SUBOPTIMAL FEEDBACK CONTROL OF TCP FLOWS IN COMPUTER NETWORK USING RANDOM EARLY DISCARD (RED) MECHANISM

N. U. AHMED AND CHENG LI

Received 8 February 2005

We consider a dynamic model that simulates the interaction of TCP sources with active queue management system (AQM). We propose a modified version of an earlier dynamic model called RED. This is governed by a system of stochastic differential equations driven by a doubly stochastic point process with intensity as the control. The feedback control law proposed observes the router (queue) status and controls the intensity by sending congestion signals (warnings) to the sources for adjustment of their transmission rates. The (feedback) control laws used are of polynomial type (including linear) with adjustable coefficients. They are optimized by use of genetic algorithm (GA) and random recursive search (RRS) technique. The numerical results demonstrate that the proposed model and the method can improve the system performance significantly.

1. Introduction

Congestion is one of the most significant problems in Internet traffic engineering. To avoid congestion collapse, the Internet Engineering Task Force (IETF) recommended deployment of active queue management (AQM) in routers [2] in 1998. Random early detection (RED), a widely used congestion avoidance mechanism originally proposed in 1993 [7], has proved to be capable of detecting the inception of congestion and avoiding global synchronization. Well-tuned routers can improve queuing delay and provide high throughput.

As illustrated in left side of Figure 1.1, the original RED algorithm has four parameters: $\min_{th}$, $\max_{th}$, $\max_p$, and α [7]. $\min_{th}$ and $\max_{th}$ are defined as two-packet dropping thresholds. The average queue size q_a is calculated by applying a lowpass filter (exponentially weighted moving average). No packets are dropped if q_a is less than $\min_{th}$. For q_a between $\min_{th}$ and $\max_{th}$, the packet dropping probability is increased linearly with q_a up to $\max_p$. When q_a exceeds $\max_{th}$, all arriving packets are dropped. The first three parameters determine the dropping decision: when and how aggressively to drop. The last parameter α is the weight given to the current value for average queue size estimation.

Mathematical Problems in Engineering 2005:5 (2005) 477–489
DOI: 10.1155/MPE.2005.477

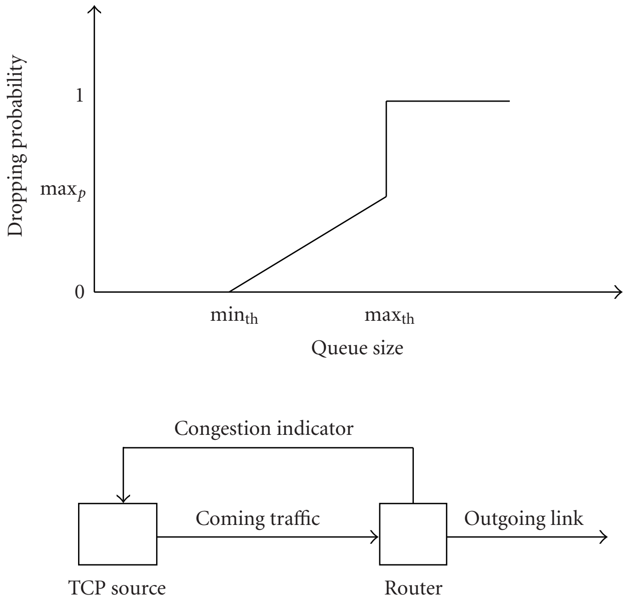


Figure 1.1. RED mechanism.

It controls the cutoff frequency of the lowpass filter, which provides a tradeoff for detecting manageable traffic variations and filtering out transient traffic oscillations [11]. By invoking the TCP congestion avoidance mechanisms, TCP traffic sources respond to packet drops by reducing window sizes as dictated by the router congestion as shown in right side of Figure 1.1. However, RED's parameters are sensitive to traffic load; network operators are required to constantly tune those parameters based on current network conditions. Multiplicity of parameters (here 4) makes adjustment of configuration complicated.

