

# OPTIMUM FEEDBACK STRATEGY FOR ACCESS CONTROL MECHANISM MODELLED AS STOCHASTIC DIFFERENTIAL EQUATION IN COMPUTER NETWORK

N. U. AHMED AND CHENG LI

*Received 23 March 2004*

We consider optimum feedback control strategy for computer communication network, in particular, the access control mechanism. The dynamic model representing the source and the access control system is described by a system of stochastic differential equations developed in our previous works. Simulated annealing (SA) was used to optimize the parameters of the control law based on neural network. This technique was found to be computationally intensive. In this paper, we have proposed to use a more powerful algorithm known as recursive random search (RRS). By using this technique, we have been able to reduce the computation time by a factor of five without compromising the optimality. This is very important for optimization of high-dimensional systems serving a large number of aggregate users. The results show that the proposed control law can improve the network performance by improving throughput, reducing multiplexor and TB losses, and relaxing, not avoiding, congestion.

## 1. Introduction

The widely used IP protocol offers best-effort service; it does not guarantee (specified) quality of service (QoS). As IP networks have been extensively deployed, all QoS efforts have to layer on top of the existing technology. The network architectures such as integrated service (IntServ) and differentiated service (DiffServ), which are proposed by Internet Engineering Task Force (IETF), provide QoS to deliver real-time applications effectively and efficiently. Token bucket (TB) algorithm, which is used as a filter to characterize traffic, has been employed in traffic control.

TB size and token generating rate are two major parameters of TB. The size of traffic burst cannot exceed the size of the TB, and average conformed traffic rate is limited up to the token generating rate. TB is usually used for traffic shaping, policing, and marking. In previous papers [2, 1], we developed a TB system model and proposed feedback control laws to police traffic access. The associated control problem treated on the basis of the Hamilton-Jacobi-Bellman (HJB) equation presents formidable computational burden and in fact cannot be solved currently [1]. Instead, a simplified feedback control

law was proposed based on neural network. As indicator functions were used in the system model (see [Section 3](#)), many irregular small random fluctuations were imposed on the overall structure of the system including the objective function. This is a nonsmooth optimization problem and so more difficult.

Network conditions vary with time and the search algorithm should quickly find better network parameters before significant changes occur in the network. Furthermore, network parameter optimization is based on network simulation which might be very time consuming.

Simulated annealing (SA) algorithm [5] was applied to optimize network parameters (weights of the neural network) to achieve a near-optimal FB control law. Although this method can improve the system performance, computational time is large. In order to save the computation time, it is necessary to apply a faster algorithm such as recursive random search (RRS) [10] which was found to reduce the computational time substantially by a factor of five.

Recent network traffic measurement studies have shown the existence of self similarity and long-range dependency (LRD) properties [3, 4, 6, 9]. Self-similarity means that the same statistical patterns of behaviour occur over time scales which can vary by many orders of magnitude. LRD exhibits slowly decaying autocorrelation. In order to analyze the system and the performance of the control law, we also propose a dynamic model for the traffic. This model is governed by a stochastic differential equation (SDE) driven by fractional Brownian motion (FBM). The process so generated possesses self-similarity and LRD properties. It is also important to mention that our model produces nonnegative random process (representing traffic arrival rate) as required [1].

The rest of the paper is organized as follows. Traffic model is presented in [Section 2](#) and the access control mechanism in [Section 3](#). Full system model is presented in [Section 4](#). Objective function and performance measures are given in [Section 5](#). Control strategies are discussed in [Section 6](#). Basic data used for simulation experiments and numerical results are presented in [Sections 7 and 8](#). The paper ends with the conclusion in [Section 9](#).

## 2. Traffic model

We proposed a traffic model based on FBM [1, 8]. In general, the traffic model is composed of a drift component and a fluctuation component. In this section, we introduce the traffic models. Indicator functions are used throughout the paper. They are defined as follows. Let  $S$  denote any logical or mathematical statement. Then the indicator function of  $S$  is defined as follows:

$$I(S) = \begin{cases} 1, & \text{if the statement } S \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

Let  $\mathcal{B}_H(t)$  denote the FBM, associated with the Hurst parameter (self-similarity parameter)  $H$  which is used to describe the degree of LRD and the burstiness of the traffic. For any Hurst parameter  $H \in (0, 1)$  and a (variance) scaling parameter  $\sigma^2$ , the FBM satisfies the following properties:

- (1)  $B_H(t)$  is Gaussian and  $B_H(0) = 0$  with probability one;

