Research Article

Suboptimal RED Feedback Control for Buffered TCP Flow Dynamics in Computer Network

N. U. Ahmed and X. H. Ouyang

Received 24 November 2006; Accepted 5 February 2007

Recommended by Semyon M. Meerkov

We present an improved dynamic system that simulates the behavior of TCP flows and active queue management (AQM) system. This system can be modeled by a set of stochastic differential equations driven by a doubly stochastic point process with intensities being the controls. The feedback laws proposed monitor the status of buffers and multiplexer of the router, detect incipient congestion by sending warning signals to the sources. The simulation results show that the optimal feedback control law from the class of linear as well as quadratic polynomials can improve the system performance significantly in terms of maximizing the link utilization, minimizing congestion, packet losses, as well as global synchronization. The optimization process used is based on random recursive search technique known as RRS.

Copyright © 2007 N. U. Ahmed and X. H. Ouyang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Since random early detection (RED) was presented by Sally Floyd and Van Jacobson in 1993 as a congestion avoidance mechanism in packet-switched networks [1], some variants of RED have been designed to deal with the setting of various parameters characterizing it [2]. The basic principle of RED-family algorithm consists of the following steps: (1) active queue management (AQM) mechanism detects incipient congestion by computing the average queue size, (2) notifies connections of congestion either by dropping packets or by setting a bit in packet headers. In [3], we proposed a modified version of RED governed by doubly stochastic Poisson-driven stochastic differential equations. The dynamic model developed there [3] contains a suboptimal feedback control law which monitors
the status of queue of the (multiplexer) router and adjusts or controls the dropping in-
tensity by sending (congestion) warning signals to the sources. The major advantage of
the algorithm is that the critical parameters suggested by the original RED-AQM system
are automatically adjusted by the suboptimal feedback law. This provides control actions
based on current network conditions replacing the constant configurations as used in the
original RED. The experimental results presented in [3] demonstrate that the proposed
model (and algorithm) can improve the system performance significantly. However, it
was observed that the feedback control law introduces global synchronization into the
system. This is due to the fact that all the connections have the same control policy and
hence have the same mean rate of warning signals prompting similar actions at the same
time leading to synchronization.

In this paper, we propose an improved model aimed at solving the problem of global
synchronization that results from many of the connections reducing their window sizes
at the same time. This we do without compromising any of the advantages of the system
proposed in [3]. In the modified system we propose the following configuration. Access
of each TCP connection to the router is controlled by a dedicated buffer (for the connec-
tion). The single feedback control law used in previous models is replaced by a system
of individual feedback control laws for each of the connections. This leads to decentral-
ized control as opposed to centralized control. For analysis of the active queue manage-
ment system, we still use doubly stochastic Poisson-driven stochastic differential equa-
tions. This models the dynamic behavior of TCP flows and queues of router in which the
drop rate (intensity) for each of the connections is a function of both the average queue
size of the multiplexer and the average queue size of the individual buffer dedicated to the
connection. The objective function which incorporates packet losses, global synchroniza-
tion, link utilization, and congestion is minimized by an iterative choice of the parameters
of the feedback control laws proposed here. Through simulation (experimental) results,
we show that the modified system is not only capable of effectively controlling congestion
and synchronization but also more robust than the earlier versions.

The rest of the paper is organized as follows: in Section 2, system model is presented.
Feedback control laws are given in Section 3. Objective function and state-space formu-
lation are presented in Section 4. Numerical results are presented in Section 5. The paper
ends with conclusion in Section 6.

2. System model

For the construction of mathematical models for the system we need indicator functions
as defined below. Let \( S \) denote any logical or mathematical statement and define the in-
dicator function as follows:

\[
I(S) = \begin{cases} 
1 & \text{if } S \text{ is true,} \\
0 & \text{otherwise.} 
\end{cases}
\] (2.1)

Consider the system presented in Figure 2.1. There are \( n - 1 \) TCP flows, labeled \( i = 1, 2, \ldots, n - 1 \), which attempt to access the router. Each individual source \( i \) is connected
to a dedicated buffer which communicates with the router through a dedicated link of
TCP source 1
TCP source 2
... 
TCP source $n-1$

<table>
<thead>
<tr>
<th>RED</th>
<th>Queue 1</th>
<th>$C_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED</td>
<td>Queue 2</td>
<td>$C_2$</td>
</tr>
<tr>
<td>RED</td>
<td>Queue $n-1$</td>
<td>$C_{n-1}$</td>
</tr>
<tr>
<td>RED</td>
<td>Queue $n$</td>
<td>$C_n$</td>
</tr>
</tbody>
</table>

Figure 2.1. System model.

bandwidth $C_i$. The packets reaching the router from all the individual sources are multiplexed for onward transmission through an outgoing link of capacity $C_n$. In the absence of drop priorities, the relationship between $C_i$ and $C_n$ may be given by

$$C_i(t) = \beta \cdot \frac{C_n(t)}{n-1}, \quad i = 1, 2, \ldots, n-1,$$

where $\beta > 0$ is a design parameter discussed at length later.

