Rate Based ABR Flow Control using Two Timescale SPSA

Shalabh Bhatnagar, Michael C. Fu and Steven I. Marcus *
Institute for Systems Research, University of Maryland, College Park, MD 20742

ABSTRACT
In this paper, a two timescale simultaneous perturbation stochastic approximation (SPSA) algorithm is developed and applied to closed loop rate based available bit rate (ABR) flow control. The relevant convergence results are stated and explained. Numerical experiments demonstrate fast convergence even in the presence of significant delays and a large number of parameterized policy levels.

Keywords: Rate-based ABR flow control, Two timescale SPSA, Optimal parameterized feedback policies.

1. INTRODUCTION

The available bit rate (ABR) service in asynchronous transfer mode (ATM) networks is used primarily for data traffic. As the name suggests, bandwidth allocation for ABR service is done only 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 flow control their traffic. Two main proposals discussed at the ATM forum for flow control in ABR service were the rate based and the credit based schemes. The rate based scheme was recently accepted by the ATM forum primarily because of the higher hardware complexity and costs involved in the latter scheme. Several heuristic algorithms for computation of explicit ABR rates have since been proposed at the ATM forum. Most other approaches use fluid models of the network. Even though of obvious practical interest, not much is so far available in terms of stochastic control approaches in the queuing framework largely because of the inherent difficulties with such approaches. For instance, the problem has been studied as a team problem in the discrete time queuing framework but with linearized queue dynamics. In another paper, a continuous time queueing model has been studied and stability conditions for various feedback policies have been obtained. However, performance analysis there has been done only under the assumptions of no delays and continuous observations. In this paper, we also study the rate based ABR flow control problem in the continuous time queuing framework. We formulate the problem as a parameter optimization problem where we use an efficient simulation based stochastic approximation scheme for computing optimal ABR rates in the presence of delays with observations and information feedback done at discrete time instants. We essentially view the ABR problem as a stochastic optimization problem wherein we consider parameterized policies that have several levels of control. We employ a two timescale simultaneous perturbation stochastic approximation (SPSA) algorithm for this purpose. This algorithm is a variant of a two timescale stochastic approximation algorithm originally developed for simulation based parametric optimization of hidden Markov models. The two timescale algorithm had the advantage that it updates the parameter vector at increasing deterministic instants unlike several other schemes which do so at regeneration epochs. However, even though it uses two parallel simulations for any $N$-vector parameter, it had the disadvantage that it updates the parameter once every $N$-cycles. Using the SPSA variant of the earlier two timescale algorithm, we still need only two parallel simulations, but we update the parameter now once every cycle (instead of every $N$-cycles as earlier). As a result, in the earlier case, the speed of convergence deteriorates considerably as $N$ becomes large, which is not so using our scheme. The gradient for the SPSA version of the two timescale algorithm is obtained from two performance measurements taken at randomly perturbed settings of the parameter components, most commonly by using i.i.d. symmetric Bernoulli random variables. Our numerical experiments indicate that the two timescale SPSA algorithm is particularly effective in obtaining optimal multilevel feedback policies (with as many levels as one wants) under non-zero delays and sampling instants. This problem was earlier studied with the uncontrolled stream Poisson using the original two timescale algorithm.

The rest of the paper is organized as follows: In Section 2, we formulate the problem, define the multilevel parameterized feedback policies, present the SPSA algorithm for obtaining an optimal such (multilevel) policy and finally state and briefly explain the relevant convergence theorems. In Section 3, numerical examples are presented
for various parameter settings and the superiority of closed loop feedback optimal parameterized policies over optimal open loop policies is illustrated, in addition to obtaining fast convergence using this scheme.