- (2)  $E\{B_H(t)\} = 0;$
- (3)  $\text{Cov}\{B_H(t), B_H(s)\} = (\sigma^2/2)(|t|^{2H} - |t-s|^{2H} + |s|^{2H}).$

In particular, when  $H = 0.5$ ,  $B_H(t)$  is the classical Brownian motion. In this case,

$$\text{Cov}\{B_H(t), B_H(s)\} = \sigma^2(t \wedge s), \quad \text{Var}(B_H(t) - B_H(s)) = \sigma^2|t - s|. \tag{2.2}$$

For  $H \neq 0.5$ , the increments  $B_H(t) - B_H(s)$  are dependent (stochastically correlated) unlike the Brownian motion. This process has LRD when  $0.5 < H < 1$ . For  $\sigma = 1$ , we have the standard Brownian motion. In this paper, we use the FBM  $B_H(t)$ ,  $t \geq 0$ , a Gaussian process, given by the integral transform of the classical Brownian motion,

$$B_H(t) = \int_0^t K_H(t-s)dB(s), \quad t \geq 0, \tag{2.3}$$

where  $\{B(t), t \geq 0\}$  is the standard Brownian motion and  $K(\cdot)$  is a suitable kernel satisfying certain properties.

For our numerical experiments, we choose the simple kernel  $K_H$  given by

$$K_H(t) = C_H t^{(H-1/2)}, \quad \frac{1}{2} < H < 1, \tag{2.4}$$

where  $C_H$  is any constant. Note that for the given choice of the kernel  $K$  and for any  $\alpha \geq 0$ ,  $B^H(\alpha t)(d/-)\alpha^H B^H(t)$ , the equality understood in the sense of their distributions. Thus this process is self-similar. If  $C_H = 1$  for  $H = 1/2$ ,  $\lim_{H \rightarrow 1/2} B_H(t) = B(t)$  for all  $t \geq 0$ .

In a recent paper [1], the authors proposed a traffic model governed by an SDE driven by FBM. This is given by the following equation:

$$da_i = h_i(t)\{I(h_i(t) > 0) + I(a_i(t) > 0, h_i(t) \leq 0)\}dt + \sigma_i I\{a_i(t) > 0\}dB_i^H(t). \tag{2.5}$$

For  $H = (1/2)$ , this equation reduces to the classical SDE of Ito type. Equation (2.5) gives the source dynamics representing the traffic rate generated by any of the  $i$ th users. This equation can be explained as follows. Change of traffic rate is given by the sum of two terms on the right-hand side of the equation. The first term determines the general trend of the traffic. It is active under two different situations: first, when  $h_i(t) > 0$  and the value of  $a_i(t)$  is arbitrary; second, when  $h_i(t) \leq 0$  but  $a_i(t) > 0$ . The last term in the equation contains noise and it is active only when  $a_i(t) > 0$ . This term affects the traffic fluctuation. In case of multiple users, say  $m$ , the scalar  $a_i$  is replaced by an  $m$  vector  $a \equiv \{a_i, i = 1, 2, \dots, m\}$ .

Thus the source dynamics is governed by the following system of SDE of Ito type:

$$da = b(t,a)dt + \sigma(t,a)dB, \tag{2.6}$$

where  $B$  represents either the standard Brownian motion or the FBM taking values in  $\mathbb{R}^m$ . Since, in general, the sources are independent users, the drift vector  $b$  is the vector of infinitesimal rates of individual users and the diffusion (fluctuation/volatility)  $\sigma$  is a diagonal matrix representing the intensity of fluctuation of each of the user's individual traffic. It should be noted that (2.6), based on (2.5), generates self-similar random process

taking values from the positive orthant  $\mathbb{R}_+^m \subset \mathbb{R}^m$  and therefore can be used to model network traffic. For the first time in the literature [1], this model was proposed to describe the source dynamics. It is flexible in the sense that choice of the parameters  $\{h_i\}$  and  $\{\sigma_i\}$  can produce a large class of traffic for each user and distinct traffic for different users. Note that the coefficients characterizing the drift and diffusion are discontinuous functions unlike in standard Ito equations.

### 3. Access control mechanism

Each of the user's traffic is processed by a dedicated TB before it is passed on to the multiplexor, where they are queued up for service into the outgoing link. Considering  $m$  users, the state of the system is described by the occupancy status of all the TBs and the multiplexor giving  $m + 1$  equations.

