Mathematical Problems in Engineering, 2002, Vol. 8(3), pp. 265–279 Taylor & Francis Taylor & Francis Group

Systems Approach to Modeling the Token Bucket Algorithm in Computer Networks

N. U. AHMED*, QUN. WANG and L. OROZCO BARBOSA

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

(Received 2 July 2002; In final form 29 August 2002)

In this paper, we construct a new dynamic model for the Token Bucket (TB) algorithm used in computer networks and use systems approach for its analysis. This model is then augmented by adding a dynamic model for a multiplexor at an access node where the TB exercises a policing function. In the model, traffic policing, multiplexing and network utilization are formally defined. Based on the model, we study such issues as (quality of service) QoS, traffic sizing and network dimensioning. Also we propose an algorithm using feedback control to improve QoS and network utilization. Applying MPEG video traces as the input traffic to the model, we verify the usefulness and effectiveness of our model.

Key words: Computer network, dynamic models, traffic, token bucket, multiplexor, quality of service (QoS), feedback, control

AMS Subject Classification: 93A30, 93C55, 37AXX, 39B99

1 INTRODUCTION AND MOTIVATION

Currently, the most active and challenging research area in the Internet management and applications is to deliver real time applications, for instance, voice and video, with minimum network resources. The Internet Engineering Task Force (IETF) has proposed new network architectures such as Integrated Service (Intserv) [1], Differentiated Serve (Diffserv) [2], and Multiple Protocol Label Switch (MPLS) [3] in order to provide Quality of Service (QoS) to these new applications effectively and efficiently. In these architectures, there is a common mechanism, which includes admission control, access control, and flow control. It is interesting to note that the Token Bucket algorithm (we call TB in short hereafter) has been employed in all of them [4–8].

A TB is actually a traffic filter. There are two parameters: one, the token bucket size T, and two, the token rate u in the algorithm. Tokens, which are credits for a user to send a certain number of bytes in packets into the network, are filled into the bucket at a certain "rate". The bucket itself has a specified capacity, "token bucket size", over which newly arriving tokens are discarded. To transmit a packet, the algorithm will remove from the bucket a number of tokens equal to the packet size. If there are not enough tokens in the bucket to send a packet, the packet either waits until the bucket has enough tokens, or the packet is considered nonconforming, and it is either discarded or marked (the marked packets will be dropped

* Corresponding author. E-mail: ahmed@site.uottawa.ca

ISSN 1024-123X print; ISSN 1563-5147 online

once the network experiences congestion). If the bucket is already full, incoming tokens are rejected.

There has been extensive research on TB [4–7]. However, most of them consider only one aspect of the network to study the TB, and none of them presents a unified and consistent approach to the problem. In this paper, for the first time in the literature, we present a dynamic model (a state space model) for a system comprised of a multiplicity of TBs connected to a multiplexor and use systems approach for it's analysis. This approach provides a systematic tool and guideline for the study of TB and it's applications in computer networks.

The rest of the paper is structured as follows. In Section 2, we present a dynamic or equivalently a state space model for the system which consists of a number of TBs connected to a multiplexor. Using the systems approach, we define traffic policing, multiplexing and network utilization. In Section 3, we define objective functions in terms of QoS to a network provider and employ feedback controls to improve QoS and network utilization. We also propose a simple feedback algorithm (control law) to dynamically allocate available bandwidth based on available information. In Section 4, we use MPEG video traces as input traffic to verify the effectiveness of our model in terms of traffic sizing and network dimensioning. In Section 5, we present our conclusions and mention certain outstanding issues for future work.

2 SYSTEM MODEL

A TB has different functions at different levels of QoS provision. In admission control, the parameters of TB are used as a set of traffic descriptors for network management to make a decision in regards to acceptance or rejection of a traffic by comparing the effective bandwidth [8] of the traffic and the available bandwidth in the network. After the network accepts the traffic, TB works as a shaper to smooth the traffic in order to force it to conform to the contract between the user and the network [14]. Also, in order to police the traffic, the TB drops the nonconforming packets at the edge of the network. For bursty traffic, sometimes renegotiation [9] is necessary to achieve the acceptable QoS and network utilization. Thus, the traffic sources and the parameters of TB need to be adjusted dynamically by the feedback information from the network.

In this section, we first present our traffic model and then we present a new dynamic model for a single TB in terms of traffic policing at the edge of the network. It is easy to demonstrate the usefulness of this model alone in various applications. We use this basic model to construct a complete dynamic model for the network which consists of a multiplicity of token buckets (one for each source) accessing a multiplexor in an access node. We introduce an objective function that includes the losses at the multiplexor and those at the TB's. This is further augmented by adding a measure of service delay at the multiplexor. Finally, we discuss the trade off between QoS and network utilization to minimize this function.

2.1 Traffic Model

The concept of multi-serve integrated network, which is the scenario of future networks, is introduced to carry out various types of services: data, voice and video, all of which fall into two important categories of traffics: Constant Bit Rate (CBR) traffic and Variable Bit Rate Traffic (VBR). CBR traffic can be simply described by its rate since it sends packets at a fixed rate, and thus, peak rate allocation of bandwidth is applied for a congestion-free transmission. However, VBR traffic could be very bursty: average rate of transmission can be a small

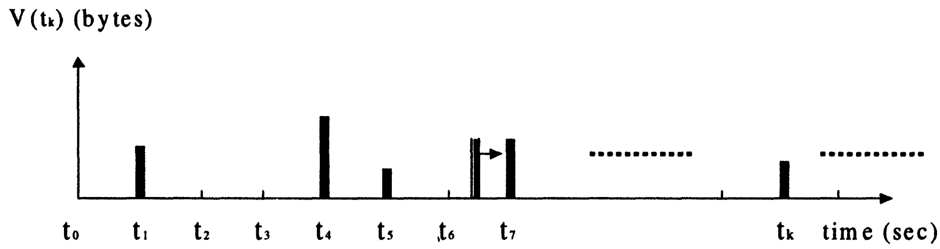

FIGURE 1 Traffic model.

fraction of the peak rate. Thus, peak rate allocation would result in under utilization of the system resources, while average rate allocation would result in losses.