Recently, several algorithms have been proposed to reduce the number of RED's parameters, and adjust the parameters in varying conditions [4, 5, 6]. Considering large-delay variation, and throughput sensitivity to traffic load, and parameter setting, Floyd et al. [6] suggested automatic adjustment of max$_p$ value in order to keep the average queue size q_a around $(\min_{th}+\max_{th})/2$. The parameter min$_{th}$ was set as a function of the link bandwidth and max$_{th}$ was chosen as 3 times of min$_{th}$. Determination of the values of min$_{th}$ and max$_{th}$ is an open question. Cnodder et al. [4] proposed a rate-based RED algorithm which reduced the number of parameters to one. It considers the packet dropping rate as a function of the long-term average arrival rate and the average queue size. However, it has two weaknesses. (1) It may drop packets when the queue is almost empty, which is not necessary. (2) The estimated traffic which is used for calculation of drop probability is based on an exponential function. Recent studies show that network traffic has long-range dependence property [3, 8, 10], and therefore the estimates based on short-range assumptions are not accurate. To analyze an active queue management system, Poisson-process-driven stochastic differential equations (SDEs) are used to model the behaviors of TCP flows and queues of routers [9]. The methodology and model are helpful for analyzing the dynamics of AQM. However, the approximations, based on assumptions such as $E\{1/R(q)\} = E\{(1/R(E(q)))\}$, (where R is the round trip time, q is current queue size) are seriously questionable. In this paper, we propose a revised model based on the earlier model. We replace the fixed propagation time a_i by a nonnegative random variable uniformly distributed over the range of values it may possibly take. For the flow dynamics (window size), we use doubly stochastic Poisson-driven stochastic differential equation where the rate (intensity) λ is an appropriate function of the multiplexor (router load) q size. We introduce an objective functional which includes throughput and congestion and then minimize this functional by an appropriate choice of the rate functional as mentioned above. By numerical simulation, we demonstrate how effective our proposed technique is. Numerical computations are based on genetic algorithm (GA) and recursive random search (RRS) algorithms [11].

The rest of the paper is organized as follows. In Section 2, system model is presented. Objective function and state-space formulation are given in Section 3. We introduce system model with delays in Section 4. Numerical results are presented in Section 5. The paper ends with conclusion in Section 6.

2. System model

To present the system model, we frequently make use of indicator functions. Let S denote any logical or mathematical statement and define the indicator function of S as follows:

$$I(S) \equiv \begin{cases} 1 & \text{if } S \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

The basic philosophy behind the TCP congestion control algorithm constitutes additive increase and multiplicative decrease (AIMD) of window size. For each round-trip time, the system increases the window size by one packet while, if a packet is lost, it halves the window size. For simplicity, here we ignore the slow start and retransmit timer mechanisms. Let $n-1$ denote the number of TCP flows connected to the router. The dynamics of the TCP congestion control system can then be described in terms of window size of the sources (users) and q size of the router. Note that the larger the window size is the larger TCP flow rate is. The window size is governed by the following equation:

$$dw_i(t) \equiv \frac{1}{R_i(q(t))}dt - I(w_i(t) > 0)\frac{w_i(t)}{2}dN_i^{\lambda}(t), \quad i = 1,2,\ldots,n-1, \tag{2.2}$$

where $R_i(q(t))$ is the round-trip time dependent on the router q size, $w_i(t)$ is the window size, and the process $N_i^{\lambda}(t)$ represents the number of packets dropped from the stream i over the time interval $[0,t]$. The round-trip time is generally given by the following expression:

$$R_i(q(t)) \equiv a_i + \frac{q(t)}{C}, \tag{2.3}$$

where a_i denotes the round-trip propagation time between the source i and the router in case of no congestion. Since the round-trip routes may vary, this is considered to be a random variable with certain distribution discussed later. The second term in this expression represents additional time required for a complete round trip if the router is not free. In summary, the first term on the right-hand side of (2.2) gives the window's opening rate and the second the closing rate.