Let  $T_i > 0$ ,  $i = 1, 2, \dots, m$ , denote the size of the  $i$ th TB, and  $\rho_i(t)$  the number of tokens available at time  $t$  in the  $i$ th TB. We can use the following equation to describe the rate of change of token population, as presented in [1]:

$$d\rho_i = \{u_i(t)I(\rho_i(t) < T_i) - a_i(t)I(\rho_i(t) > 0)\}dt. \quad (3.1)$$

This is simply a balance equation, where the rate of change of token population (count) is given by the algebraic sum of fresh token supply rate  $u_i$  (provided the bucket is not full) and the consumption rate  $a_i$  due to the arrival of traffic from the source  $i$  (subject to the condition that the corresponding bucket is not empty). Clearly, TBs will drop nonconforming packets whenever they are empty.

Let  $q(t)$  denote the state of the multiplexor measured in terms of (any suitable unit, such as bytes) space occupied in its buffer which has size  $Q > 0$ . Let  $C(t)$  denote the (outgoing) link capacity available at time  $t$ . The multiplexor injects the traffic into the link at a rate determined by the available capacity. The dynamics of the multiplexor can then be formulated as follows:

$$dq = \left\{ -C(t)I(q(t) > 0) + \left[ \sum_{i=1}^m a_i(t)I(\rho_i(t) > 0) \right] I(q(t) < Q) \right\} dt. \quad (3.2)$$

This equation describes the occupancy status of the multiplexor. The rate of change of the multiplexor queue is determined by the algebraic sum of link service rate (provided the multiplexor is not empty) and the sum of all conforming traffic rates from all the TBs (subject to the condition that the corresponding multiplexor is not full).

In order to achieve QoS satisfaction of each accessing traffic and maximum utilization of the multiplexor, the controller adjusts the token supply at each TB to control access rate of each traffic. This can be used in current networks to improve utilization by dynamically allocating bandwidth to each user [2, 1].

*Remark 3.1.* Note that in developing these dynamic equations, we have assumed that both the TBs and the multiplexor are noiseless.

**4. Full system model**

Considering (2.6), (3.1), and (3.2) and defining

$$\begin{aligned}
 f_i(t, \rho_i, a_i, u_i) &\equiv \{u_i I(\rho_i < T_i) - a_i I(\rho_i > 0)\}, \\
 f(t, \rho, a, u) &\equiv \text{col}\{f_i(t, \rho_i, a_i, u_i), i = 1, 2, \dots, m\}, \\
 g(t, \rho, a, q) &\equiv \{-C(t)I(q > 0) + (\sum_{i=1}^n a_i I(\rho_i > 0))I(q < Q)\},
 \end{aligned} \tag{4.1}$$

we can rewrite (2.6), (3.1), and (3.2) as a system given by

$$\begin{aligned}
 da &= b(t, a)dt + \sigma(t, a)dB, \\
 d\rho &= f(t, \rho, a, u)dt, \\
 dq &= g(t, \rho, a, q)dt.
 \end{aligned} \tag{4.2}$$

This is a  $(2m + 1 \equiv n)$ -dimensional SDE describing the dynamics of our system that consists of the source (traffic) dynamics and the dynamics of the access control mechanism at the edge of a backbone network. For analysis of the system, it is more convenient to write these equations as an  $n$ -dimensional stochastic differential system. Define the state vector as  $\xi \equiv \text{col}\{a, \rho, q\}$ , the control vector as  $u$ , and the drift and diffusion matrices as

$$F \equiv \begin{pmatrix} b \\ f \\ g \end{pmatrix}, \quad G \equiv \begin{pmatrix} \begin{matrix} \sigma & \dots & \dots & 0 \\ \vdots & \ddots & & \vdots \\ (m \times m) & 0 & & 0 \end{matrix} \\ \vdots \\ 0 \quad \dots \quad \dots \quad 0 \quad \dots \quad \dots \quad 0 \\ \vdots \\ 0 \quad \dots \quad \dots \quad 0 \quad \dots \quad \dots \quad 0 \end{pmatrix}, \tag{4.3}$$

where  $F$  is an  $n$ -vector and  $G$  is an  $n \times m$  matrix-valued function as indicated. Using these notations, we can rewrite our system in the compact form as follows:

$$d\xi(t) = F(t, \xi(t), u(t))dt + G(t, \xi(t))dB, \quad t \geq 0. \tag{4.4}$$

**5. Objective function and performance measures**

**5.1. Objective function.** The objective functional for a network provider proposed in [1] is given by

$$\begin{aligned}
 J(u) &\equiv E \left\{ \int_l \lambda_1(t) \left\{ \sum_{i=0}^n a_i(t) I(\rho_i(t) = 0) \right\} dt \right. \\
 &\quad \left. + \int_l \lambda_2(t) \left\{ \left( \sum_{i=0}^n a_i(t) I(\rho_i(t) > 0) \right) I(q(t) = Q) \right\} dt + \int_l \lambda_3(t) q(t) dt \right\}.
 \end{aligned} \tag{5.1}$$