In this paper, we consider a general traffic model $\{V(t_k), k = 0, 1, 2, \ldots\}$ as shown in Figure 1, which could be any one of data, voice, or video traffic. The input traffic denoted by, $\{V(t_k), k = 1, 2, \ldots\}$ is, in general, a random process observed at times t_k for $k = 1, 2, \ldots, K$. It gives the size of the arriving packet or a frame, that is, $V(t_k)$ is the size of the packet (or the frame) arriving at time t_k. The time intervals $[t_{k-1}, t_k)$ are the observation periods and we assume they are equal and small enough during which only one packet (or one frame) may arrive.

2.2 A Dynamic Model for TB

The TB that executes a policing function is a TB without a buffer as shown in Figure 2. When a packet arrives at the TB, if there are enough tokens in the bucket, the packet is considered conforming and a corresponding number of tokens is withdrawn from the token bucket. In case there are not enough tokens in the bucket, it is considered nonconforming and the token bucket stays the same. We denote by $u(t_k)$ the arriving tokens during the time interval $[t_{k-1}, t_k)$.

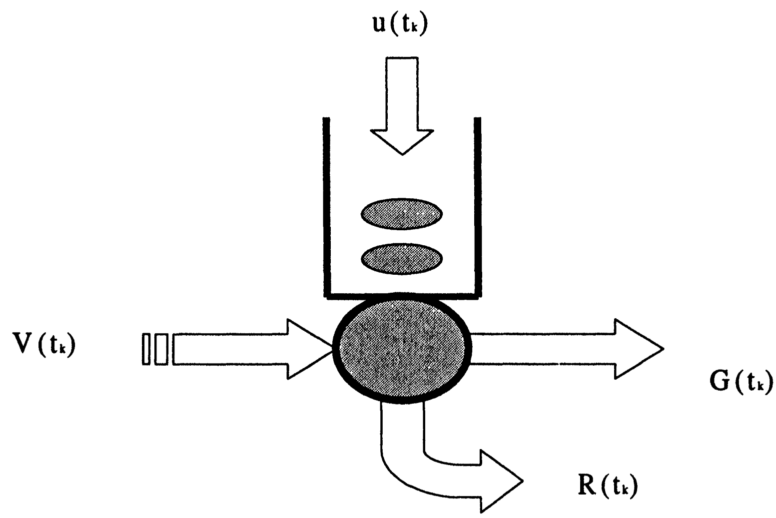

FIGURE 2 Token bucket model.

The following symbols are used throughout the paper and unit is measured in terms of bytes or bytes/sec when applicable:

1. $\{a \wedge b\} = \mathrm{Min}\{a, b\}$; $\{a \vee b\} = \mathrm{Max}\{a, b\}$;
2. I is an indicator function defined by:

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

3. T: physical size of TB;

The state of the token bucket at any time t is given by the number of tokens in the bucket at the time t and we denote this by $\rho(t)$. The evolution of token population with time is governed by the following difference equation which is based on a similar philosophy used in previous works on dynamic routing [12, 13]:

$$\begin{aligned} \rho(t_k) = {} & \rho(t_{k-1}) + \{u(t_k) \wedge [T - \rho(t_{k-1})]\} \\ & - V(t_k) I\{V(t_k) \le \rho(t_{k-1}) + [u(t_k) \wedge [T - \rho(t_{k-1})]]\}. \end{aligned} \tag{1}$$

This is fundamentally a balance equation, where the first term on the right hand side denotes the number of unused tokens present at the previous time instance t_{k-1}, the second term represents the number of tokens received, more precisely, accepted by the bucket during the time period $[t_{k-1}, t_k)$, and finally the last term represents the number of tokens consumed at time t_k. The balance is the current token population. The quantity $u(t_k)$ represents the number of tokens transmitted to the TB. This is treated as a control variable and it is determined by the network management on the basis of the status of the network. This is further discussed as we introduce the network model.

Thus the conforming traffic, denoted by $G(t_k)$ at time t_k, is given by:

$$G(t_k) = V(t_k) I\{V(t_k) \le \rho(t_{k-1}) + [[u(t_k)] \wedge [T - \rho(t_{k-1})]]\} \tag{2}$$

where the item $[u(t_k) \wedge [T - \rho(t_k)]]$ represents the number of tokens accepted by the TB during the time interval $[t_{k-1}, t_k)$ and the item within the parenthesis, $\{\cdots\}$, represents the condition that the size of the arriving packet is not larger than the number of tokens available in the TB at time t_k.

The nonconforming traffic $R(t_k)$ is then given by:

$$R(t_k) = V(t_k) - G(t_k). \tag{3}$$

2.3 Model of Network Scenario

Now we consider an extension of the model to a network scenario as shown in Figure 3. A number of sources are assumed to share the same multiplexor in an access node which has a finite buffer size Q and service rate C. The TBs at the edge of the network shape the traffic by dropping the nonconforming packets. There is a controller at the access node dynamically controlling the access rate of each traffic by adjusting traffic sources or the token supply at each TB to achieve maximum utilization of the multiplexor under the constraint that the QoS of each accessing traffic is satisfied. This scenario has very practical meaning for current networks. This provides QoS by dynamically allocating bandwidth to each of the users in the network to improve utilization.

