mu_\theta(i,\, x)$ represents the invariant measure corresponding to the ergodic Markov process $\{(q_n,\, X_n)\}$ that has transition probabilities $[[p_\theta(i,\, x;\; j,\, y)]]$. Then $\mu_{\pi(\theta-\delta\Delta)}\;(i,\, x)$ [resp. $\mu_{\pi(\theta+\delta\Delta)}\;(i,\, x)$] is the invariant measure of the ergodic Markov process $\{(q_n,\, X_n)\}$ that has transition probabilities $[[p_{\pi(\theta-\delta\Delta)}\;(i,\, x;\; j,\, y)]]$ (resp. $[[p_{\pi(\theta+\delta\Delta)}\;(i,\, x;\; j,\, y)]]$).

Theorem A.2: Almost surely, $(\mu_m^1,\, \mu_m^2,\, \theta(m),\, \Delta(m))$, $m \geq 0$, converges to the compact set $\{(\mu_{\pi(\theta-\delta\Delta)},\, \mu_{\pi(\theta+\delta\Delta)},\, \theta,\, \Delta) \mid \theta \in C,\, \Delta \in E\}$.

Proof: Let us consider $\{M_{1,\,n}(A \times D)\}$ first. From Lemma A.1 and the fact that $\sum_{j=n_m+1}^{n_{m+1}} a(j)/b(m) \rightarrow 1$ as $m \rightarrow \infty$, one has

$$\left.\sum_{j=n_m+1}^{n_{m+1}} \left[a(j)[I\{q_j^1 \in A,\, X_j^1 \in D\} - E[I\{q_j^1 \in A,\, X_j^1 \in D\} \,|\, \mathcal{F}_{j-1}]]\right] \right/ \left[\sum_{j=n_m+1}^{n_{m+1}} a(j)\right] \rightarrow 0 \text{ a.s.},$$

for any $A \subset S$ and $D \subset S_u$. Hence

$$\left.\sum_{j=n_m+1}^{n_{m+1}} \left[a(j)\left[I\{q_j^1 \in A,\, X_j^1 \in D\} - \sum_{i \in S} \sum_{x \in S_u} p_{\pi(\theta(m)-\delta\Delta(m))}(i,\, x;\; A,\, D) \cdot I\{q_{j-1}^1 = i,\, X_{j-1}^1 = x\}\right]\right] \right/ \left[\sum_{j=n_m+1}^{n_{m+1}} a(j)\right] \rightarrow 0 \text{ a.s.} \tag{15}$$

Now any limit point of $\{(\mu_m^1,\, \pi(\theta(m) - \delta\Delta(m)))\}$ must be of the form $(\mu,\, \pi(\theta - \delta\Delta))$. From (15), Corollary A.1, the definition of $\{\mu_m^1\}$ and (5), it follows that μ must satisfy $\mu(A \times D) = \sum_{i \in S} \sum_{x \in S_u} p_{\pi(\theta-\delta\Delta)}(i,\, x;\; A,\, D)\mu(i,\, x)$. Thus $\mu = \mu_{\pi(\theta-\delta\Delta)}$. An analogous argument applies to $\{\mu_m^2\}$. The claim now follows since any continuous image of a compact set is compact. □

The proof of Theorem 3.1 proceeds through a sequence of steps and is given in detail in [6]. We sketch it below very briefly.

Proof of Theorem 3.1: (sketch) One proceeds through a series of approximation steps, first noting as a consequence of Theorem A.2 and the fact that $\sum_{i=n_m+1}^{n_{m+1}} a(i)/b(m) \rightarrow 1$ as $m \rightarrow \infty$, that the algorithm (10) can be written as

$$\begin{aligned} &\lambda_i(m+1) \\ &\quad = \pi_i(\lambda_i(m) + (b(m) \\ &\qquad \cdot [J(\pi(\theta(m) - \delta\Delta(m))) - J(\pi(\theta(m) + \delta\Delta(m)))] \\ &\qquad /[2\delta\Delta_{m,\,i}]) + b(m)\xi_i(m)) \end{aligned}$$