The first term represents (traffic or cell) losses at TB, the second term gives the losses at the multiplexor, and the third term is a measure of service delay. The functions  $\lambda_i$ ,  $i = 1, 2, 3$ , are nonnegative measurable functions used as weights or importance given to each of the losses. This can be chosen by network designers to reflect as different concerns and scenarios as necessary. Note that the cost function is an implicit functional of the control policy  $\{u\}$ . Clearly, in terms of our final-state equation given by (4.4), the cost functional (5.1) can be compactly written as

$$J(u) = E \left\{ \int_0^T \ell(t, \xi(t)) dt \right\}, \quad (5.2)$$

where  $\ell$  denotes the integrand of the expression (5.1). One of the objectives of the network provider is to improve the system performance by using control strategies that minimize this cost functional. Note that control policy has no direct influence on the source though this information must be used by the controller to achieve its goal. However, the individual sources may adjust their rates by adjusting  $\{h_i\}$  on the basis of information regarding nonconforming traffic at the TBs. This provision requires information feedback which is not included in the model.

**5.2. Other performance measures.** In addition to the concern for cell losses in the system, there are other measures of system performance which are equally important. For example, first time of occurrence of congestion and residence time in state of congestion are important factors characterizing the network performance. Here we define several related concepts.

(A) *System congestion.* Congestion is a state of the network resources in which the aggregate demand may exceed the capacity of the system or the multiplexor is close to its full capacity. Clearly, it is expected that congestion may result in degradation of QoS to the end users [1, 2]. For the system under consideration, we may define the congestion as follows.

*Definition 5.1 (congestion).* The system is said to be in state of congestion whenever the multiplexor state  $q$  enters the closed interval  $Q_\alpha \equiv [\alpha Q, Q]$ , where  $0 < \alpha \leq 1$  and  $Q$  is the multiplexor buffer size.

In general, one may consider the system to be in a state of congestion if 90% of the buffer space is occupied. In this case, one may choose  $\alpha = 0.90$ . The value of  $\alpha$  can be chosen arbitrarily by the service provider. Choice of  $\alpha = 1$  may induce traffic oscillation leading to degradation of QoS.

(B) *Mean first time to congestion.* Traffic loads in a network keep changing randomly, with peaks and valleys occurring during certain periods of operation. Considering the traffic observation period to be  $[0, T]$ , let  $C_T$  denote the set of time instants during which the multiplexor hits the congestion zone. In symbols, this is given by

$$C_T \equiv \{t \in [0, T] : q(t) \in Q_\alpha\}, \quad (5.3)$$

where  $q(t)$  is the state of the multiplexor. The first time to congestion, denoted by  $\tau_c$ , is then given by

$$\tau_c \equiv \begin{cases} \inf \{C_T\}, & \text{if the set is nonempty,} \\ T, & \text{if the set is empty.} \end{cases} \tag{5.4}$$

We are interested in the expected value of this, denoted by

$$T_c \equiv E\{\tau_c\}. \tag{5.5}$$

(C) *Mean residence time in congestion zone.* Residence time in congestion zone is the total time spent by the system in the state of congestion.

*Definition 5.2 (residence time).* For any sample path  $q(t)$ ,  $t \in I \equiv [0, T]$ , representing the history of multiplexor queue, the residence time denoted by  $\tau_r$  is the time spent in the set  $Q_\alpha$  as given by

$$\tau_r \equiv \int_0^T I(q(t) \in Q_\alpha) dt. \tag{5.6}$$

Again, we are interested in the expected value giving

$$T_r \equiv E\{\tau_r\}. \tag{5.7}$$

Note that the cost functional given by (5.6) can be absorbed in the general functional given by (5.2) by adding the integrand of (5.6) to those of (5.1) with desired weight. Later in the sequel, we will use Monte Carlo techniques to compute functionals (5.5) and (5.7) and the measures of performance.

### 6. Control strategies

To find the optimal control for the dynamic system (5.1) with the cost functional (5.3), we may make use of dynamic programming. Let  $\mathcal{F}_t$  be an increasing family of (right continuous) the smallest sigma algebras with respect to which the solution process  $\{\xi(t), t \geq 0\}$  is measurable, and let  $U$  be a closed bounded possibly convex subset of the set  $\mathbb{R}_+^m \equiv \{v \in \mathbb{R}^m : v_i \geq 0, i = 1, 2, \dots, m\}$ , where the controls can take values from it. Let  $\mathcal{U}_{ad}$  denote the class of  $\mathcal{F}_t$  adapted random processes taking values from  $U$ . This is the set of admissible controls.