For the specific case of differentiated services model (DiffServ) where different connections have different priorities, network designers just provide wider bandwidths between buffers and the multiplexer for those connections having lower drop priority and narrower bandwidths for those with higher drop priority.

The dynamic model of the TCP flow control system can be described in terms of window size of the sources and queue sizes at buffers and the multiplexer. The window size is governed by the following equation:

$$dw_i(t) \equiv \frac{1}{R_i(q_i(t), q_n(t))} dt - I(w_i(t) > 0) \frac{w_i(t)}{2} dN_i^\lambda(t), \quad i = 1, 2, \ldots, n-1,$$

where $w_i(t)$ denotes the window size of TCP connection $i$ at time $t \geq 0$ and the process $N_i^\lambda(t)$ represents the number of packets dropped from the connection $i$ over the time interval $[0, t]$. This is a point process with intensity $\lambda_i$. The process $q_i(t)$ is the queue size at buffer $i$ and $q_n(t)$ is the queue size at the multiplexer measured at time $t \geq 0$. The expression

$$R_i(q_i(t), q_n(t)),$$

which is dependent on both $q_n(t)$ and $q_i(t)$, is the round trip time of TCP connection $i$ and the destination. This is generally given by the following expression:

$$R_i(q_i(t), q_n(t)) \equiv a_i + \frac{q_i(t)}{C_i} + \frac{q_n(t)}{C_n}, \quad i = 1, 2, \ldots, n-1,$$
where $a_i$ denotes the propagation delay between the source and destination and it is considered here as a random variable uniformly distributed over the range $D \subset (0, \infty)$. The set $D$ depends on the network size and its topology and this can be estimated easily while it is impossible to incorporate propagation delay packet by packet and present a reasonably correct analysis of the traffic process. For our numerical experiments, we have chosen \{a_i\} to be uniformly distributed independent random variables with values in $D = [0.01, 0.2]$. The lower and the upper limits of the set $D$ are determined from the topology of the network. The remaining part of the expression models the queuing delay for the source $i$. In summary, the first term on the right-hand side of the expression (2.3) gives the windows opening rate and the second the closing rate.

The dynamics of buffers and multiplexer queue loads can be described as follows:

\[
dq_i(t) = u_i(t)I(q_i(t) < Q_i)dt - v_i(t)dt, \quad i = 1, 2, \ldots, n,
\]

where $Q_i$ is the size of buffer $i$ and $Q_n$ is the size of the multiplexer (main buffer). The processes $u_i(t)$ and $v_i(t)$ expressed as follows represent the instantaneous rates of incoming and outgoing packets at time $t$:

\[
\begin{align*}
  u_i(t) & \equiv \begin{cases} 
    \frac{w_i(t)}{R_i(q_i(t), q_n(t))}, & \text{for } i = 1, 2, \ldots, n-1, \\
    n-1 \\ j=1 \\
    \sum v_j(t), & \text{for } i = n,
  \end{cases} \\
  v_i(t) & \equiv C_i(t)I(q_i(t) > 0) \quad i = 1, 2, \ldots, n.
\end{align*}
\]

**Time delay.** The models presented above are independent of the two-way propagation delay between the router and the sources. In reality there is always a communication delay. Assuming an average delay of $\delta$ units of time, (2.3) and (2.7) can be modified as follows:

\[
\begin{align*}
  dw_i(t) & \equiv \frac{1}{R_i(q_i(t), q_n(t))} dt - I(w_i(t) > 0) \frac{w_i(t)}{2} dN_i^\lambda(t - \delta), \quad i = 1, 2, 3, \\
  u_i(t) & = \frac{w_i(t - \delta)}{R_i(q_i(t), q_n(t))}, \quad i = 1, 2, 3.
\end{align*}
\]

The first equation takes care of delay from router to sources and the second from sources to router.

### 3. Feedback control laws

The process $\{N_i^\lambda(t), t \geq 0\}$ in (2.3) is a counting process with intensity process $\lambda_i(t), t \geq 0$. In the context of internet traffic, open loop control is impractical and so out of question. The intensity process is used as a control variable and its choice should depend on the current status of the traffic which means feedback control. Thus, in general, one may
choose \( \lambda_i \) as an appropriate nonnegative function of the current state \( \xi = \{w_1, w_2, \ldots, w_{n-1}, q_1, q_2, \ldots, q_n\} \) as shown below:

\[
\lambda_i(t) \equiv f_i(w_1(t), w_2(t), \ldots, w_{n-1}(t), q_1(t), q_2(t), \ldots, q_n(t)), \quad i = 1, 2, \ldots, n - 1, \quad t \geq 0.
\]  

(3.1)