2. THE OPTIMIZATION PROBLEM

A single queue with buffer size $B$ is fed with two input streams. This queue may represent a bottleneck node on the virtual circuit of an ABR source. One of the input streams is controlled (representing the traffic from the ABR source) and the other one is uncontrolled (representing all the other traffic in the network passing through this node). The uncontrolled stream is modelled as a Markov modulated Poisson process (MMPP). The service time process is assumed to be i.i.d. with a general distribution which is same for both the streams. Generalization to the case where this is different for the two streams is straightforward. The size of the bottleneck buffer is $B$ which we assume to be a large quantity. The ABR stream is modelled as a controlled Poisson process with instantaneous intensity $\lambda_c(t)$ at time $t$ and which in turn is defined by a feedback controlled law defined in (1) below. The ABR rate is computed at the node and is fed back to the source with a delay $D_b$. The source then starts sending packets with the new rate. These packets experience a propagation delay $D_f$ in arriving at the node. The rate $\lambda_c(t)$ of the ABR source is held fixed in time intervals $[nT,(n+1)T)$ (as $\lambda_c(nT)$), $n \geq 0$, for some given fixed $T > 0$. By abuse of notation, we shall represent $\lambda_c(nT)$ by just $\lambda_c(n)$. The queue length process $(q(t), t \geq 0)$ is observed at instants $nT$ $(q_n \overset{\Delta}{=} q(nT))$, $n = 0, 1, \ldots$, and from it the desired ABR rate for the next time interval is computed at the node using the following feedback law: Let $S = \{0, 1, \ldots, B\}$ represent the set of all queue-length values. Let $a_i$, $i = 0, 1, \ldots, N$, be integers such that $-1 = a_0 < a_1 < a_2 < \ldots < a_{N-1} < a_N = B$. Define the following subsets of $S$: $S_i = \{a_{i-1} + 1, \ldots, a_i\}$, $i = 1, \ldots, N$. Then for $i = 1, \ldots, N$,

$$\lambda_c(n) = \lambda_i \text{ if } q_n \in S_i,$$

(1)

for some $\lambda_1, \ldots, \lambda_N$. Let $\theta \overset{\Delta}{=} (\lambda_1, \lambda_2, \ldots, \lambda_N)^T$ represent the $N$-dimensional parameter vector to be optimized that takes values in a compact set $C \subset \mathbb{R}^N$. We shall assume in particular that $C$ is of the form $\prod_{i=1}^{N} [\lambda_{i,\text{min}}, \lambda_{i,\text{max}}]$, with $\lambda_{i,\text{min}} > 0$, for all $i = 1, \ldots, N$. Let $X_n$ represent the state of the modulating chain of the uncontrolled MMPP at time $nT$ that takes values in a finite set $S_u$. Further, let $r_n$ represent the residual service time of the customer in service at time $nT$. Then under the type of policies (1), for $D_b = D_f = 0$, it can be seen that $\{(q_n, X_n, r_n)\}$, $n \geq 0$, is ergodic Markov. When $D_b, D_f$ are non-zero, we will assume for simplicity that $D_b + D_f = MT$ for some integer $M > 0$. Simple modifications can however take care of the case when $D_b + D_f \neq MT$ for any $M > 0$. It can also be seen that when $D_b + D_f = MT$ for some $M > 0$, the joint process $\{(q_n, X_n, r_n, q_{n-1}, X_{n-1}, r_{n-1}, \ldots, q_{n-M}, X_{n-M}, r_{n-M})\}$, $n \geq 0$, is ergodic Markov under the above policies. For ease of exposition, we will 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 $D_b = D_f = 0$, for any given $\theta$, $\{(q_n, X_n, r_n)\}$ is ergodic Markov with $\{\lambda_c(n)\}$ given by (1). Let $h : S \rightarrow \mathbb{Z}^+$ be a given bounded and nonnegative cost function ($\mathbb{Z}^+$ is the space of nonnegative integers). Our aim is to minimize

Figure 1. The ABR Model
the long-run average cost

\[ J(\theta) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} h(q_i). \]

It can be shown⁶ that if the underlying process is ergodic Markov under all policies, the limit above exists w.p. 1 and is deterministic for each \( \theta \). Let \( \delta > 0 \) be a fixed small constant. Define sequences \( \{a(n)\} \) and \( \{b(n)\} \) in \( (0,1] \) as follows: \( a(0) = b(0) = 1, a(i) = i^{-1}, b(i) = i^{-\alpha}, i \geq 1, \) for some \( \frac{1}{2} < \alpha < 1. \) Then it is clear that

\[ \frac{a(n+1)}{a(n)}, \frac{b(n+1)}{b(n)} \to 1, \text{ as } n \to \infty, \]

\[ \sum_{n} a(n) = \sum_{n} b(n) = \infty, \sum_{n} a(n)^2, \sum_{n} b(n)^2 < \infty, a(n) = o(b(n)). \] (3)