For any  $t \in I \equiv [0, T]$ , let  $\xi(t) = x$  denote the state of the system given at time  $t$ . Let  $u \in \mathcal{U}_{ad}$  and suppose  $\xi_{t,x}(s), s \in (t, T]$ , is the unique strong solution of (5.1) starting from the state  $x$  at time  $t$  and driven by the control policy  $u$ . Define the functional

$$J(t, x, u) = E \int_t^T \ell(s, \xi_{t,x}(s)) ds + \Psi(\xi_{t,x}(T)), \quad t \in [0, T], \tag{6.1}$$

where  $\Psi$  is terminal cost, possibly representing penalty for failing to clear the traffic by the end of the observation or contract period. The value function is then given by

$$V(t, x) \equiv \inf_{u \in \mathcal{U}_{[t,T]}} J(t, x, u), \tag{6.2}$$

where  ${}^0u_{[t,T]}$  is the restriction of  ${}^0u_{\text{ad}}$  over the time interval  $[t, T]$ . This is the optimal cost corresponding to the starting time  $t \in [0, T]$  and the state  $x \in \mathbb{R}^n$ . Our goal is to determine the value function  $V$  and then, from this, find the optimal cost  $V(0, x)$  and the corresponding optimal control  $u^o \in {}^0u_{\text{ad}}$  if any.

In case  $B$  is a standard Brownian motion, with values in  $\mathbb{R}^m$ , following Bellman's principle of optimality, one can easily derive the so-called HJB equation which is given by

$$\begin{aligned} \frac{\partial V(t, x)}{\partial t} + \ell(t, x) + H(t, x, DV) + \left(\frac{1}{2}\right) \text{Tr}(D^2 V G G^*) &= 0, \\ V(T, x) &= \Psi(x), \end{aligned} \quad (6.3)$$

where

$$H(t, x, p) \equiv \inf_{v \in U} (p, f(t, x, v)). \quad (6.4)$$

Under certain regularity assumptions on the coefficients, one can prove the existence of a viscosity solution for (6.3). Given the existence of such a solution and a verification theorem, our original control problem will have a solution.

Unfortunately, the HJB equation given above has two major difficulties. Firstly, the coefficients  $\{\ell, F, G\}$  are discontinuous functions of the state variable  $x$ ; secondly, the matrix  $GG^*$  is singular. In other words, we have a degenerate HJB equation with discontinuous coefficients. We believe that such problems have not been considered in the literature so far. We leave this as an outstanding open problem.

In case  $B$  is a FBM, one may be able to construct an integro-partial differential equation for the value function which is more complex than the standard HJB equation.

In addition to these theoretical difficulties, there is also the computational complexity in solving the HJB equation in multidimensional space ( $n = 2m + 1$ ). In case we have  $m = 100$  users, we must solve the HJB equation (a nonlinear partial differential equation) in 201-dimensional real Euclidean space. This is beyond any super computer at this time. In view of this, we have chosen not to seek the optimum and, instead, look for simple, practically realizable but sufficiently efficient control strategies. We propose a simple feedback control law based on neural networks. Before we present this feedback law, we discuss some standard open-loop controls (OCs) used in current network management practice. We compare the performance of the feedback policy with those corresponding to open-loop strategies and show that the proposed feedback control offers superior performance compared with open-loop policies.

*Open-loop control.* In current practice [7], simple OC strategies with fixed token generation rates are used. Token generation rates are chosen based on statistical properties of the user traffic such as the mean rate, peak rate, and variance. In our numerical experiments, we choose such rates as controls and compute the corresponding performance and then compare the results with those corresponding to the proposed feedback law.



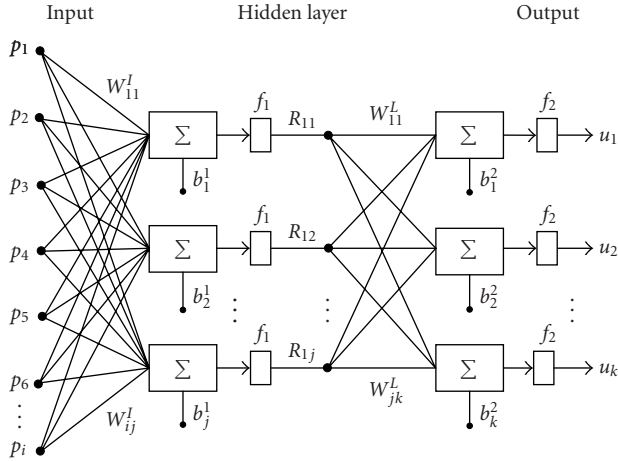


Figure 6.1. Neural network structure.