Suppose there are n sources attempting to access the multiplexor. Access of each source to the multiplexor is controlled by a distinct TB dedicated for the source. Each user is allocated

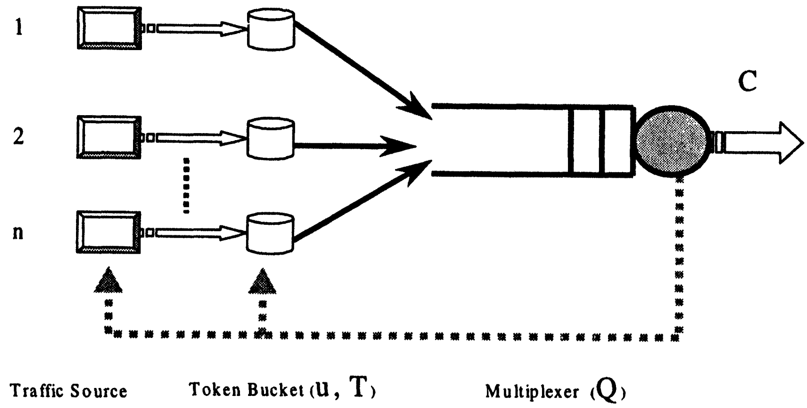

FIGURE 3 System model.

a bandwidth on the basis of its request and the network resources. The state of the system is now described by the status of all the TBs and the multiplexor. The status of the token buckets as discussed before is now given by a vector valued function $\rho \equiv (\rho_1, \rho_2, \ldots, \rho_n)'$ of dimension n as follows:

$$\begin{aligned}
\rho_1(t_k) &= \rho_1(t_{k-1}) + \{u_1(t_k) \wedge [T_1 - \rho_1(t_{k-1})]\} \\
&\quad - V_1(t_k)I\{V_1(t_k) \leq \rho_1(t_{k-1}) + [u_1(t_k) \wedge [T_1 - \rho_1(t_{k-1})]]\} \\
\rho_2(t_k) &= \rho_2(t_{k-1}) + \{u_2(t_k) \wedge [T_2 - \rho_2(t_{k-1})]\} \\
&\quad - V_2(t_k)I\{V_2(t_k) \leq \rho_2(t_{k-1}) + [u_2(t_k) \wedge [T_2 - \rho_2(t_{k-1})]]\} \\
&\;\;\vdots \\
\rho_n(t_k) &= \rho_n(t_{k-1}) + \{u_n(t_k) \wedge [T_n - \rho_n(t_{k-1})]\} \\
&\quad - V_n(t_k)I\{V_n(t_k) \leq \rho_n(t_{k-1}) + [u_n(t_k) \wedge [T_n - \rho_n(t_{k-1})]]\}.
\end{aligned} \tag{4}$$

This is a system of difference equations describing the dynamics of token population in each of the TBs which are similar to Eq. (1). In these equations there are also expressions for the conforming traffic. Again the conforming traffic is also given by a vector valued function of dimension n, which we denote by $G \equiv (G_1, G_2, \ldots, G_n)'$. These are given by:

$$\begin{aligned}
G_1(t_k) &= V_1(t_k)I\{V_1(t_k) \leq \rho_1(t_{k-1}) + [u_1(t_k) \wedge [T_1 - \rho_1(t_{k-1})]]\} \\
G_2(t_k) &= V_2(t_k)I\{V_2(t_k) \leq \rho_2(t_{k-1}) + [u_2(t_k) \wedge [T_2 - \rho_2(t_{k-1})]]\} \\
&\;\;\vdots \\
G_n(t_k) &= V_n(t_k)I\{V_n(t_k) \leq \rho_n(t_{k-1}) + [u_n(t_k) \wedge [T_n - \rho_n(t_{k-1})]]\}.
\end{aligned} \tag{5}$$

Thus the vector of nonconforming traffic is given by:

$$\begin{aligned}
R_1(t_k) &= V_1(t_k) - G_1(t_k) \\
R_2(t_k) &= V_2(t_k) - G_2(t_k) \\
&\;\;\vdots \\
R_n(t_k) &= V_n(t_k) - G_n(t_k).
\end{aligned} \tag{6}$$

Again these equations are similar to the Eqs. (2) and (3). These vectors are used later in formulating the objective functional.

In order to complete the dynamics of the whole system we must now formulate the status of the multiplexor. Let $q(t)$, at time t, denote the size of the queue at the multiplexor waiting for service, that is, for onward transmission into the network. This variable is taken as the state of the multiplexor. It is governed by the following difference equation:

$$q(t_k) = \{[q(t_{k-1}) - C\tau] \vee 0\} + \left\{ \left[\sum_{i=1}^{n} G_i(t_k) \right] \wedge [Q - [[q(t_{k-1}) - C\tau] \vee 0]] \right\}, \tag{7}$$

where τ denotes the length of the time intervals $[t_{k-1}, t_k)$, $k = 1, 2, \ldots, K$. The first term on the right hand side of the above expression describes the leftover traffic at time t_k after the multiplexor has injected traffic into the network during the period $[t_{k-1}, t_k)$. The second term represents the traffic accepted from all the sources during the same period. It is clear from this expression that the traffic accepted by the multiplexor is given by the smaller of the available (empty) space in the buffer and the sum of all the conforming traffic. Thus the complete system dynamics is given by the set of $n + 1$ difference Eqs. (4) and (7). Even though the system of equations (4) appear to be mathematically uncoupled, they are actually coupled through Eq. (7) which contains all the conforming traffic from all the TBs. Further, coupling is also present through the controls which are dependent on the status of the multiplexor and other network variables such as the incoming traffic and the status of the TBs. This will be clear later as we define the (feedback) control strategies.