with $i = 1,\, \ldots,\, N$, where $\xi_1(m),\, \ldots,\, \xi_N(m)$ are asymptotically diminishing error terms. The second term on the RHS above can then be replaced by its conditional expectation taken w.r.t. $\sigma(\theta(1),\, \ldots,\, \theta(m))$, the sigma field generated by $\theta(1),\, \ldots,\, \theta(m)$, since the difference of these terms forms a martingale difference sequence with the martingale thus obtained being almost surely convergent. Now, $\Delta(m)$ is independent of $\sigma(\theta(1),\, \ldots,\, \theta(m))$ for all m. Hence, using appropriate Taylor series expansions of $J(\pi(\theta(m) - \delta\Delta(m)))$ and $J(\pi(\theta(m) + \delta\Delta(m)))$ around the point $\theta(m)$, one obtains the claim under Condition A) in the limit as $\delta \downarrow 0$, with the terms corresponding to components other than i averaging to zero. We refer the reader to [6] for the detailed proof. □

References

[1] E. Altman, T. Basar, and R. Srikant, "Robust rate control for ABR sources," in *Proc. INFOCOM'98*, San Francisco, CA, Mar. 1998, pp. 166–173.

[2] J. D. Bartusek and A. M. Makowski. (1994) On stochastic approximation driven by sample averages: Convergence results via the ODE method, Tech. Rep.. Institute for Systems Research, Univ. of Maryland, College Park. [Online]. Available: http://www.isr.umd.edu/TechReports/ISR/1994/TR_94-4/

[3] 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, pp. 693–707, Dec. 1993.

[4] S. Bhatnagar, "Multiscale stochastic approximation schemes with applications to ABR service in ATM networks," Ph.D. dissertation, Dept. Elect. Eng., Indian Institute of Science, Bangalore, July 1997.

[5] S. Bhatnagar and V. S. Borkar, "Multiscale stochastic approximation for parametric optimization of hidden Markov models," *Prob. Engg. and Info. Sci.*, vol. 11, pp. 509–522, 1997.

[6] S. Bhatnagar, M. C. Fu, and S. I. Marcus. (1999) Optimal multilevel feedback policies for ABR flow control using two timescale SPSA, Tech. Rep.. Institute for Systems Research, Univ. of Maryland, College Park. [Online]. Available: http://www.isr.umd.edu/TechReports/ISR/1999/TR_99-18/

[7] S. Bhatnagar, M. C. Fu, S. I. Marcus, and S. Bhatnagar, "Two-timescale algorithms for simulation optimization of hidden Markov models," IIE Trans., vol. 33, no. 3, pp. 245–258, 2001, to be published.

[8] F. Bonomi and K. W. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, pp. 25–39, 1995.

[9] A. Charny, D. D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Proc. ICC*, June 1995, pp. 1954–1963.

[10] F. M. Chiussi, Y. Xia, and V. P. Kumar, "Dynamic max rate control algorithm for available bit rate service in ATM networks," in *Proc. IEEE Globecom*, Nov. 1996, pp. 2108–2117.

[11] E. K. P. Chong and P. J. Ramadge, "Stochastic optimization of regenerative systems using infinitesimal perturbation analysis," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1400–1410, 1994.

[12] M. C. Fu, "Convergence of a stochastic approximation algorithm for the $GI/G/1$ queue using infinitesimal perturbation analysis," *J. Optim. Theory Appl.*, vol. 65, pp. 149–160, 1990.

[13] M. C. Fu, "Optimization via simulation: A review," *Ann. Oper. Res.*, vol. 53, pp. 199–248, 1994.

[14] M. C. Fu and S. D. Hill, "Optimization of discrete event systems via simultaneous perturbation stochastic approximation," *IIE Trans.*, vol. 29, no. 3, pp. 233–243, 1997.

[15] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Digital Equipment Corporation, MA, Tech. Rep. DEC-TR-506, Aug. 1987.

[16] 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, pp. 87–98, Feb. 2000.

[17] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*: Springer–Verlag, 1997.

[18] S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle," *Automatica*, vol. 35, pp. 1921–1935, 1999.

[19] S. Mascolo, D. Cavendish, and M. Gerla, "ATM rate-based congestion control using a Smith predictor: An EPRCA implementation," in *Proc. IEEE INFOCOM'96*, Mar. 1996, pp. 569–576.

[20] J. Neveu, *Discrete Parameter Martingales*. Amsterdam, The Netherlands: North Holland, 1975.

[21] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.

[22] L. Roberts, "Enhanced proportional rate control algorithm (PRCA)," ATM forum contribution 94-0735R1, Aug. 1994.