*Feedback control based on neural network.* Neural network (Figure 6.1) has been found to be a powerful tool in the design of feedback controls for many physical systems. This may not produce optimal policies, but it can certainly improve performance without going into rigorous and complex methodologies such as HJB equations. A neural network is simply a nonlinear memoryless transformation of one set of variables, say  $x = \{x_\alpha, \alpha = 1, 2, \dots, \ell\}$ , into another, say  $u = \{u_r, r = 1, 2, \dots, m\}$ , with adjustable parameters,  $W = \{w_1, w_2, \dots, w_r\}$ ,  $v = n(\ell + 1) + m(n + 1)$ , called the weights. We consider a 2-layer neural network with  $\ell = (2m + 1)$  inputs,  $m$  outputs, and one hidden layer consisting of  $n$  neurons. This can be described by the following general expression:

$$u_r = g_r \left( \sum_{s=1}^n W_{s,r}^2 f_s \left( \sum_{\alpha=1}^{\ell} W_{\alpha,s}^1 x_\alpha + b_s^1 \right) + b_r^2 \right), \quad r = 1, 2, \dots, m, \quad (6.5)$$

where the functions  $\{f_s, g_r\}$  are the basic nonlinear elements and  $\{b_s^1, b_r^2\}$  are the biases constituting the neurons. The weight vector  $W$  is given by  $W \equiv \{W_{\alpha,s}^1, W_{s,r}^2, b_s^1, b_r^2, \alpha = 1, 2, \dots, \ell; s = 1, 2, \dots, n; r = 1, 2, \dots, m\}$  conveniently arranged as a vector of dimension  $n(m + \ell)$ . All these can be compactly described by the following weighted nonlinear transformation:

$$u \equiv N(W, x), \quad (6.6)$$

where  $u_r = N_r(W, x)$ ,  $r = 1, 2, \dots, m$ . Identifying  $x$  with the state vector  $(a, \rho, q)'$  and substituting the control law given by (6.6) into the state equation (5.1), we arrive at the closed-loop system given by

$$d\xi(t) = F(t, \xi(t), N(W, \xi(t))) dt + G(t, \xi(t)) dB, \quad t \geq 0. \quad (6.7)$$

Table 7.1. System configuration and parameters.

$T_i$	$Q$	$\Delta t$	$K$	$M$
1500 bytes	4000 bytes	0.001 s	4000	300

The corresponding cost functional (5.3) may now be written as a function of the weight vector  $W$ :

$$J(W) \equiv E \int_0^T \ell(t, \xi(W, t)) dt, \quad (6.8)$$

where now the state obviously depends on the choice of  $W$ . This is the functional to be minimized by an appropriate choice of the weight. In a previous paper [1], we have used SA algorithm to find the lowest cost and the corresponding weights. In this paper, we have used RRS technique which appears to be much faster as illustrated by our numerical results shown in Table 8.2. For details on the RRS algorithm, see [10].

## 7. Basic data used for simulation experiments

For numerical simulation, we assume the system has a simple scenario comprised of three traffic sources policed by three TBs with their (conforming) outputs multiplexed by a single multiplexor.

**7.1. System parameters used for simulation.** In all the numerical experiments, we make the following three assumptions. First, the traffic generated by the three independent sources has the same statistical characteristics such as mean rate and variance. Second, the unit of tokens is measured in terms of bytes, and one byte is consumed for one byte of traffic. Third, there is no feedback control delay.

Depending on the requirements of different applications, the weights chosen for TB losses, multiplexor loss, and penalty for delay can be taken differently. For some applications, it is preferable to drop packets at TBs rather than at the multiplexor. In our experiments, we give the largest weight to losses at the multiplexor. The losses at TB are weighted more heavily than those for delay. We choose  $\lambda_1(t) = 5$ ,  $\lambda_2(t) = 10$ , and  $\lambda_3(t) = 0.1$  for all  $i$ . System configuration and parameters are given in Table 7.1; here  $K$  represents the number of time intervals in a trace and  $M$  is the number of sample paths. State initialization is set as  $\rho_i(t_0) = 0$ ,  $i = 1, 2, 3$ ,  $q(t_0) = 0$ .

The structure of neural network that is used for the experiment is shown in Table 7.2.