It is expected that the multiplexor may not always be able to serve all the conforming traffic because of its buffer size limitation Q and the bandwidth limitation C of the outgoing link. Thus the traffic lost at (or rejected by) the multiplexor is given by:

$$L(t_k) = \left[\sum_{i=1}^{n} G_i(t_k) \right] - \left\{ \left[\sum_{i=1}^{n} G_i(t_k) \right] \wedge [Q - [[q(t_{k-1}) - C\tau] \vee 0]] \right\}. \tag{8}$$

From this equation and Eq. (3), we may define network utilization as:

$$\eta = \frac{\sum_{i=1}^{n} \sum_{k=0}^{K} V_i(t_k) - [\sum_{k=0}^{K} L(t_k) + \sum_{i=1}^{n} \sum_{k=0}^{K} R_i(t_k)]}{C(t_K - t_0)}. \tag{9}$$

This expression will be used later in our numerical simulation experiments to present the level of network utilization under different (token) control policies. Another measure of network efficiency is the (average) throughput which is intimately related to the network utilization as defined above.

$$\begin{aligned}
\Upsilon = C\eta &= \frac{\sum_{i=1}^{n} \sum_{k=0}^{K} V_i(t_k) - [\sum_{k=0}^{K} L(t_k) + \sum_{i=1}^{n} \sum_{k=0}^{K} R_i(t_k)]}{(t_K - t_0)} \\
&= \frac{1}{t_K - t_0} \sum_{k=0}^{K} \Upsilon(t_k) \\
&= \frac{1}{t_K - t_0} \sum_{k=0}^{K} \left\{ \left[\sum_{i=1}^{n} G_i(t_k) \right] \wedge [Q - [[q(t_{k-1}) - C\tau] \vee 0]] \right\},
\end{aligned}$$

where $\Upsilon(t_k)$ denotes the throughput during the period $[t_{k-1}, t_k)$, $k \geq 1$.

3 OBJECTIVE FUNCTIONS AND TRAFFIC CONTROL

The objective functional for a network provider is given by:

$$J(u) = \sum_{k=0}^{K} \alpha(t_k)L(t_k) + \sum_{i=1}^{n}\sum_{k=0}^{K} \beta_i(t_k)R_i(t_k) + \sum_{k=0}^{K} \gamma(t_k)q(t_k) \tag{10}$$

where $u = (u_1, u_2, \ldots, u_n)'$ is the control vector that appears in the dynamic model of the TBs. The parameters $\{\alpha, \beta_i, \gamma, i = 1, 2, \ldots, n\}$ are nonnegative functions of time assigning relative weights given to various losses as defined earlier. The first term penalizes losses in the multiplexor, the second term penalizes lost traffic at the TB's and the third term is an approximate measure of waiting time or delay at the multiplexor before being served. The problem is to find a control policy that minimizes this function. As regards control, there are two options: the controls can be open loop or feedback. In the case of feedback controls, one may visualize this as a function of the state of the system and the user demand (incoming traffic). Therefore the control vector may be given by:

$$u(t_k) = \Gamma(V(t_k), q(t_{k-1}), \rho(t_{k-1})) \tag{11}$$

for $k \in \{1, 2, 3, \ldots, K\}$, where Γ is a suitable feedback control law.

3.1 Optimal Control

To find an optimal control law $u(t_k)$, one must minimize the functional (10) subject to constraints on the control. In general the control constraint set, denoted by U, is itself a function of the state variables $\{\rho, q\}$ and the input traffic vector V. Since the controls are n-tuples of nonnegative integers, it is clear that at any time, say, t_k, $U \equiv U(V(t_k), \rho(t_{k-1}), q(t_{k-1}))$ is a subset of N_0^n. Thus the optimal control problem is to choose at time t_k, for every $k \in D \equiv \{0, 1, 2, \ldots, K\}$, a control $u(t_k) \in U(V(t_k), \rho(t_{k-1}), q(t_{k-1}))$ that minimizes the functional (10) subject to the dynamic constraints (4) and (7). This is a difficult problem which can be solved by use of dynamic programming principle and genetic algorithm. We address this problem in a later paper. In this paper we consider both open loop and feedback controls without optimization. We propose a simple proportional feedback (control) law which is much easier to implement compared to complex control laws that may result from rigorous optimization.

3.2 Open Loop Controls

In Section 4, we consider five different cases of simple open loop controls based on the statistical properties of the traffic such as the mean rate, peak rate and the variance. The corresponding results are discussed in details in that section.

3.3 Feedback Controls

Here we propose some simple feedback control laws which are easy to implement. To demonstrate the usefulness and effectiveness of our model, we compare the results corresponding to the feedback control law with those corresponding to open loop controls.

Towards this goal, we consider some special versions of the cost functional (10). Assume that all the users (sources) can manage traffic at different QoS levels and that minimum QoS is guaranteed by admission control. Furthermore, we set $\alpha(t_k)$ in Eq. (10) equal to infinity in order to eliminate losses in the multiplexor which may cause unpredictable QoS for each user. This means that losses at the multiplexor must be forced to equal zero. We achieve this by imposing appropriate constraints on the controls as presented below.

3.3.1 Cost Functional J_1

Here we assume that the delay in the multiplexor is negligible. The objective function is then given by

$$J_1(u) = \sum_{k=0}^{K} \alpha(t_k) L(t_k) + \sum_{i=1}^{n} \sum_{k=0}^{K} \beta_i(t_k) R_i(t_k). \tag{12}$$

This is the cost associated with traffic losses at the token buckets and the multiplexor. Since by choice of appropriate feed back control law we eliminate losses at the multiplexor, the first term is effectively zero. We consider two cases, one without priority, and one with priority.