Define \( \{n_m, m \geq 0\} \) as follows: \( n_0 = 1 \) and \( n_{m+1} = \min\{j > n_m | \sum_{i=n_m}^{j} a(i) \geq b(m)\}, m \geq 1. \) Let \( \pi_i(x) \) denote the point closest to \( x \in \mathcal{R} \) in the interval \( [\lambda_{i,\min}, \lambda_{i,\max}] \subset \mathcal{R} \) (defined earlier) and \( \pi(\theta) \) be defined by \( \pi(\theta) = (\pi_1(\lambda_1), \pi_2(\lambda_2), ..., \pi_N(\lambda_N))^T. \) Let \( \Delta(i) = (\Delta_1, i, ..., \Delta_N, i)^T, i = 1, 2, ..., \) be such that for any \( i, \Delta_1, i, ..., \Delta_N, i \) are symmetric Bernoulli distributed i.i.d. random variables (i.e., \( \Delta_i, m = \pm 1 \) w.p. \( 1/2, \forall i = 1, 2, ..., m = 1, 2, ..., N \) and the random vector \( \Delta(i) \) is independent of \( \Delta(j) \) for any \( i \neq j \). Define parallel processes \( \{(q_j^1, X_j^1, r_j^1)\} \) and \( \{(q_j^2, X_j^2, r_j^2)\} \) such that for \( n_m < j \leq n_{m+1}, \{(q_j^1, X_j^1, r_j^1)\} \) is governed by \( \pi(\theta(m) - \delta \Delta(m)) \hat{\sim} \pi_1(\lambda_1(m) - \delta \Delta_{m,1}), ..., \pi_N(\lambda_N(m) - \delta \Delta_{m,N})^T \). Similarly, \( \{(q_j^2, X_j^2, r_j^2)\} \) is governed by \( \pi(\theta(m) + \delta \Delta(m)) \) for \( n_m < j \leq n_{m+1} \) defined analogously. In the above, \( \theta(m) \hat{\sim} \pi_1(\lambda_1(m), ..., \lambda_N(m))^T \) represents the \( m \)-th update of \( \theta \) and is governed by the following recursion equations: For \( i = 1, ..., N, \)

\[ \lambda_i(m+1) = \pi_i \left( \lambda_i(m) + \sum_{j=n_{m+1}}^{n_{m+1}} a(j) \frac{h(q_j^1) - h(q_j^2)}{2\delta \Delta_{m,i}} \right). \] (4)

Let us now briefly mention the original two timescale algorithm⁴ for comparison purposes. Let \( \{(q_j, X_j, r_j)\} \) and \( \{(q_j^*, X_j^*, r_j^*)\} \), be processes that are respectively governed by \( \{\tilde{\theta}_j\}, \{\tilde{\theta}_j\} \), where \( \tilde{\theta}_j = \theta(m) \) for \( n_{m} < j \leq n_{m+1}, \tilde{\theta}_j = \pi(\theta(m) + \delta e_i), i = 1, ..., N, \) for \( j = n_{m+1}, n_{m+1} + 1, ..., n_{m+1} + 1, i = 1, ..., N, m \geq 0, \) and where \( e_i \) is the unit vector with 1 in the \( i \)-th place and 0 elsewhere. Then the algorithm⁴ is as follows: For \( i = 1, ..., N, \)

\[ \lambda_i(m+1) = \pi_i \left( \lambda_i(m) + \sum_{j=n_{m+1}}^{n_{m+1}} a(j) \frac{h(q_j^*) - h(q_j)}{\delta} \right). \] (5)

The SPSA algorithm (4) has faster rate of convergence than (5). In fact, (4) tracks trajectories of the ordinary differential equation (o.d.e) (6) below.⁵ It can be shown⁴ that (5) tracks trajectories of an o.d.e that is similar to (6) but has a factor of \( 1/N \) multiplying its RHS. Thus even though the qualitative behaviour of the two algorithms is similar, the factor of \( 1/N \) in effect slows down the rate of convergence of (5). Lemma 1 and Theorem 1 below establish the preliminary hypotheses for convergence of the SPSA scheme and Theorem 2 gives the main convergence result. We have skipped detailed proofs of the results presented here. Let \( p_\theta(i; x, r; i', x', d') \), i, i' \( \in \mathcal{S}, x, x' \in \mathcal{X}, r, r' \in [0, \infty), \) represent the transition probabilities of the Markov chain \( \{(q_n, X_n, r_n)\} \) for given \( \theta \). Lemma 1 establishes continuity of the transition kernel and can be shown using sample path arguments.