The router queue dynamics can be described as follows:

$$dq \equiv -CI(q(t) > 0)dt + \left(\sum_{i=1}^{n-1} \frac{w_i(t)}{R_i(q(t))}\right)dt, \tag{2.4}$$

where $q(t)$ is the current queue size, C is the channel (or link) capacity. The first term on the right-hand side of (2.4) represents the service rate and the second term measures the traffic arrival rate at the router (from all the sources). The model presented above is a modified version of a similar model proposed in [9].

The process N_i^λ for each $i = 1, 2, \ldots, n-1$ is a (nondecreasing) counting process with intensity process λ which is a nonnegative function of the current q size at the router. Since q is a random process, N_i^λ is a doubly stochastic counting process. The variable λ can be chosen by the system designer as follows:

$$\lambda(t) \equiv f(q(t)), \quad t \geq 0, \tag{2.5}$$

where f is a suitable nonnegative, nondecreasing function with the domain, range, and the boundary conditions as follow:

$$f : [0, Q] \longrightarrow [0, \infty], \quad f(0) = 0,\ f(Q) = \gamma, \tag{2.6}$$

with γ being equal to the reciprocal of the smallest propagation time. One possible choice of the feedback policy f is given by

$$f(x) = g(x)I(g(x) > 0), \tag{2.7}$$

where

$$g(x) = a_n x^n + \cdots + a_i x^i + \cdots + a_2 x^2 + a_1 x + a_0 \tag{2.8}$$

is a polynomial of degree n with a_i, $i = 1, \ldots, n$, being real-valued coefficients.

Physical interpretation of λ is simple. It denotes the mean rate of congestion warnings sent out to the users (sources). The parameter Q is the size of the (router) multiplexor (defining it's storage limit). The function f defines the packet dropping scheme (programme). This is the function that can be chosen by the network designer. Once this is chosen, (2.5) determines the intensity, and hence the dropping rate. When the current queue size is less than q_p, λ=0, no packet is dropped. Our feedback model, once optimized, will automatically determine q_p from the first (positive) root of g on the interval $[0,Q]$. For numerical simulation using our model, we assume that $N_i^\lambda(t)$, $t \geq 0$, is a conditional Poisson process, conditioned on λ (or doubly stochastic Poisson process). In general, λ should be a function of the current queue size $q(t)$ as indicated above. But since the frequency of variation of q can be very large forcing frequent changes of the window size leading to oscillation (and instability), it is desirable to use a filtered version of this. This can be partially achieved by using the moving average of q denoted by q_a which is determined by the following equation:

$$q_a(t) = (1-\kappa)q_a(t-\Delta t) + \kappa q(t), \quad t \geq 0, \tag{2.9}$$

for some $\kappa \in [0,1]$. The parameter κ is chosen sufficiently small to avoid unnecessary frequent changes in congestion warnings. This frequency cannot exceed, in fact should be sufficiently smaller than, the frequency of acknowledgement (ACK) flows.

3. Objective function and state-space formulation

For evaluation of network performance over the running period $I \equiv [0,T]$, the objective functional can be chosen as

$$J(f) \equiv E\left\{-\left(\frac{1}{T}\right)\left(\int_I \lambda_1(t)CI(q(t)>0)\,dt\right) + \left(\frac{1}{T}\right)\left(\int_I \lambda_2(t)I(q(t)\in Q_\alpha)\,dt\right)\right\}, \tag{3.1}$$

where the set $Q_\alpha \equiv [\alpha Q, Q]$ and $E\{z\}$ denotes the expected value of the random variable z. The first term within the expectation gives the throughput of the router with a negative sign. The second term is a measure of loss due to congestion at the router. The set Q_α with $\alpha \in (0,1]$ defines the congestion zone. The functions $\lambda_i(t)$, $i = 1,2$, are nonnegative measurable functions used as relative weights (or importance) given to each of the costs. These can be chosen by network designers to reflect different concerns and scenarios as necessary. Once these are chosen and fixed, one can then attempt to choose the dropping strategy f to minimize the cost functional J. The optimal strategy guarantees maximum expected (average) throughput and minimum expected losses due to jitter and congestion.