Case A (Without Priority) In order to reduce the losses in the multiplexor to zero, we define

$$\Theta_i(t_k) \equiv \frac{V_i(t_k)\{Q - [(q(t_{k-1}) - C\tau) \vee 0]\}}{\sum_{i=1}^{n} V_i(t_k)} \tag{13}$$

as being the maximum allowable token permit for the ith user. This can also be interpreted as being the maximum permissible allocation of multiplexor (buffer) space to the ith user. Note that no priority is given to any of the users. Resources are allocated proportional to their demands. Weighting this against the actual traffic or demand, we obtain the true allocation

$$A_i(t_k) = \{\Theta_i(t_k) \wedge V_i(t_k)\}. \tag{14}$$

Note that

$$\sum_{i=1}^{n} A_i(t_k) \leq \{Q - [(q(t_{k-1}) - C\tau) \vee 0]\} \tag{15}$$

and therefore the losses in the multiplexor is completely eliminated for all $\{t_k\}$. Thus we can now define the control policy as

$$u_i(t_k) = \{[A_i(t_k) - \rho_i(t_{k-1})]\} I\{A_i(t_k) \geq \rho_i(t_{k-1})\}. \tag{16}$$

The term inside the large bracket of the expression (16) gives the number of tokens needed at time t_k which may be positive or negative. The actual provision made to the ith user is then given by multiplying this quantity by the indicator function as shown. Thus if the allocation $A_i(t_k)$ exceeds the number of existing tokens in the bucket, then the user is supplied with the difference only, otherwise no token provision is necessary. This is realized by use of the indicator function.

Case B (With Priority) Some time it is necessary to assign special priorities to some users depending on various factors; such as the source type: like data, video, audio etc. or simply because, some users are willing to pay for the time and the volume of traffic. In this case we modify our definition of proportional allocation given by (13) to priority allocation defined by the following expression,

$$\hat{\Theta}_i(t_k) = \frac{\delta_i(t_k)V_i(t_k)\{Q - [(q(t_{k-1}) - C\tau) \vee 0]\}}{\sum_{i=1}^{n} \delta_i(t_k)V_i(t_k)} \tag{17}$$

where

$$0 \leq \delta_i(t_k) \leq 1, \ \text{ and } \sum_{i=1}^{n} \delta_i(t_k) = 1, \quad \forall\ t_k.$$

Note that the priority index δ may vary with time from user to user. For example, if we wish to provide absolute priority to user j, for certain periods of time, say, P, we set

$$\delta_j(t_k) = 1, \quad \forall\ t_k \in P, \ \ P \subset \{t_r, r = 1, 2, 3, \ldots, K\}.$$

Again we can define the true allocation as in (14) by

$$\hat{A}_i(t_k) = \{\hat{\Theta}_i(t_k) \wedge V_i(t_k)\}, \tag{18}$$

and note that, exactly as in (15), once again we have

$$\sum_{i=1}^{n} \hat{A}_i(t_k) \leq \{Q - [(q(t_{k-1}) - C\tau) \vee 0]\}, \tag{19}$$

ensuring zero loss at the multiplexor. In this case the control law given by (16) is slightly modified as presented below,

$$\hat{u}_i(t_k) = \{[\hat{A}_i(t_k) - \rho_i(t_{k-1})]\} I\{\hat{A}_i(t_k) \geq \rho_i(t_{k-1})\}. \tag{20}$$

Remark In order to prevent excessive monopoly by any one of the n users, the network provider may put a ceiling on the allocation fraction by replacing the simple fraction by a regulated one as follows:

$$\frac{V_i(t_k)}{\sum_{i=1}^{n} V_i(t_k)} \Longrightarrow \left(\frac{V_i(t_k)}{\sum_{i=1}^{n} V_i(t_k)}\right) \wedge \left(\frac{\mu_i(t_k)}{n}\right)$$

where $1 \leq \mu_i(t_k) \leq n$ can be chosen by the network provider.

3.3.2 Cost Functional $J_2 = J$

Here we include the waiting time or equivalently, the delay. The objective function is now given by

$$J_2(u) \equiv J = \sum_{k=0}^{K} \alpha(t_k)L(t_k) + \sum_{i=1}^{n}\sum_{k=0}^{K} \beta_i(t_k)R_i(t_k) + \sum_{k=1}^{K} \gamma(t_k)q(t_k). \tag{21}$$

Again we consider both the cases, without priority and with priority. The control laws for these cases remain exactly the same as in the previous subsection, that is Eqs. (16) and (20). Since we have now included the delay, we expect the cost functional $J_2 \geq J_1$.

3.4 Cost Functional J as a Function of C and Q

In the preceding sub sections we have proposed certain feedback control laws that eliminate multiplexor losses. We can now compute the cost functionals corresponding to these control laws. Since these control laws are now fixed by the expressions (16) and (20), we may consider the cost functionals as functions of the network parameters such as the multiplexor size Q and the link capacity C. Thus in this section we consider J_1 as a function of Q and C and denote this by $J_1(C, Q)$. Similarly we consider J_2 also as a function of these parameters and denote this by $J_2(C, Q)$. For a given link capacity, say, C we may plot $J_{1,C}(Q)$, $J_{2,C}(Q)$ as functions of the multiplexor size Q and similarly, for a given multiplexer size, say, Q, we plot $J_{1,Q}(C)$, $J_{2,Q}(C)$ as functions of C. Finally we shall present 3-d plots of these functions for several traffic profiles. We discuss these results in details in the numerical section. These results will clearly demonstrate the usefulness of the dynamic model and the proposed control laws.

4 NUMERICAL RESULTS

To demonstrate the usefulness and effectiveness of our models, we implement these along with the feedback control algorithm in a network simulation software, OPNET. We consider a scenario comprised of three traffic sources policed by three TBs whose conforming outputs are multiplexed into a single buffer.

We choose three independent five-minute MPEG video traces from a cartoon, movie, and sports as our traffic sources. Table I summarizes the specification of the traces. For convenience, we denote by $\{P_i, M_i, \Sigma_i\}$ the peak rate, the mean, and the standard deviation of the ith traffic source.