**Lemma 1** Under all policies of type (1), the map \( (\theta, r) \to p_\theta(i; x, r; i', x', d') \) is continuous in \( \theta \). \( \square \)

The following theorem shows that the average cost \( J(\theta) \) is continuously differentiable in the parameter \( \theta \) and has been shown⁶ for the case when the service times are exponential. Let \( \mu_\theta(q, x, dr) \) represent the stationary distribution of the Markov chain \( \{(q_n, X_n, r_n)\} \) when \( \theta \in \mathcal{C} \) is a given fixed parameter. Let \( \nu_\theta(q) \) represent its marginal along the process \( \{q_n\} \) alone. Then \( J(\theta) \) can be written as

\[ J(\theta) = \sum_{q \in \mathcal{S}} h(q) \nu_\theta(q). \]
Theorem 1 Under all policies of type (1) and under exponential service times, \( J(\theta) \) is continuously differentiable in \( \theta \).

The proof of Theorem 1 is based on Theorem 2\(^{14} \) from which one can show that the stationary distribution of a parametrized Markov process that is ergodic under all policies is continuously differentiable in the parameter if its stationary transition probabilities are. This result is however valid only for finite state chains. Note that the service time distribution is independent of the parameter \( \theta \). Moreover, for exponential service distributions, the residual service times are exponentially distributed as well (because of the memoryless property of exponential distribution), and in fact the subchain \( \{(q_n, X_n)\}, n \geq 0 \), for exponential service distributions and under the type of feedback policies (1), is by itself ergodic Markov for every \( \theta \in C \). Also note that this subchain (for any given \( \theta \)) has a finite state space since both \( S \) and \( S_u \) are finite. Now the stationary distribution \( \mu_\theta(q, x, dr) \) of the original chain \( \{(q_n, X_n, r_n)\} \) for given \( \theta \) can be decomposed as

\[
\mu_\theta(q, x, dr) = \phi_\theta(q, x)\psi(dr),
\]

where \( \phi_\theta(q, x) \) is the stationary distribution of the subchain \( \{(q_n, X_n)\} \) and \( \psi(dr) \) is exponential with rate \( \mu \) (the service rate). Moreover, it can be shown\(^5 \) that the transition probabilities of the subchain \( \{(q_n, X_n)\} \) are continuously differentiable in the parameter. Hence \( \phi_\theta(q, x) \) and hence \( \mu_\theta(q, x, dr) \) are continuously differentiable in \( \theta \). As a result, \( \nu_\theta \) (defined earlier) is continuously differentiable in \( \theta \) as well and hence the average cost \( J(\theta) \) trivially is (since \( S \) is finite). The problem of proving Theorem 1 for i.i.d. general service times remains an open problem.

We now present our main convergence theorem. The ordinary differential equation (o.d.e.) technique is commonly used to prove convergence of stochastic approximation algorithms. Theorem 2 shows that the algorithm (4) asymptotically tracks the stable points of the o.d.e. (6) below. Let \( \tilde{Z}(t) \equiv (\tilde{Z}_1(t), \ldots, \tilde{Z}_N(t)) \), where \( \tilde{Z}_i(t), i = 1, \ldots, N \), satisfy the o.d.e.

\[
\dot{\tilde{Z}}_i(t) = \tilde{\pi}_i(-\nabla_i J(\tilde{Z}(t))), \quad i = 1, \ldots, N, \quad \tilde{Z}(0) \in C,
\]

where for any \( v(.) \in C_b(\mathcal{R}) \),

\[
\tilde{\pi}_i(v(y)) = \lim_{\delta \to 0} \frac{\pi_i(y + \Delta v(y)) - \pi_i(y)}{\Delta}.
\]

For \( x = (x_1, \ldots, x_N) \in \mathcal{R}^N \), let \( \tilde{\pi}(x) = (\tilde{\pi}_1(x_1), \ldots, \tilde{\pi}_N(x_N))^T \). Let \( K = \{ \theta \in C \mid \tilde{\pi}(\nabla J(\theta)) = 0 \} \). Note that, since \( \tilde{\pi}(\nabla J(\theta)).\tilde{\pi}(\nabla J(\theta)) < 0 \), for \( \theta \not\in K \), by definition, \( J(.) \) itself is a strict Lyapunov function for (6), and the o.d.e. (6) for arbitrary initial conditions converges to the set \( K \) (which serves as the asymptotically stable attractor for the o.d.e\(^6 \)). For \( \eta > 0 \), let \( K^\eta = \{ \theta \in C \mid 3\delta \in K \text{ s.t. } ||\theta - \theta'|| \leq \eta \} \).

Theorem 2 Given \( \eta > 0 \), \( \exists \delta > 0 \) such that for any \( \delta \in (0, \delta) \), the algorithm (4) converges to \( K^\eta \) a.s.