**7.2. Specification of traffic traces.** Different 4-second (traffic) traces are used as our traffic sources. We use FBM to generate the traffic through the SDE (2.5). Based on Bellcore real traffic, we calculated the Hurst parameter to be 0.7625. This is then used as the Hurst parameter for the FBM which, in turn, determines the traffic via the SDE (2.5). Since we use the Monte Carlo method to calculate the expected values, the number of sample paths required for numerical experiments is crucial and has to be determined carefully.

Table 7.2. Neural network parameters.

Number of inputs	7
Number of outputs	3
Number of layers	2
Number of neurons in layer 1	3
Number of neurons in layer 2	3
Transfer function in layer 1	Tangent sigmoid
Transfer function in layer 2	Positive linear

Table 8.1. Cost versus control strategies.

Control strategies	Cost $J$			
	$C = 2\text{ M}$	$C = 3\text{ M}$	$C = 4\text{ M}$	$C = 5\text{ M}$
FB(SA)	48.4954	27.7329	9.6738	0
FB(RRS)	51.2574	27.7329	9.8481	0
OPC	71.4317	47.7816	13.1386	0.1486
OPM	95.4174	55.3937	13.3965	1.0167
OPP	96.4339	56.4101	14.5820	0.1465

This number is very strongly dependent on the variance of the traffic. The larger the variance, the larger the number of sample paths required. See [1] for details. For numerical computation, we found 300 sample paths to be sufficient.

## 8. Numerical results

In order to study the performance of the system and the feedback control law, we compute the objective functional (5.1) and other measures of performance given by the expected values (5.5) and (5.7). We consider three OC laws: open loop with token generation rate equal to the channel capacity, which is denoted by (OPC), open loop with mean incoming traffic rate (total mean incoming traffic rate for our experiment is 4.0994 Mbps), denoted by (OPM), and open loop with peak incoming traffic rate denoted by (OPP). The corresponding results are compared with those corresponding to the feedback control law.

The first two rows of Table 8.1 show, for four different channel capacities ( $C = 2, 3, 4, 5$  Mbps), the cost corresponding to feedback control law based on neural network optimized by use of SA and RRS, respectively. The last three rows show similar results associated with OC laws: OPC, OPM, and OPP as described above. It is clear from this table that for any given control strategy, the cost decreases as the channel capacity increases. Similarly, for any given channel capacity, the costs corresponding to feedback controls are much less than those corresponding to OCs.

Table 8.2. Cost versus congestion.

Control strategies	First time to congestion (ms)			Residence time (ms)		
	$C = 2\text{M}$	$C = 3\text{M}$	$C = 4\text{M}$	$C = 2\text{M}$	$C = 3\text{M}$	$C = 4\text{M}$
FB(RRS)	3578	3991	4000	310	10	0
OPC	34	44	1106	3871	3621	1627
OPM	16	30	1054	3985	3969	1747
OPP	16	30	1015	3985	3969	2006

Table 8.3. Computation time versus optimization methods.

Optimization methods	Computation time (h)			
	$C = 2\text{M}$	$C = 3\text{M}$	$C = 4\text{M}$	$C = 5\text{M}$
FB(SA)	20.43	20.23	20.06	19.98
FB(RRS)	3.88	3.84	3.87	3.88

The mean first time to congestion and mean residence time in the state of congestion for different control strategies and channel capacities are given in Table 8.2. For any given control strategy, the first time to congestion increases with increasing channel capacity, while the residence time decreases as expected. Also, we note that the feedback control reduces the chance (probability) of congestion by way of increasing the mean first time to congestion. Further, it reduces the mean residence time in the state of congestion. It is clear from these results that the performance of the feedback control law is superior to those of OCs.

Further, we observe that the cost corresponding to feedback control law, optimized using SA algorithm, is somewhat less than that optimized using RRS algorithm. However, the computation time (we used Matlab 6.5 and Pentium 4 1.8 G) for RRS algorithm is five times shorter than that of SA. This is a remarkable improvement compared with the corresponding results of our paper [1] where SA algorithm was used. Thus for high-dimensional problems arising from a large number of users connected to a large network, the RRS algorithm is most suitable. This is shown in Table 8.3.

We use RRS algorithm to search in the space of weights  $W \in \mathbb{R}^{n(m+\ell)}$  for the minimal cost and the corresponding weight  $W^o$ . This is done for two different situations: (i) feedback control based on full information  $\{a, \rho, q\}$  and (ii) feedback control based on partial information  $\{\rho, q\}$ . For case (i), we have used  $\{a, \rho, q\}$  as the input to the neural network, while for case (ii), the input is  $\{\rho, q\}$ . It is clear that case (i) requires elaborate source monitoring which is too costly. The optimal cost associated with case (ii), as shown in Figure 8.1, is inferior to that corresponding to case (i) but still superior to those corresponding to the open-loop strategies OPC, OPP, and OPM. This result is practically important since it uses only state feedback and does not require any direct measurement of the incoming traffic.