System configurations: • $T_i = \Sigma_i,\ \ i = 1, 2, 3$; • $C \equiv \sum_{1 \leq i \leq 3} M_i = 1.79$ kbytes; • $Q =$ 3 Mbytes; • $\tau = 0.5$ seconds (interval of GoP[1]);

State initialization: • $\rho_i(t_0) = T_i$; • $q(t_0) = 0$;
Relative weights: • In regards to the weights given to the various components of the cost functional J, we choose $\alpha(t_k) = 10$, for all t_k, $\beta_i(t_k) = 1$ for all $\{i, t_k\}$, and $\gamma(t_k) = 1$ for all k. This choice puts ten times more weight on multiplexor losses compared with other losses. We compute the cost functions J_1 and J_2 corresponding to these weights.

4.1 Dependence of Cost on Control Strategies

We compare the cost function corresponding to the feedback control, as given by (16), with those corresponding to open loop controls for different configurations.

TABLE I Summary of Video Traces Specification.

	Peak rate (P_i) (Mbytes/sec)	*Average rate (M_i) (Mbytes/sec)*	*Standard deviation (Σ_i)*
Trace 1	4.32	0.64	0.65
Trace 2	2.18	0.28	0.14
Trace 3	5.64	0.87	0.64

[1]Group of Picture (GoP): MPEG video trace is broken into GoPs and each constrains 15 video frames. In NTSC, the video is transmitted at 30 frames per second.

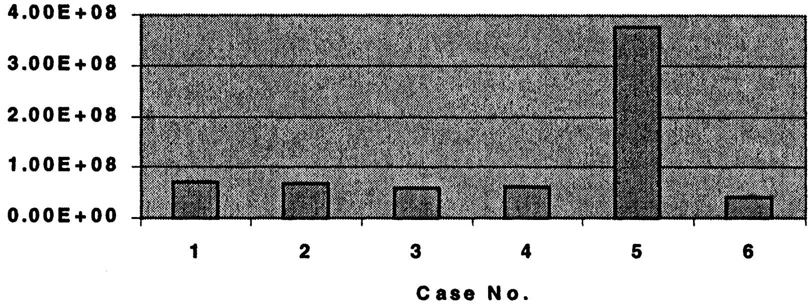

FIGURE 4 Cost function J_1.

Cases Studied
We consider 5 cases of open loop control strategies and one case of feedback control as described below.

Open loop control: • Case 1: $u_i(t_k) = M_i$; • Case 2: $u_i(t_k) = 1.01M_i$; • Case 3: $u_i(t_k) = 1.04M_i$; • Case 4: $u_i(t_k) = 1.05M_i$; • Case 5: $u_i(t_k) = P_i$ for $i \in \{1, 2, 3\}$;
Feedback Control: • Case 6 with control law given by Eq. (16).
Remark 1. The service rate of the multiplexor C (Link capacity) could be larger in reality; here we want to study the performance of the system in a heavy-load situation; 2. The buffer size of a multiplexor is determined by delay constraints. Note that usually larger buffer causes greater delay.

Figure 4 gives a plot of the cost functional J_1 corresponding to all the six cases (1–6). Clearly the result shows that the case 6 which corresponds to the feedback control offers the minimum cost. The open loop control provided by the peak rate, Case 5, is the worst of all. These simple open loop controls are either too "aggressive (Case 5)" causing losses in the multiplexor or too "conservative (Cases 1–4)" degrading QoS by under-utilization of available network resources.

Next we consider the full cost functional $J_2 = J$ which adds the multiplexor delay to J_1. Obviously $J_2 \geq J_1$. Figure 5 illustrates a different scenario. The cost corresponding to Case 6, associated with the feed back control, is now larger compared to those of Cases (1–4) but less than that of Case 5. This is because the open loop strategies do not allow sufficient access to the multiplexor (too conservative) thereby reducing multiplexor losses while degrading throughput as seen in Figure 6. Note that the throughput for all the Cases (1–4) is lower than

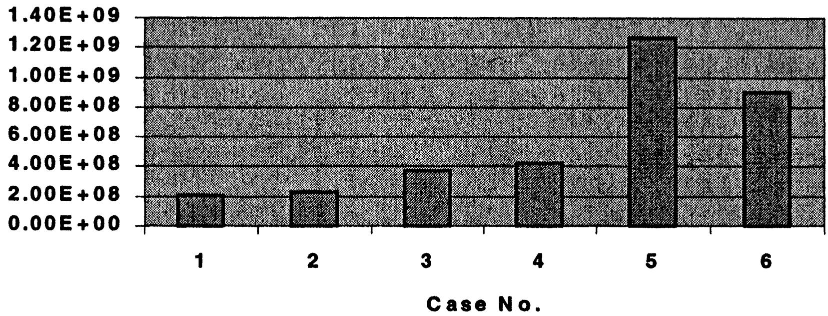

FIGURE 5 Cost function J_2.

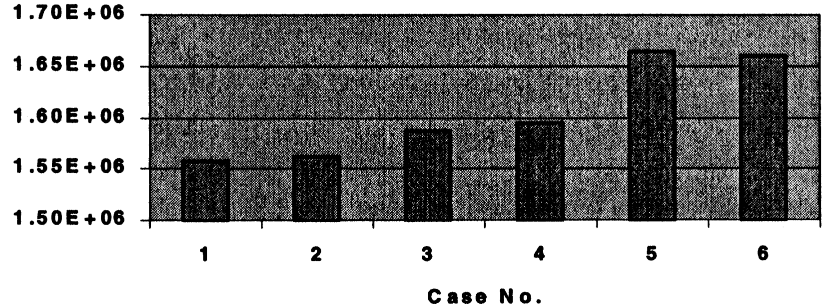

FIGURE 6 Throughput (bytes/sec).

those of Case 5 and Case 6. Comparing these figures with respect to the Cases 5 (peak control) and 6 (feedback control), it is clear that Case 6 beats 5 in terms of cost (Fig. 5) while the throughput (Fig. 6) corresponding to Case 6 is very close to that of Case 5. In case of video transmission this means that the feedback control (Case 6) maintains QoS as well as throughput, while the open loop control (Case 5) degrades QoS though it maintains throughput.