Remark Note that \( K \) is the set of all critical points of (6), and not just the set of local minima. However, points in \( K \) that are not local minima will be unstable equilibria, and because of the presence of noise (randomness) in the algorithm, under fairly general conditions, the algorithm will converge to the \( \eta \)-neighborhood of \( K_0 \equiv \{ \text{the set of local minima of } J(.) \} \subset K \).\(^{12} \) Note that we assumed the cost function to be merely bounded and continuous. If on the other hand, we assume the cost function \( h(.) \) to be in addition convex, the average cost \( J(.) \) will be convex as well. Moreover, if \( J(.) \) is strictly convex, it will have a unique minimum, to which our algorithm will a.s. converge within an \( \eta \)-neighborhood.

3. NUMERICAL EXPERIMENTS

Next, we provide numerical results to illustrate the two-timescale SPSA scheme. As mentioned earlier, 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 \( N_0 \).\(^{18,10,15,3} \) We adopt a similar approach here, i.e., we select our cost function to be \( h(x) = |x - N_0| \), where \( N_0 \) is assumed given. In other words we assume that the network knows the value of \( N_0 \) at the time of setting up the ABR connection. A stochastic approximation scheme for obtaining an optimal such \( N_0 \) can also be provided.\(^5 \)

We performed experiments with policies that have five and eleven parameter levels. However, we describe here experiments with five level policies only since results for those with eleven level policies are similar.\(^5 \) We assume for
simplicity 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:

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

(7)

In the above, we assume that $\epsilon$ is a given fixed constant in addition to $N_0$. The eleven level policies are similarly defined. In the numerical experiments, we actually consider a generalization of the model in Fig.1, where rate feedback is done at instants $nF_b, n \geq 1$, where $F_b$ is a fixed multiple of $T$. This gives us added flexibility in studying the effect of changes in $F_b$ in addition to those in $T$. Thus $F_b$ in some sense plays the role 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 parameter vector is computed off-line using the two timescale SPSA algorithm (4). The uncontrolled process is an MMPP with the underlying Markov chain being an irreducible two state chain. For simplicity we assume that the underlying chain undergoes state transitions every $T$ units of time. We make this assumption only to simplify the simulation code. The buffer size $B$ in these experiments is $5 \times 10^6$. In what follows, we shall illustrate simulation experiments using our stochastic approximation scheme. We also compare closed loop performance (with delays etc.) with the optimal open loop performance that is obtained using the scalar parameter ($N = 1$) version of (5).

Let $\theta^*$ denote the parameter value for the corresponding optimal policy, i.e., $\theta^* = (\lambda_1^*, ..., \lambda_5^*)^T$ for the case of 5-level closed loop policies that we show here and $\theta^* = \lambda^*$ for the open loop policy. In the following, subscript $\theta^*$ is used in the definition of various performance measures to indicate $\theta^*$-parametrized stationary distributions of the various quantities. Thus, $\text{Var}_{\theta^*}(q_n)$ represents the stationary variance of $\{q_n\}$ parameterized by $\theta^*$. 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; $\bar{q}, P_{\text{band}}, \sigma_q, \lambda_c, \lambda_{\text{c}}$, $P_{\text{idle}}$ and $J(\theta^*)$. These quantities and their estimates are defined as follows:

$$
\bar{q} \triangleq E_{\theta^*}[q_n] \approx \frac{1}{N} \sum_{i=0}^{N-1} q_i, \quad \sigma_q \triangleq \text{Var}_{\theta^*}(q_n) \approx \left( \frac{1}{N} \sum_{i=0}^{N-1} q_i^2 \right) - (\bar{q})^2,
$$

$$
P_{\text{band}} \triangleq P_{\theta^*}(q_n \in B_d) \approx \frac{1}{N} \sum_{i=0}^{N-1} I\{q_i \in B_d\}, \quad P_{\text{idle}} \triangleq P_{\theta^*}(q_n = 0) \approx \frac{1}{N} \sum_{i=0}^{N-1} I\{q_i = 0\},
$$

$$
\lambda_c \triangleq E_{\theta^*}[\lambda_c(n)] \approx \frac{1}{N} \sum_{i=0}^{N-1} \lambda_c(i), \quad J(\theta^*) \approx \frac{1}{N} \sum_{i=0}^{N-1} |q_i - N_0|,
$$

where $N$ is taken as $10^6$ in our experiments. The last performance measure is the one that the algorithm seeks to minimize, but clearly the others are closely related and are included here because they are often taken as measures of performance in ABR service. One desires $P_{\text{band}}$ to be high in order to satisfy the various other performance criteria. One expects as a consequence of the above minimization that this quantity will be maximized. The measure $P_{\text{idle}}$ gives the stationary probability of the server lying idle and should be close to zero. The average ABR throughput rate $\lambda_c$ 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. We now illustrate our numerical experiments.