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

$$d\xi = F(\xi)dt + G(\xi)dN^\lambda(t), \tag{3.2}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is given by

$$\begin{aligned} F_i(\xi) &\equiv \left(\frac{1}{R_i(\xi_n)}\right), \quad 1 \leq i \leq n-1, \\ F_n(\xi) &\equiv -CI(\xi_n > 0) + \sum_{i=1}^{n-1}\left(\frac{\xi_i}{R_i(\xi_n)}\right). \end{aligned} \tag{3.3}$$

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

$$\begin{gathered} G_{i,j}(\xi) \equiv I(\xi_i > 0)\left(\frac{\xi_i}{2}\right)\delta_{i,j}, \quad i,j = 1,2,\dots,n-1, \\ G_{n,j}(\xi) = 0, \quad j = 1,2,\dots,n-1, \\ N^{\lambda} \equiv \left(N_1^{\lambda}, N_2^{\lambda}, \dots, N_{n-1}^{\lambda}\right)'. \end{gathered} \tag{3.4}$$

Since λ is determined by the choice of f, we may rewrite the system (3.2) as

$$d\xi = F(\xi)dt + G(\xi)dN^{f}(t), \tag{3.5}$$

where f is the packet dropping scheme to be determined for optimum performance. Clearly, the cost functional (3.2) can be compactly written as

$$J(f) = E\left\{\left(\frac{1}{T}\right)\int_0^T \ell(t,\xi(t))\,dt\right\}, \tag{3.6}$$

where ℓ denotes the integrand of the expression (3.1). One of the objectives of the network provider is to improve the system performance by using strategies that minimize this cost functional.

Our principal objective is to determine the best feedback policy $f(\cdot)$, from the class of polynomial functions modulated by indicator functions as shown in the expression (2.7), that minimizes the cost functional (3.1) or equivalently (3.6). Our approach does not need a priori specification of the threshold q_p at which packet drops are initiated. We let the optimization process determine this through the choice of the coefficients of the polynomial g which determine the roots. Due to the presence of indicator functions, the cost function versus the coefficients of the polynomial g is not smooth. Most traditional optimization methods require computation of gradients. In a nonsmooth situation like we have, those methods are difficult to implement and tend to get trapped in local minima. Random recursive search (RRS) algorithm appears to be a good choice to solve this kind of problems. In order to verify the simulation results, we use two different optimization techniques: GA and RRS. The numerical results based on these two techniques are very close. However, the computation time with RRS is approximately 50 percent of that of GA.

4. System model with delays

The models presented above do not take into account the time delay due to information exchange between the router and the users. Assuming an average delay of δ units of time, the models can be rewritten as follows:

$$dw_i(t) \equiv \frac{1}{R_i(q(t))}dt - I(w_i(t) > 0)\frac{w_i(t)}{2}dN_i^{\lambda}(t-\delta), \quad i = 1,2,\dots,n-1, \tag{4.1}$$

$$dq \equiv -CI(q(t) > 0)dt + \left(\sum_{i=1}^{n-1}\frac{w_i(t-\delta)}{R_i(q(t))}\right)dt. \tag{4.2}$$

Table 5.1. Cost and loss at queue.

Order of polynomial	Linear	Quadratic	Cube
Cost J	-14301	-13085	-11263
Loss at queue	2119.8	2139.1	2179.4

In fact, the communication delay is a function of several factors such as time, the physical distance between sources and router, and traffic condition (router queue). If one wants to be exact, the problem becomes mathematically intractable. Therefore, for simplicity one may view δ as the mean of a random variable having uniform distribution (considering the worst case/maximum entropy) with a compact support $D \not\ni 0$.