4.2 Dependence of Cost on Network Parameters

Next we study how the network design parameters such as (output) link capacity C and buffer size Q of the multiplexor affect the cost functions J_1 and J_2 for the fixed feedback control strategy given by (16).

Figure 7 presents a plot of the cost function J_1 as a function of Q ($J_{1,C}(Q)$) corresponding to different but fixed link capacities C, and Figure 8 presents the cost function J_1 as a function of C ($J_{1,Q}(C)$) for different values of the multiplexor size Q. As expected, increasing buffer size Q and link capacity C decreases the cost. It is also clear from Figure 7 that

$$C_1 \leq C_2 \leq C_3 \Longrightarrow J_{1,C_1}(Q) \geq J_{1,C_2}(Q) \geq J_{1,C_3}(Q), \quad \forall\, Q > 0. \tag{22}$$

As Q is increased beyond certain value which is sufficient to cover the peaks and bursts, the cost reaches a plateau, here zero, thereby eliminating all losses and QoS degradation. Similarly it is clear from Figure 8 that

$$Q_1 \leq Q_2 \leq Q_3 \Longrightarrow J_{1,Q_1}(C) \geq J_{1,Q_2}(C) \geq J_{1,Q_3}(C), \quad \forall\, C > 0. \tag{23}$$

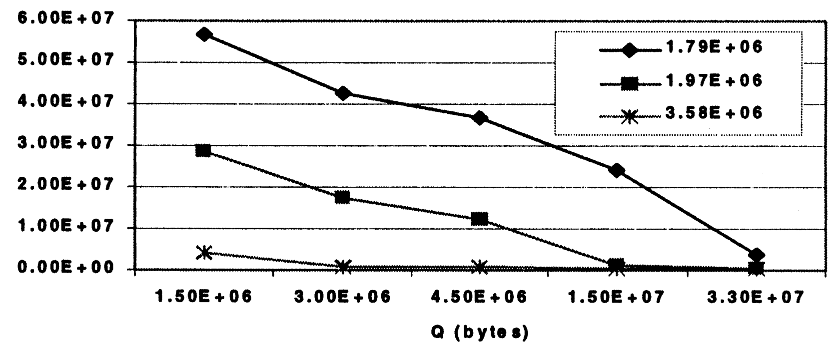

FIGURE 7 Cost function $J_1(Q)$ with different Cs.

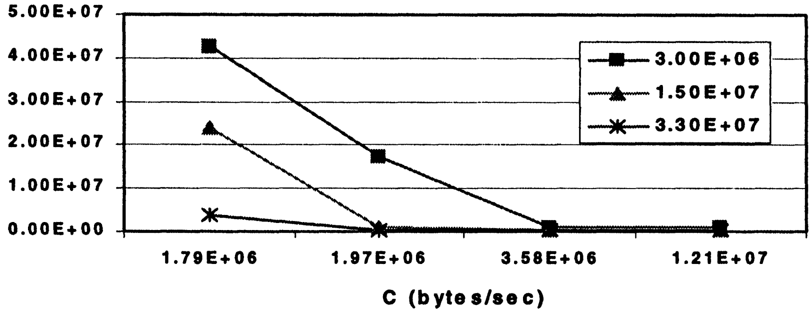

FIGURE 8 Cost function $J_1(C)$ with different Qs.

In this case also similar conclusions hold. Now we consider the full cost function $J_2 = J$. Figure 9 shows the cost function J_2 as a function of Q for different values of capacity C, giving a different profile as compared to Figures 7 and 8. However the order of dominance remains the same as given below.

$$C_1 \leq C_2 \leq C_3 \Longrightarrow J_{2,C_1}(Q) \geq J_{2,C_2}(Q) \geq J_{2,C_3}(Q), \quad \forall\, Q > 0. \tag{24}$$

A correct explanation for the increasing cost with increasing Q, for any fixed C, is given as follows. When the link capacity C is small, increasing multiplexor size Q only increases

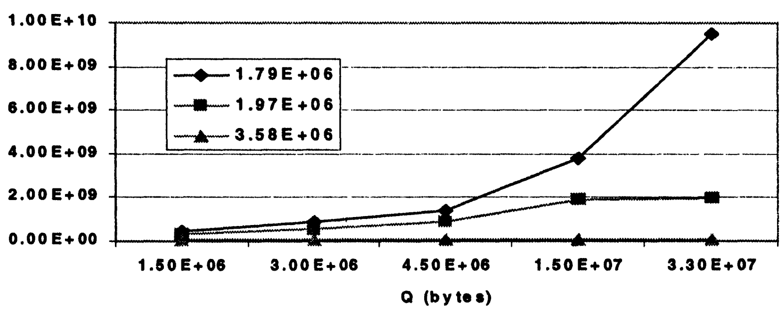

FIGURE 9 Cost function $J_2(Q)$ with different Cs.

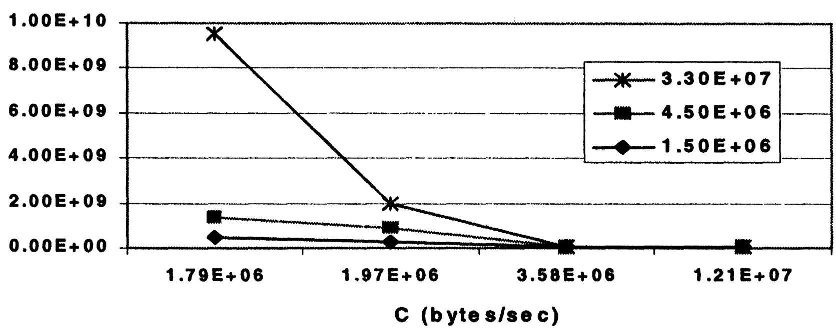

FIGURE 10 Cost function $J_2(C)$ with different Qs.