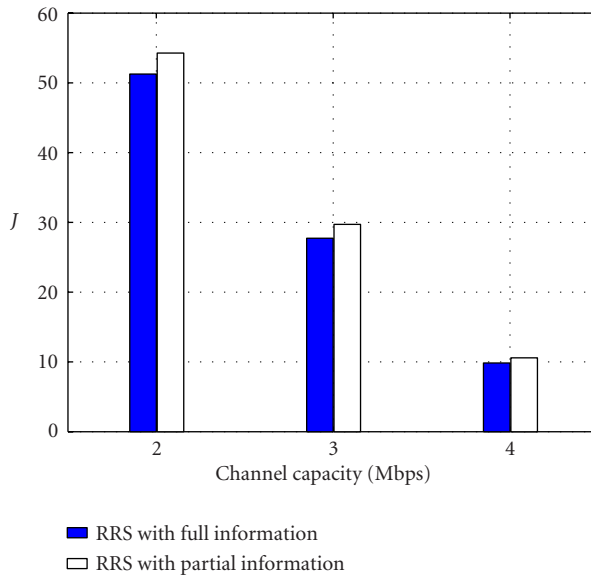


Figure 8.1. Cost comparison for full and partial information.

## 9. Conclusion

We have developed a continuous-time stochastic traffic model that captures the network traffic characteristics of self-similarity, LRD, and nonnegativity as observed in real networks. This model is then combined and coupled with the system describing the dynamics of the TB access control mechanism and the multiplexor. An objective functional reflecting the losses at the TBs and the multiplexor, including the service delay, is used to evaluate the performance of the system. We have proposed a feedback control law based on neural network with full or partial information. In case of full information, we use incoming traffic and the system state as the input and for partial information, we use only state as the input. The output of the neural network is the control. We have used SA algorithm and RRS algorithm to compute the optimum parameters determining the optimal feedback law. The numerical results clearly demonstrate that this feedback system can improve performance. Compared with SA algorithm, RRS algorithm has a slightly higher cost but can save significant computation time. To achieve optimum performance, one must solve the complex HJB equation. By using the proposed control law, we have avoided this complexity and achieved good performance.

## Acknowledgment

This work was partially supported by the National Science and Engineering Research Council of Canada under Grant A7109.

## References

- [1] N. U. Ahmed and C. Li, *Stochastic models for traffic dynamics and access control mechanism in computer network*, to appear in Engineering Simulation.
- [2] N. U. Ahmed and K. L. Teo, *Dynamic models for computer communication networks and their mathematical analysis*, Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms **9** (2002), no. 4, 507–524.
- [3] M. Crovella and A. Bestavros, *Self-similarity in world wide web traffic: evidence and possible causes*, IEEE/ACM Transactions on Networking **5** (1997), no. 6, 835–846.
- [4] S. Floyd and V. Paxson, *Difficulties in simulating the internet*, IEEE/ACM Transactions on Networking **9** (2001), no. 4, 392–403.
- [5] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), no. 4598, 671–680.
- [6] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, *On the self-similar nature of ethernet traffic (extended version)*, IEEE/ACM Transactions on Networking **2** (1994), no. 1, 1–15.
- [7] Computer Computer Ltd, *Quality of Service Solutions Configuration Guide, Policing and Shaping Overview*, Cisco 105 Release 12.0, 2002, [http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos\\_c/qcpart4/qcpolts.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm).
- [8] B. B. Mandelbrot and J. W. Van Ness, *Fractional Brownian motions, fractional noises and applications*, SIAM Rev. **10** (1968), 422–437.
- [9] V. Paxson and S. Floyd, *Wide area traffic: the failure of Poisson modeling*, IEEE/ACM Transactions on Networking **3** (1995), no. 3, 226–244.
- [10] T. Ye and S. Kalyanaraman, *A recursive random search for optimizing network protocol parameters*, Tech. report, ECSE Department, Rensselaer Polytechnic Institute, New York, 2002.

N. U. Ahmed: School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5

*E-mail address:* [ahmed@site.uottawa.ca](mailto:ahmed@site.uottawa.ca)

Cheng Li: School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5

*E-mail address:* [lichengca@yahoo.com](mailto:lichengca@yahoo.com)



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