In practice it is prohibitively costly to monitor the entire state \( \xi \) of the network. The only information that can be easily monitored by the AQM system in the router is the multiplexer queue \( q_n \) and possibly the buffer queues \( \{q_i\} \). Also it is not practical for the router to monitor each individual window size. Thus the feedback control for each individual source may be required to depend only on the state of its own buffer and the state of the multiplexor. This leads to a partially observed feedback control law. This is given by the following expression:

\[
\lambda_i(t) \equiv f_i(q_i(t), q_n(t)), \quad i = 1, 2, \ldots, n - 1, \quad t \geq 0.
\]  

(3.2)

In high-speed traffic environment, the state of queue changes too fast and therefore direct use of this fast changing traffic would result in high-frequency fluctuation of the control. To avoid this phenomenon, one uses smoothed version of the queue given by the following exponentially weighted moving average [1]:

\[
\overline{q}_i(t) = (1 - \kappa_q) \overline{q}_i(t - \Delta t) + \kappa_eq_i(t), \quad t \geq 0, \quad i = 1, 2, 3, \ldots, n
\]  

(3.3)

leading to the following control laws

\[
\lambda_i(t) \equiv f_i(\overline{q}_i(t), \overline{q}_n(t)), \quad i = 1, 2, \ldots, n - 1, \quad t \geq 0.
\]  

(3.4)

Thus, the short-term increases in the queue sizes, resulting from bursty traffic or from transient congestion, do not result in significant fluctuation of the average queue size. Since both \( \overline{q}_i(t) \) and \( \overline{q}_n(t) \) are random processes, \( N_i(t) \) is a doubly stochastic counting process. The feedback law \( f_i \) can be chosen by the system designer as a suitable nonnegative and nondecreasing function defined on \( \mathbb{R}^{+2} \equiv \{(x, y) \in \mathbb{R}^2 : x, y \geq 0\} \). One possible choice of the feedback law \( f_i \) is given by

\[
f_i(x, y) = g_i(x, y)I(g_i(x, y) > 0), \quad i = 1, 2, \ldots, n - 1,
\]  

(3.5)

where, in general, each \( g_i \) is given by a bivariate polynomial of degree \( m \) with real valued coefficients such as

\[
g_i(x, y) \equiv \sum_{r=0}^{m} \sum_{j=0}^{r} a_{r,j}^i x^j y^{r-j}.
\]  

(3.6)

For numerical experiments, we consider a simpler version of this such as

\[
g_i(x, y) = a_0^i + a_1^i x + a_2^i y + a_3^i x^2 + a_4^i y^2 + \cdots + a_{2m-1}^i x^m + a_{2m}^i y^m, \quad i = 1, 2, \ldots, n - 1.
\]  

(3.7)
6 Mathematical Problems in Engineering

The variable $\lambda_i$ actually denotes the mean rate of congestion warnings sent out to the source $i$. The function $f_i$, defining the packet dropping scheme for the stream $i$, is eventually determined by optimization process. This is done by choosing the coefficients of the above polynomials that minimizes the cost functional introduced in the following section.

4. Objective functional and state space formulation

In the following, we use the notation $E\{z\}$ to denote the expected value of any random variable $z$. For evaluation of network performance over the running period $I \equiv [0, \tau]$, an objective functional is chosen as follows:

$$J(f) \equiv \frac{1}{\tau} E\{-J_T + J_C + J_L + J_S\} = -\bar{J}_T + \bar{J}_C + \bar{J}_L + \bar{J}_S,$$  \hspace{1cm} (4.1)

where

$$f = (f_1, f_2, \ldots, f_{n-1})',$$  \hspace{1cm} (4.2)

$$J_T = \int_I \alpha_1(t) C_n(t) I(q_n(t) > 0) \, dt,$$

$$J_C = \int_I \alpha_2(t) I(q_n(t) \in Q_\alpha) \, dt,$$  \hspace{1cm} (4.3)

$$J_L = \int_I \alpha_3(t) \sum_{j=1}^n (u_j(t) I(q_j(t) Q_j) + \max \{0, u_j(t) - q_j(t)\} I(q_j(t) < Q_j)) \, dt.$$  \hspace{1cm} (4.4)

The last component of the expression (4.1) is given by

$$J_S = \int_I \alpha_4(t) I\left(\sum_{i=1}^{n-1} I(\text{Sign } \varphi_i(t) < 0) > \gamma(n-1)\right) dt,$$  \hspace{1cm} (4.5)

where $\gamma \in (0, 1)$ is also a design parameter discussed later and

$$\varphi_i(t) = \frac{w_i(t + \Delta t) - w_i(t)}{\Delta t}, \quad i = 1, 2, \ldots, n - 1.$$  \hspace{1cm} (4.5)