In the simulations for the five-level policies, $\lambda_1, ..., \lambda_4 \in [0.10, 3.0]$, with $\lambda_5 \in [0.10, 0.90]$. We consider the following setting for the uncontrolled traffic: $\lambda_{u,1} = 0.05, \lambda_{u,2} = 0.15, p(1; 1) = p(1; 2) = p(2; 1) = p(2; 2) = 0.5$. Thus, the mean uncontrolled traffic rate ($\lambda_u$) is 0.1. The service rate $\mu = 1.0$. Further, the constants $N_0$ and $\epsilon$ are chosen to be 10 and 1, respectively. Thus the band $B_d$ is [9, 11]. For the SPSA algorithm (4), in the step-size sequence $\{b(n)\}$ in (2)-(3), $\alpha$ is chosen to be 2/3. Thus $b(0) = 1$ and $b(i) = i^{2/3}$, $i \geq 1$. Moreover, the perturbation parameter $\delta$ is chosen to be 0.12. Throughout, ‘O.L.’ represents the optimal open loop policy. We present various experiments for the cases $D_b = D_f = 0$ with increasing $T$ and $F_b$ (Table 1) and increasing $D_b, D_f$ with fixed $T$ and $F_b$ (Table 2). We also show experiments (Table 3) with two controlled sources feeding into the same bottleneck node but with rate
information fed back with different delays (almost without any delay to the first and with a significant delay to the second). The bandwidth in this case is found to be shared equally by the two sources. This amounts to our scheme showing fairness in performance. However, we are aware of the fact that the appropriate framework to study fairness is in tandem queues with different ABR sources feeding packets through different sets of nodes. For small $D_b, D_f, T$ and $F_b$, our algorithm converges in about 130-150 iterations while for large values of the above parameters, the algorithm takes about 200-250 iterations for convergence. On a Sun Ultra10 workstation with UNIX operating system, the algorithm takes about 5-10 minutes for small $D_b, D_f, T$ and $F_b$, while for large values of these, it takes about 30-50 minutes for convergence. We also tried running the original two timescale stochastic approximation algorithm (under the same settings of $\alpha$ and $\delta$ as the SPSA algorithm) with $D_b = D_f = 0$. It did not converge even after 350 iterations (but was close to it) after running for nearly 200 minutes. This confirms that the SPSA version of the two timescale stochastic approximation scheme shows faster convergence than the original two timescale scheme. Some of our important observations are as follows:

1. We get the best performance for lowest $T$, $F_b$, $D_b$ and $D_f$ (see Tables 1 and 2) below. The difference between the lowest and the highest rates ($\lambda_b^*$ and $\lambda_f^*$ respectively) decreases as any of $T$, $F_b$, $D_b$ or $D_f$ increase.

2. We utilize almost the entire bandwidth, viz., $\lambda_c \approx 0$, and $\lambda_c + \lambda_u \approx \mu$, even when $T$ and $F_b$ or the delays $D_b$ and $D_f$ are high (Tables 1 and 2).

3. The performance degrades as $T$ and $F_b$ or $D_b$ and $D_f$ increase, but remains better than the optimal open loop case even when these parameters become significantly high. For $T = 1$ and $F_b = 2$ (Table 2), performance is better than the optimal open loop case even for $D_b + D_f = 150$.  

4. We considered the case of two controllers feeding arrivals into the same bottleneck node in addition to the uncontrolled MMPP stream (Table 3). Explicit rate information was fed back to the two sources with different delays $D_{b1}$ and $D_{b2}$. Further there were different delays $D_{f1}$ and $D_{f2}$ in customers arriving to the bottleneck node from the two sources. We observed that the stationary mean rates $\lambda_{c1}$ and $\lambda_{c2}$, for the two sources are almost the same even when the difference in the delays is significantly high. This amounts to our scheme showing 'fairness' in performance. We also observed the other performance metrics in this case and found that the performance here is not as good as that of a single source with the lower delays (in the setting of Fig.1) and also it is not as bad as that of a single source with the higher delays. Thus in some sense the performance of the two sources here is getting averaged and which is possibly the reason for getting fairness in performance.