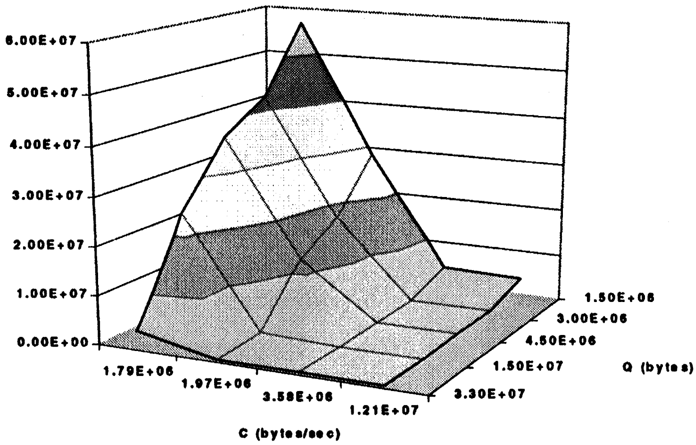

FIGURE 11 Cost function $J_1(C, Q)$.

queue size and the delay and hence the cost. Thus for any fixed C, $J_{2,C}(Q)$ is a nondecreasing function of Q.

Again, Figure 10 illustrates the cost function J_2 as a function of C for different values of Q. Note that we have similar order relation

$$Q_1 \leq Q_2 \leq Q_3 \Longrightarrow J_{2,Q_1}(C) \leq J_{2,Q_2}(C) \leq J_{2,Q_3}(C), \quad \forall Q > 0. \tag{25}$$

Here $J_{2,Q}(C)$ is a decreasing function of C. This follows from the fact that, for any fixed Q, increasing link capacity C increases the service rate reducing the delay and thereby reducing the cost.

The pictures presented by the Figures 11 and 12 give three dimensional plots of the cost functionals $J_1(C, Q)$, $J_2(C, Q)$ considered as functions of the two network parameters Q and C. These figures are self explanatory.

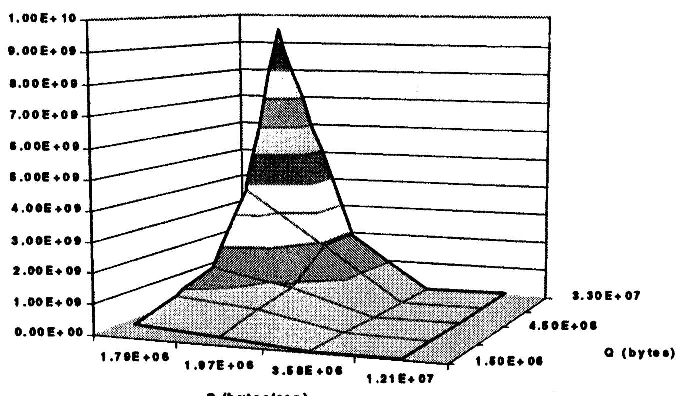

FIGURE 12 Cost function $J_2(C, Q)$.

5 CONCLUSION

The dynamic models, we have presented in this paper, formally define the interactions between the elements of any TB-based traffic policing and multiplexing system at the edge of a computer network. This gives a simple and scalable infrastructure to the problems of TB control mechanism and related issues. By applying the models to network simulation, we find that the results from simulation perfectly agree with the numerical results provided by our dynamic models. Thus the models also give a guideline to traffic sizing, network dimensioning and trade-off over losses, delay and network utilization etc. The future work based on these models could fall into three areas: first, employ feasible optimization techniques to obtain optimal feedback control strategies. The second: extend the models to a complete system of interconnected network and study end-to-end performance. The third, use the models to analyze current network architectures and applications.

References

[1] Braden, R., Clark, D. and Shenker, S. (1994) Integrated services in the internet architecture: An overview, RFC 1633.
[2] Blake, S., Clark, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998) An architecture for differentiated service, RFC 2475.
[3] Rosen, E., Viswanathan, A. and Callon, R. (2001) Multi protocol label switching architecture, RFC 3031.
[4] Shenker, S. and Wroclawski, J. (1997) General Characterization Parameters for Integrated Service Network Elements, RFC 2215.
[5] Heinanen, J. and Guerin, R. (1999) A single rate three color marker, RFC 2697.
[6] Heinanen, J. and Guerin, R. (1999) A two rate three color marker, RFC 2698.
[7] Aboul-Magd, O. and Jamoussi, B. (1999) InformationalA Framework for Service Definition and Inter working Using CR-LDP, Internet Draft, draft-aboulmagd-srvc-def-crldp-00.txt, October 1999.
[8] Butto, M., Cavallero, E. and Tonietti, A. (1991) Effectiveness of the leaky bucket policing mechanism in ATM networks, *IEEE Journal on Selected Areas in Communications*, **9**(3).
[9] Berger, A. W. (1991) Performance Analysis of a Rate Control where tokens and jobs queue, *IEEE Journal on Selected Areas in Communications*, **9**(2).
[10] Ahmadi, H., Guerin, R. and Sohraby, K. (1990) Analysis of leaky bucket access control mechanism with Bacth arrival process, *Globecom* **90**.
[11] Yin, N. and Muchyj, M. G. (1991) Analysis of the leaky bucket algorithm for on-off data sources, *Globecom* **91**.
[12] Wahida, C. and Ahmed, N. U. (1992) Congestion control using dynamic routing and flow control, *Stochastic Analysis and Applications*, **10**(2), 123–142.
[13] Ahmed, N. U., Dabbous, T. E. and Lee, Y. E. (1988) Dynamic routing for computer queuing networks, *Int. J. Systems SCL*, **19**(6).
[14] Tanenbaum, A. S. (1966) *Computer Networks* (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall PTR.