Here $f$ denotes a feedback control law which has to be chosen to optimize the performance integral given by the expression (4.1). The overall performance of the system is measured in terms of throughput, congestion, packet losses, and synchronization. The quantities $\bar{J}_C, \bar{J}_L, \bar{J}_S$ are, respectively, the expected costs of congestion, packet losses, and synchronization per unit time, while $\bar{J}_T$ is a measure of expected throughput. The quantity $J_T$ gives the utilization of the link bandwidth $C_n$; $J_C$ is a measure of congestion when the average queue size at the multiplexer falls into the congestion zone given by $Q_\alpha \equiv [\alpha Q, Q]$ with $\alpha \in (0, 1)$. The factor $J_L$ is a measure of packet losses at the multiplexer and the buffers due to overflow suffered during the period $[0, \tau]$. The quantity $J_S$ is a measure of global synchronization. The value of $\gamma$ in (4.4) can be chosen by network designers in accordance with the number of connections. The real numbers $\alpha_i, i = 1, 2, 3, 4$ are used as relative weights given to each of the costs. These can be chosen by network designers.
to reflect different concerns and scenarios and assign appropriate weights as necessary. Once all of them are chosen and fixed, one can then attempt to choose the control (or dropping) strategies \( f \) \((4.2)\) to minimize the cost functional \((4.1)\). The optimal strategy guarantees maximum expected throughput and minimum expected congestion, packet losses, and global synchronization.

For further analysis, it is convenient to write the state-space model for the system. Denoting the state of the system by \( \xi \equiv (w_1, w_2, \ldots, w_{n-1}, q_1, q_2, \ldots, q_n)' \), one can write the system in the state-space form as follows:

\[
d\xi = F(\xi)dt + G(\xi)dN^\lambda(t), \tag{4.6}
\]

where the vector field \( F(\xi) \) of dimension \( 2n - 1 \) is given by

\[
F_i(\xi) = \begin{cases} 
1 & 1 \leq i \leq n - 1, \\
R_i(\xi_{i+n-1}, \xi_{2n-1})' & \text{otherwise},
\end{cases}
\tag{4.7}
\]

in which

\[
\triangle_i = \begin{cases} 
\xi_{i+n-1} & n \leq i \leq 2(n - 1), \\
\Sigma_{k=1}^{n-1} \left( C_k I(\xi_{k+n-1} > 0) + \min \left\{ C_k, \frac{\xi_{k+n-1}}{R_k(\xi_{k+n-1}, \xi_i)} \right\} I(\xi_{k+n-1} = 0) \right) & i = 2n - 1.
\end{cases}
\tag{4.8}
\]

The function \( G(\xi) \) is a \((2n - 1) \times (n - 1)\) matrix with entries given by

\[
G_{i,j}(\xi) = \begin{cases} 
I(\xi_i > 0) \left( \frac{\xi_i}{2} \right) \delta_{i,j} & 1 \leq i, j \leq n - 1, \\
0 & n \leq i \leq 2n - 1, 1 \leq j \leq n - 1,
\end{cases}
\tag{4.9}
\]

Since \( \lambda_i \) is determined by the choice of \( f_i \), we may rewrite the system \((4.6)\) as

\[
d\xi = F(\xi)dt + G(\xi)dN^f(t), \tag{4.10}
\]

where the vector function \( f \) is the packet dropping scheme to be determined for optimum performance. Clearly, the cost functional \((4.1)\) can be compactly written as

\[
J(f) = \frac{1}{\tau} E \left\{ \int_0^\tau \ell(t, \xi(t))dt \right\}, \tag{4.11}
\]

where \( \ell \) denotes the integrand of the expression \((4.1)\). One of the objectives of the network provider is to improve the system performance by using control strategies \((4.2)\) that minimize this cost functional.
The principal objective is to determine the optimal feedback law modulated by indicator function as shown in the expression (3.5). The feedback law $f$ of the expression (4.2) based on (2.10) and (3.2) is dependent on a matrix of coefficients of dimension $(n - 1) \times (2m + 1)$ or equivalently a set of vectors

$$a_i \in \mathbb{R}^{(2m+1)}, \quad i = 1, 2, \ldots, n - 1.$$ (4.12)

Rearranging these coefficients in any suitable order, one may consider the coefficient vector $a \in \mathbb{R}^{(n-1)(2m+1)}$ as the choice variable in the optimization process. So the objective is to choose the coefficient vector $a^*$ that minimizes the cost functionals (4.1) or equivalently (4.11). This is accomplished by the optimization process through the iterative choice of $a$. In our simulation, random recursive search (RRS) algorithm is used as the optimization tool.

5. Numerical Results

We present detailed numerical results in order to demonstrate that the modified version of RED mechanism as proposed here is more practical and robust than the earlier version given in [3]. To obtain the optimal feedback control law $f$ (3.7), numerical simulations are performed using MATLAB. For numerical simulation only, we consider a simple scenario in which the system is comprised of three TCP flows without drop priority.