[23] L. Roberts, B. Makrucki, T. Tofigh, A. Barnhart, B. Holden, G. Fedorkow, J. Daigle, M. Hluchyj, H. Suzuki, G. Ramamurthy, P. Newman, N. Giroux, R. Kositpaiboon, S. Sathe, G. Garg, and N. Yin, "Closed loop rate-based traffic management," ATM Forum contribution 94-0438R1, July 1994.

[24] P. J. Schweitzer, "Perturbation theory and finite Markov chains," *J. Appl. Prob.*, vol. 5, pp. 401–413, 1968.

[25] V. Sharma and J. Kuri, "Stability and performance of rate-based feedback flow controlled ATM networks," *Queueing Syst.*, vol. 29, pp. 129–159, 1998.

[26] K. Y. Siu and H. Y. Tzeng, "Intelligent congestion control for ABR service in ATM networks," in *ACM SIGCOMM Comput. Commun. Rev.*, 1995, pp. 81–106.

[27] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 332–341, 1992.

[28] Y. T. Wang and B. Sengupta, "Performance analysis of a feed back congestion control policy," in *ACM SIGCOMM Comput. Commun. Rev.*, 1991, pp. 149–157.

Shalabh Bhatnagar received the Bachelors degree (Hons.) in physics from the University of Delhi, Delhi, India, in 1988, and the Masters and Ph.D. degrees in electrical engineering from the Indian Institute of Science, Bangalore, India, in 1992 and 1998, respectively.

From October 1997 to July 2000, he was a Research Associate at the Institute for Systems Research, University of Maryland, College Park. He was a Divisional Postdoctoral Fellow at the Division of Mathematics and Computer Science, Free University, Amsterdam, The Netherlands. He is currently with the Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi, India. His research interests include stochastic approximation algorithms for optimization of discrete event systems and control of communication networks.

Dr. Bhatnagar is on the Review Committee of many journals and conferences, and in 1999 was recognized by the IEEE TRANSACTIONS ON AUTOMATIC CONTROL for an outstanding job in reviewing papers.

Michael C. Fu received degrees in mathematics and electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA.

He is with the University of Maryland, College Park, as a Professor in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research and an affiliate faculty position in the Department of Electrical and Computer Engineering. He is currently the Simulation Area Editor of *Operations Research*, and has served on the editorial boards of *Management Science*, *INFORMS Journal on Computing*, *IIE Transactions*, and *Production and Operations Management*. He is co-author (with J. Q. Hu) of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications* (Norwell, MA: Kluwer, 1997) which received the INFORMS College on Simulation Outstanding Publication Award in 1998.

Dr. Fu's other awards have included a 1999 IIE Operations Research Division Award, a 1998 *IIE Transactions* Best Paper Award, and the 1995 Maryland Business School's Allen J. Krowe Award for Teaching Excellence.

Steven I. Marcus (F'86) received the B.A. degree in electrical engineering and mathematics from Rice University, Houston, TX, in 1971, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1972 and 1975, respectively.

From 1975 to 1991, he was with the Department of Electrical and Computer Engineering, the University of Texas at Austin, where he was the L. B. (Preach) Meaders Professor in Engineering. He was Associate Chairman of the Department during the period 1984–1989. In 1991, he joined the University of Maryland at College Park, where he was Director of the Institute for Systems Research until 1996. He is currently a Professor in the Electrical and Computer Engineering Department and the Institute for Systems Research, and is Acting Chairman of the Electrical and Computer Engineering Department. He has worked extensively in many aspects of systems and control theory, estimation, stochastic and adaptive control, and discrete event systems. Currently, his research is focused on stochastic control and estimation, with applications in semiconductor manufacturing, telecommunication networks, and preventive maintenance.

Dr. Marcus is Editor-in-Chief of the SIAM *Journal on Control and Optimization*, Associate Editor of *Discrete Event Dynamic Systems: Theory and Applications*, Associate Editor of *Acta Applicandae Mathematicae*, and Associate Editor of *Mathematics of Control, Signals, and Systems*.

Pedram J. Fard received the B.S.E.E. degree from Sharif University of Technology, Tehran, Iran, in 1992, and the M.S.E.E. degree from University of Maryland, College Park, in 1997, where he is currently working toward the Ph.D. degree in electrical engineering.

He is currently with Hughes Network Systems, Germantown, MD. His research interests lie in the area of control theory and its applications in data communication networks.