We also performed other several experiments with varying $\epsilon$, $N_0$ and uncontrolled MMPP settings (in addition to those with eleven level policies). These have not been reported here.

4. CONCLUSIONS

We studied the problem of rate based ABR flow control in the presence of information and propagation delays, in the continuous time queueing framework by developing and applying a numerically efficient two timescale SPSA algorithm. Relevant convergence theorems were stated and explained. Numerical experiments were conducted to verify and illustrate the theoretical results. In the numerical experiments, we addressed the problem (an approach common in the literature) by minimizing the distance of stationary mean queue length from a given constant $N_0$. Our experiments indicate that closed loop policies lead to a significant improvement in performance for reasonable values of information and propagation delays. The algorithm readily computes optimal policies amongst multilevel parametrized policies, even when there are significant delays, by using just two parallel simulations for any number of parameter levels. It was found that the scheme converges orders of magnitude faster than a previously proposed two timescale stochastic approximation scheme.

We also considered experiments with two ABR sources sharing the same bottleneck node but with the two sources experiencing significantly different propagation and information delays. We found that the sources under stationarity share the bandwidth equally between them. This is a very interesting observation and it amounts to our scheme exhibiting fairness in performance. The same should however be tried on tandem queues to conclusively demonstrate the above claim.

ACKNOWLEDGEMENTS

This research was supported by the NSF under Grant DMI-9713720, by the Semiconductor Research Corporation under Grant 97-FJ-491 and by DoD contract Number MDA90497C3015.
REFERENCES


Author for Correspondence: Email: marcus@isr.umd.edu; Fax: (301) 314-9920
Table 1: $D_b = D_f = 0$

<table>
<thead>
<tr>
<th>$T$</th>
<th>$F_b$</th>
<th>$\lambda_0^*$</th>
<th>$\lambda_1^*$</th>
<th>$\lambda_2^*$</th>
<th>$\lambda_3^*$</th>
<th>$\lambda_4^*$</th>
<th>$\tilde{q}$</th>
<th>$P_{\text{band}}$</th>
<th>$\sigma_q$</th>
<th>$\lambda_c$</th>
<th>$P_{\text{idle}}$</th>
<th>$J(\theta^*)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.7</td>
<td>1.7</td>
<td>2.6</td>
<td>10.1</td>
<td>0.45</td>
<td>7.3</td>
<td>0.90</td>
<td>0</td>
<td>2.01</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>0.2</td>
<td>0.5</td>
<td>0.8</td>
<td>1.3</td>
<td>1.8</td>
<td>9.5</td>
<td>0.34</td>
<td>12.8</td>
<td>0.90</td>
<td>0</td>
<td>2.71</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0.3</td>
<td>0.6</td>
<td>0.9</td>
<td>1.2</td>
<td>1.5</td>
<td>10.0</td>
<td>0.26</td>
<td>21.3</td>
<td>0.89</td>
<td>0.01</td>
<td>3.52</td>
</tr>
<tr>
<td>1</td>
<td>O.L.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5.3</td>
<td>0.08</td>
<td>32.3</td>
<td>0.76</td>
<td>0.17</td>
<td>6.39</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.3</td>
<td>0.7</td>
<td>0.9</td>
<td>1.1</td>
<td>1.4</td>
<td>9.8</td>
<td>0.26</td>
<td>21.6</td>
<td>0.89</td>
<td>0.02</td>
<td>3.59</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>0.4</td>
<td>0.8</td>
<td>0.9</td>
<td>1.1</td>
<td>0.9</td>
<td>8.2</td>
<td>0.17</td>
<td>34.1</td>
<td>0.85</td>
<td>0.07</td>
<td>4.77</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>0.3</td>
<td>0.4</td>
<td>0.6</td>
<td>0.9</td>
<td>0.9</td>
<td>6.4</td>
<td>0.12</td>
<td>38.3</td>
<td>0.75</td>
<td>0.16</td>
<td>5.97</td>
</tr>
<tr>
<td>5</td>
<td>O.L.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5.1</td>
<td>0.09</td>
<td>31.9</td>
<td>0.76</td>
<td>0.16</td>
<td>6.39</td>
</tr>
</tbody>
</table>