**Basic system parameters used.** The basic system parameters used for numerical simulation are buffer sizes $\{Q_1, Q_2, Q_3, Q_4\}$, weighting factor $\kappa_q$, outgoing link capacity $C_4$, size of the time slot $\Delta t$, and three other factors $\{\gamma, \delta, \beta\}$ representing synchronization as defined in (4.4), average time delay appearing in (2.9), (2.10), and capacity allocation factor as shown in the expression (2.2), respectively (see Table 5.1).

As a matter of fact, $\beta$ balances the traffic between the buffers and the multiplexer. Specifically, if $\beta$ is set too low, sources are assigned less bandwidth thereby limiting their flows and causing packet losses at the user buffers. This may result in reduced utilization of the system significantly. For $\beta \leq 1$, no queue build-up occurs at the multiplexer. In contrast, if $\beta$ is too large, more packet losses and congestion may occur at the multiplexor depending on the volume of incoming traffic. As a compromise, we set $\beta$ equal to 2 in our experiments. Later we vary $\beta$ to determine the robustness of the optimal control law.

5.1. Global optimization.

5.1.1. Linear feedback. For linear state feedback control law, we have chosen the following general form:

$$\lambda_i = f_i(\vec{q}_i, \vec{q}_n) \left( a'_{0i} + a'_{1i} \vec{q}_i + a'_{2i} \vec{q}_n \right) I(a'_{0i} + a'_{1i} \vec{q}_i + a'_{2i} \vec{q}_n > 0), \quad i = 1, 2, 3.$$ (5.1)
The optimization process is started by randomly selecting a coefficient vector \( a \in \mathbb{R}^9 \) which, for convenience, is written in the matrix form

\[
a = \begin{bmatrix}
a_0^1 & a_1^1 & a_2^1 \\
a_0^2 & a_1^2 & a_2^2 \\
a_0^3 & a_1^3 & a_2^3 \\
\end{bmatrix} = \begin{bmatrix}
-1 & -1 & -1 \\
2 & 2 & 2 \\
-3 & -3 & -3 \\
\end{bmatrix},
\]

where the \( i \)th row represents the coefficients of the feedback law of source \( i \). Using the RRS algorithm given in [4], we minimize the cost functional (4.1) by iteratively choosing appropriate \( a \). If a choice decreases the value of the cost functional, it is accepted and the corresponding cost is retained. The process is terminated once the stopping criterion is satisfied. Specific steps of sample choice using RRS are illustrated as follows. First, 44 random samples (points in \( \mathbb{R}^n \)) are taken from the parameter space and the corresponding costs are computed. The point at which the cost functional (4.1) is minimum is taken as the center of the promising region, for example a hypercube, which is further explored. In this step, 11 random samples are chosen from the hypercube. If a better point is found within these 11 samples, the center of the cube is moved to this point keeping the size of the cube unchanged. If a better point is not found in the 11 samples, the size of the hypercube is reduced by half while keeping the center unchanged. This process of contraction and translation is repeated until the size of the region is reduced below a threshold. The search process is restarted until obtaining a point satisfying a certain stopping criterion.

Based on this optimization process, the optimum vector obtained for linear feedback control law is given by

\[
a^* = \begin{bmatrix}
a_0^1* & a_1^1* & a_2^1* \\
a_0^2* & a_1^2* & a_2^2* \\
a_0^3* & a_1^3* & a_2^3* \\
\end{bmatrix} = \begin{bmatrix}
-1.83 & 0.027 & 0.012 \\
-2.72 & 0.039 & 0.019 \\
-2.13 & 0.033 & 0.015 \\
\end{bmatrix},
\]

which is able to keep the value of cost functional (4.1) to a minimum. In order to prevent the RRS from getting trapped in a local optima, we have tried with very large perturbation of the parameter \( a \) from the starting point and also (the local optimum \( a^* \)) with two different starting vectors denoted by \( \tilde{a} \) and \( \tilde{\tilde{a}} \) as given below:

\[
\tilde{a} = \begin{bmatrix}
a_0^1 + 1000 & a_1^1 + 1000 & a_2^1 + 1000 \\
a_0^2 - 1000 & a_1^2 - 1000 & a_2^2 - 1000 \\
a_0^3 + 1000 & a_1^3 + 1000 & a_2^3 + 1000 \\
\end{bmatrix} = \begin{bmatrix}
999 & 999 & 999 \\
-998 & -998 & -998 \\
997 & 997 & 997 \\
\end{bmatrix},
\]

\[
\tilde{\tilde{a}} = \begin{bmatrix}
a_0^1 - 1000 & a_1^1 - 1000 & a_2^1 - 1000 \\
a_0^2 + 1000 & a_1^2 + 1000 & a_2^2 + 1000 \\
a_0^3 - 1000 & a_1^3 - 1000 & a_2^3 - 1000 \\
\end{bmatrix} = \begin{bmatrix}
-1001 & -1001 & -1001 \\
1002 & 1002 & 1002 \\
-1003 & -1003 & -1003 \\
\end{bmatrix}.
\]
Clearly these are very far from the initial choice of $a$. Running the RRS algorithm once again with these vectors as the starting points, we arrive at the following optimal vectors:

$$\bar{a}^* = \begin{bmatrix} -1.80 & 0.025 & 0.013 \\ -2.69 & 0.036 & 0.022 \\ -2.15 & 0.031 & 0.018 \end{bmatrix},$$

$$\tilde{a}^* = \begin{bmatrix} -1.82 & 0.023 & 0.010 \\ -2.72 & 0.034 & 0.024 \\ -2.18 & 0.025 & 0.017 \end{bmatrix},$$

which are very close to $a^*$. This shows that the RRS algorithm is quite efficient, it is able to escape traps of local minima.

5.1.2. Nonlinear feedback. The general form of the nonlinear state feedback control law is taken from the following class of polynomials:

$$\lambda_i = f_i(q_i, q_n)(a_0^i + a_1^i q_i + a_2^i q_n + \cdots + a_{2m-1}^i q_i^m + a_{2m}^i q_n^m) \times I(a_0^i + a_1^i q_i + a_2^i q_n + \cdots + a_{2m-1}^i q_i^m + a_{2m}^i q_n^m > 0).$$

(5.6)

Our objective is to find the best one. To reduce computational time we have concentrated on second-order polynomials, $m = 2$. Again, we use the RRS algorithm to determine the best feedback law from the class of polynomials of degree 2. Randomly, we start with two starting vectors $b \in \mathbb{R}^{3 \times 5} = \mathbb{R}^{15}$ as given below:

$$b = \begin{bmatrix} b_0^1 & b_1^1 & b_2^1 & b_3^1 & b_4^1 \\ b_0^2 & b_1^2 & b_2^2 & b_3^2 & b_4^2 \\ b_0^3 & b_1^3 & b_2^3 & b_3^3 & b_4^3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 \\ -3 & -3 & -3 & -3 & -3 \end{bmatrix},$$

$$\bar{b} = \begin{bmatrix} b_0^5 + 1000 & b_1^5 + 1000 & b_2^5 + 1000 & b_3^5 + 1000 & b_4^5 + 1000 \\ b_0^5 - 1000 & b_1^5 - 1000 & b_2^5 - 1000 & b_3^5 - 1000 & b_4^5 - 1000 \\ b_0^5 + 1000 & b_1^5 + 1000 & b_2^5 + 1000 & b_3^5 + 1000 & b_4^5 + 1000 \end{bmatrix},$$

(5.7)

$$= \begin{bmatrix} 999 & 999 & 999 & 999 & 999 \\ -998 & -998 & -998 & -998 & -998 \\ 997 & 997 & 997 & 997 & 997 \end{bmatrix}.$$
The optimum vectors obtained for the quadratic feedback control law are given by

\[
b^* = \begin{bmatrix}
b_0^* & b_1^* & b_2^* & b_3^* & b_4^* \\
b_0^2 & b_1^2 & b_2^2 & b_3^2 & b_4^2 \\
b_0^3 & b_1^3 & b_2^3 & b_3^3 & b_4^3 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-1.826 & 0.0285 & 0.0110 & 0.00021 & 0.000023 \\
-2.762 & 0.0400 & 0.0144 & 0.000019 & 0.000024 \\
-2.154 & 0.0343 & 0.0123 & 0.00032 & 0.000042 \\
\end{bmatrix},
\]

\[(5.8)\]

\[
\overline{b}^* = \begin{bmatrix}
\overline{b}_0^* & \overline{b}_1^* & \overline{b}_2^* & \overline{b}_3^* & \overline{b}_4^* \\
\overline{b}_0^2 & \overline{b}_1^2 & \overline{b}_2^2 & \overline{b}_3^2 & \overline{b}_4^2 \\
\overline{b}_0^3 & \overline{b}_1^3 & \overline{b}_2^3 & \overline{b}_3^3 & \overline{b}_4^3 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-1.254 & 0.0346 & 0.0546 & 0.00001 & 0.000042 \\
-5.141 & 0.0058 & 0.0173 & 0.000058 & 0.000052 \\
-3.541 & 0.0658 & 0.0211 & 0.00023 & 0.000055 \\
\end{bmatrix}.
\]

It is interesting to note that the last two columns of the above matrices are very small. In other words, the nonlinearity is negligible. The effectiveness of linear control law is close to that of the quadratic law. However, we note a small difference in the two quadratic laws determined by \(b^*\) and \(\overline{b}^*\).

### 5.2. System performance corresponding to the optimal feedback control

It is important for an AQM algorithm to have adaptive and robust control performance for dynamic TCP flows. We examine the robustness of the optimal feedback control laws obtained above with respect to variations in the network parameters and discuss how tunable parameters should be set to optimize the system.

In this set of experiments, the parameter setup is the same as in the above experiments, except the value of \(\beta\). For \(\beta = 3\), the optimization process is started from

\[
a = \begin{bmatrix}
a_0^1 & a_1^1 & a_2^1 \\
a_0^2 & a_1^2 & a_2^2 \\
a_0^3 & a_1^3 & a_2^3 \\
\end{bmatrix} = \begin{bmatrix}
-1 & -1 & -1 \\
2 & 2 & 2 \\
-3 & -3 & -3 \\
\end{bmatrix}.
\]

\[(5.9)\]

The optimum parameter vector \(\hat{a}^*\) was found as

\[
\hat{a}^* = \begin{bmatrix}
\hat{a}_0^1 & \hat{a}_1^1 & \hat{a}_2^1 \\
\hat{a}_0^2 & \hat{a}_1^2 & \hat{a}_2^2 \\
\hat{a}_0^3 & \hat{a}_1^3 & \hat{a}_2^3 \\
\end{bmatrix} = \begin{bmatrix}
-1.78 & 0.023 & 0.011 \\
-2.70 & 0.041 & 0.020 \\
-2.18 & 0.035 & 0.014 \\
\end{bmatrix}.
\]

\[(5.10)\]

In Figure 5.1, we have plotted, for two different values of \(\beta\), the convergence of the optimization algorithm against the number of iterations.
Given $\gamma = 1/3, \delta = 1$, we study the variation of performance for different values of $\beta$. For the experiments, we have

$$C_i(t) = \beta \cdot \frac{C_4(t)}{3}, \quad i = 1, 2, 3.$$  \quad (5.11)

The expected costs for different values of $\beta$, such as $\{1.2, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, are computed by the use of Monte Carlo technique and shown in Table 5.2.
Figure 5.2 shows the comparison of expected costs per unit time of individual factors such as throughput, congestion, packet losses, and synchronization for different values of $\beta$. As shown in Figure 5.2(b), RED (almost) fully utilizes the bandwidth of the outgoing link as $\beta$ approaches 4. This is because the queue at the multiplexor is seldom empty. The incoming TCP (traffic) flows always keep the queue full. This in turn leads to more congestion and more packet losses at the multiplexor as shown in Figures 5.2(c) and 5.2(d). Consequently, AQM system would generate warning signals more frequently to force the sources to cut their window sizes more often, which may result in more global synchronization as shown in Figure 5.2(e). Despite the variation of the individual costs, Figures 5.2(b)–5.2(e), the total system cost, as shown in Figure 5.2(a), and hence the overall performance, is robust with respect to variation of the parameter $\beta$ beyond 2.

Figure 5.4 shows variations of queues at the buffers and the multiplexor for values of $\beta = 1, 1.2, 2, 10$ corresponding to the linear feedback control law. It was observed that if $\beta > 2$, queue build at the buffers is substantially reduced over the simulation period. Note that Figure 5.3(a) also shows that for identical network conditions, the cost corresponding to the linear feedback policy is almost the same as that of the quadratic policy, both of them reaching the minimum at the point $\beta = 2$. The proposed feedback control algorithm shows the robustness in control performance subject to changes of network environments. This ensures robustness in stability and good performance for a wider range of network parameter uncertainties.

5.3. Impact of time delay on performance. In general, the control performance of an AQM system is affected by several network parameters such as the buffer size ($Q_i, i = 1, 2, 3, n - 1$), the link capacity ($C_n$), the traffic balancer (i.e., $\beta$), and the propagation delay of warning signals. The network parameters such as buffer size, the link capacity and the traffic balancer are design parameters and hence fixed once the AQM is installed in a router. In contrast, the propagation delay is a dynamic factor because it changes dynamically over time. In fact, the propagation delay is a function of several factors such as the physical distance between sources, destinations, and the router including traffic conditions. If one wants to be exact, the problem becomes mathematically very difficult and intractable. Therefore, for simplicity one may view $\delta$ as the mean of a random variable having uniform distribution (considering the worst case/maximum entropy) over a compact interval $\Delta \subset [0, \infty)$. For our experiments we have only considered a fixed set of time delays measured in units of basic time slots (used for computing the solutions of the system equations). Given $\gamma = 1/3, \beta = 2$, we use $\delta = \{1\tau, 3\tau, 5\tau\}$. The results shown (see Table 5.3) correspond to the optimum linear and quadratic control laws with $a^*$ and $b^*$ as given in Sections 5.1.1-5.1.2. It is clear from these results that time delay in controls has significant negative impact on system performance. An increase of time delay not only degrades the control performance of an AQM system but may also lead to instability. Thus, the effect of propagation delay of warning signals should be taken into account in designing a robust AQM system.

5.4. Discussion of results. Figure 5.5 plots the variations of window sizes of the sources 1, 2, and 3 over the given period. Examining carefully, one can detect from these plots
Figure 5.2. Individual costs as functions of $\beta$. 