Table 2: $T = 1$, $F_b = 2$

<table>
<thead>
<tr>
<th>$D_b$</th>
<th>$D_f$</th>
<th>$\lambda_0^*$</th>
<th>$\lambda_1^*$</th>
<th>$\lambda_2^*$</th>
<th>$\lambda_3^*$</th>
<th>$\lambda_4^*$</th>
<th>$\tilde{q}$</th>
<th>$P_{\text{band}}$</th>
<th>$\sigma_q$</th>
<th>$\lambda_c$</th>
<th>$P_{\text{idle}}$</th>
<th>$J(\theta^*)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.3</td>
<td>0.7</td>
<td>1.8</td>
<td>2.2</td>
<td>10.1</td>
<td>0.38</td>
<td>9.7</td>
<td>0.89</td>
<td>0</td>
<td>2.41</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.8</td>
<td>1.3</td>
<td>2.2</td>
<td>9.8</td>
<td>0.38</td>
<td>9.7</td>
<td>0.88</td>
<td>0</td>
<td>2.42</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.4</td>
<td>0.7</td>
<td>0.9</td>
<td>1.0</td>
<td>1.2</td>
<td>8.8</td>
<td>0.20</td>
<td>28.0</td>
<td>0.87</td>
<td>0.03</td>
<td>4.46</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.6</td>
<td>1.0</td>
<td>8.4</td>
<td>0.14</td>
<td>39.5</td>
<td>0.84</td>
<td>0.06</td>
<td>5.23</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>0.5</td>
<td>0.8</td>
<td>0.9</td>
<td>1.1</td>
<td>1.0</td>
<td>8.2</td>
<td>0.13</td>
<td>44.0</td>
<td>0.80</td>
<td>0.06</td>
<td>5.43</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.9</td>
<td>8.1</td>
<td>0.13</td>
<td>48.0</td>
<td>0.79</td>
<td>0.08</td>
<td>5.74</td>
</tr>
<tr>
<td>40</td>
<td>10</td>
<td>0.5</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.9</td>
<td>6.4</td>
<td>0.12</td>
<td>52.7</td>
<td>0.78</td>
<td>0.12</td>
<td>5.83</td>
</tr>
<tr>
<td>40</td>
<td>30</td>
<td>0.5</td>
<td>0.5</td>
<td>0.6</td>
<td>0.9</td>
<td>0.9</td>
<td>6.9</td>
<td>0.12</td>
<td>56.4</td>
<td>0.78</td>
<td>0.12</td>
<td>5.88</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>0.5</td>
<td>0.5</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>7.1</td>
<td>0.12</td>
<td>65.2</td>
<td>0.78</td>
<td>0.10</td>
<td>5.97</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>0.5</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.9</td>
<td>6.7</td>
<td>0.11</td>
<td>59.5</td>
<td>0.77</td>
<td>0.13</td>
<td>6.07</td>
</tr>
<tr>
<td>O.L.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5.3</td>
<td>0.08</td>
<td>32.3</td>
<td>0.76</td>
<td>0.17</td>
<td>6.39</td>
</tr>
</tbody>
</table>

Table 3 (Two ABR Sources): $T = 1$, $F_b = 2$

<table>
<thead>
<tr>
<th>$D_b$</th>
<th>$D_{b1}$</th>
<th>$D_{b2}$</th>
<th>$D_{f1}$</th>
<th>$D_{f2}$</th>
<th>$\lambda_0^*$</th>
<th>$\lambda_1^*$</th>
<th>$\lambda_2^*$</th>
<th>$\lambda_3^*$</th>
<th>$\lambda_4^*$</th>
<th>$\tilde{q}$</th>
<th>$P_{\text{band}}$</th>
<th>$\sigma_q$</th>
<th>$\lambda_{c1}$</th>
<th>$\lambda_{c2}$</th>
<th>$P_{\text{idle}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
<td>0.8</td>
<td>8.6</td>
<td>0.26</td>
<td>18.1</td>
<td>0.44</td>
<td>0.45</td>
<td>0.02</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>4</td>
<td>10</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.7</td>
<td>0.7</td>
<td>8.9</td>
<td>0.23</td>
<td>29.4</td>
<td>0.44</td>
<td>0.44</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>4</td>
<td>20</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>8.1</td>
<td>0.21</td>
<td>29.1</td>
<td>0.42</td>
<td>0.43</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>4</td>
<td>100</td>
<td>0.1</td>
<td>0.1</td>
<td>0.6</td>
<td>0.5</td>
<td>0.6</td>
<td>8.7</td>
<td>0.18</td>
<td>35.9</td>
<td>0.41</td>
<td>0.42</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>