(a) Total cost $J$
(b) Throughput $J_T$
(c) Congestion $J_C$
(d) Packet losses $J_L$
(e) Synchronization $J_S$
Figure 5.3. Variations of queues for different values of $\beta$ (linear case).

<table>
<thead>
<tr>
<th>Case</th>
<th>$J$</th>
<th>$\bar{J}_T$</th>
<th>$\bar{J}_C$</th>
<th>$\bar{J}_L$</th>
<th>$\bar{J}_S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear with 1 interval delay</td>
<td>$-195.95$</td>
<td>$199.50$</td>
<td>$0.38$</td>
<td>$3.16$</td>
<td>$0.010$</td>
</tr>
<tr>
<td>Quadratic with 1 interval delay</td>
<td>$-195.94$</td>
<td>$199.42$</td>
<td>$0.37$</td>
<td>$3.10$</td>
<td>$0.011$</td>
</tr>
<tr>
<td>Linear with 3 interval delay</td>
<td>$-190.58$</td>
<td>$197.91$</td>
<td>$0.29$</td>
<td>$7.03$</td>
<td>$0.011$</td>
</tr>
<tr>
<td>Quadratic with 3 interval delay</td>
<td>$-190.10$</td>
<td>$197.38$</td>
<td>$0.29$</td>
<td>$6.98$</td>
<td>$0.012$</td>
</tr>
<tr>
<td>Linear with 5 interval delay</td>
<td>$-186.81$</td>
<td>$195.16$</td>
<td>$0.40$</td>
<td>$7.93$</td>
<td>$0.012$</td>
</tr>
<tr>
<td>Quadratic with 5 interval delay</td>
<td>$-186.74$</td>
<td>$194.98$</td>
<td>$0.40$</td>
<td>$7.83$</td>
<td>$0.013$</td>
</tr>
</tbody>
</table>

significant reduction of synchronization. This corresponds to optimal feedback control laws $f_1, f_2, f_3$ with coefficients $a = a^*$ (linear) and $b = b^*$ (quadratic). This is expected as the intensities $\lambda_1, \lambda_2, \lambda_3$ (which are functions of average queue size at the multiplexer and
buffers 1, 2, and 3, resp.) differ with one another due to the difference of the coefficients of the optimum vectors \((a^* \text{ or } b^*)\). This ensures that each of the connections has different dropping strategies over the same period thereby avoiding global synchronization (i.e., reducing window sizes at the same time).

It is interesting to note that the vector \(b^*\), with elements \(b_{0}^*, b_{1}^*, b_{2}^*, i = 1,2,3\), is very close to the vector \(a^*\) with elements \(a_{0}^*, a_{1}^*, a_{2}^*\). In other words, the coefficients of the second-order terms (in the quadratic control law) such as \(b_{3}^*, b_{4}^*, i = 1,2,3\), are very close to zero. In fact, the simulation results obtained by use of optimum nonlinear feedback control laws are almost the same as those obtained by use of optimum linear
feedback control as seen in Figure 5.3. It is important to mention that computation of optimum nonlinear feedback control laws take much longer time than linear ones and does not provide any noticeable improvement. Therefore, we conclude that in our particular setting it is not necessary to consider nonlinear feedback control laws. The linear optimal feedback control laws are fairly satisfactory. Consequently, (3.7) can be replaced by the simpler linear law

\[ g_i(x, y) = a_0^i + a_1^i x + a_2^i y, \quad i = 1, 2, \ldots, n - 1. \] (5.12)

6. Conclusion

A variant of RED model presented in this paper formally defines the interactions between TCP connections and the AQM system in the computer network. The dynamic model which contains a feedback control law is governed by a system of stochastic differential equations driven by doubly stochastic point processes with intensities being the controls for each of the TCP connections. The frequency of warning signals represented by the intensities is taken as function of available information such as the router and buffer loads. The optimal (linear) feedback control law based on such information is obtained by random recursive search (RRS) technique. According to the feedback control law proposed, the controller observes the behavior of all the queues (at buffers and the multiplexer) and controls every individual intensity by sending congestion signals (warnings) to the sources for adjustment of their transmission rates. The simulation results demonstrate that the model and the methodology proposed can improve the system performance significantly via maximizing the link utilization and minimizing congestion, packet losses, and global synchronization.

References


N. U. Ahmed: School of Information Technology and Engineering (SITE), University of Ottawa, 161 Louis Pasteur, P.O. Box 450, Ottawa, ON, Canada K1N 6N5
Email address: ahmed@site.uottawa.ca

X. H. Ouyang: School of Information Technology and Engineering (SITE), University of Ottawa, 161 Louis Pasteur, P.O. Box 450, Ottawa, ON, Canada K1N 6N5
Email address: xouya073@uottawa.ca