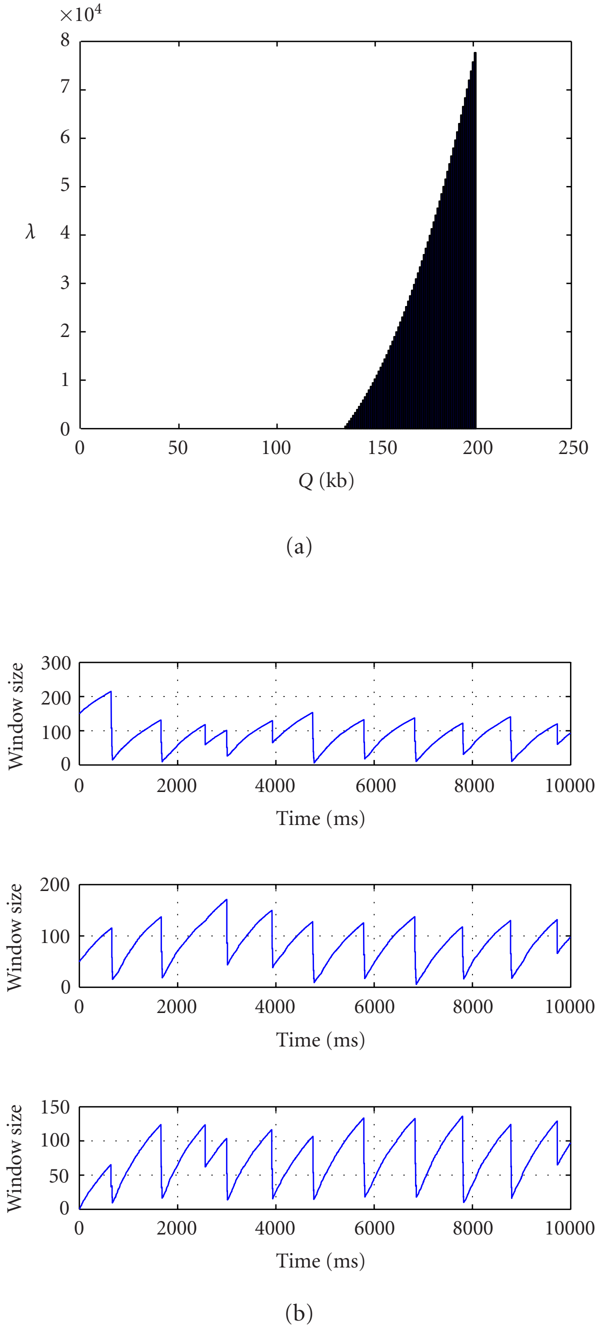
5. Numerical results

For numerical simulation, we consider a system comprised of three TCP traffic sources connected to one router. The observation (measurement) time interval at router side is taken as 10 milliseconds, the (moving average) factor κ is set as 0.02, the maximum queue size $Q = 200\,\text{Kb}$, and the set D is taken as $D \equiv [10\,\text{ms}, 200\,\text{ms}]$. This range is based on recent measurement studies [1].

We use RRS to optimize the cost functional (3.1) with feedback policy given by (2.7). For illustration, we choose only polynomials of degree 3, 2, and 1 (linear) in that order. Figures 5.1, 5.2, 5.3 show the values of λ as functions of average queue size based on the optimized feedback law f associated with the polynomials as mentioned above. More precisely, we choose any given polynomial and start with an arbitrary set of coefficients and substitute the corresponding f, and hence $\lambda \equiv f(\cdot)$ given by (2.5), in the system dynamics (2.2) and (2.4). Then we minimize the cost functional (3.1) using RRS algorithm by iteratively choosing the coefficients. In all of the three cases, the system does not start dropping packets until the queue size has increased to about 125 Kb, approximately 62 percent of the router size. Dropping packets too early can reduce throughput even though there is no noticeable trend towards congestion. Comparing Figures 5.1, 5.2, 5.3 controls based on polynomials of degree 2 and 3 lead to global traffic synchronization which is not desirable. The reason is that, for degrees 2 and 3, with the increase of the queue size, the λ values increase much faster than the linear case. This generates (warning) signals too frequently, forcing the sources to cut their window sizes leading to reduced throughput and global synchronization. In fact, when the queue size starts to build up, reducing the traffic rates of one or two users will relieve the congestion while maintaining high level of throughput.

To compare the results corresponding to the linear case with those based on the polynomials of degree 2 and 3, we normalize the maximum values $f(Q)$ (associated with quadratic and cubic polynomials) to the same level γ as the linear case. Table 5.1 shows the cost and losses at the router (queue) for different cases.

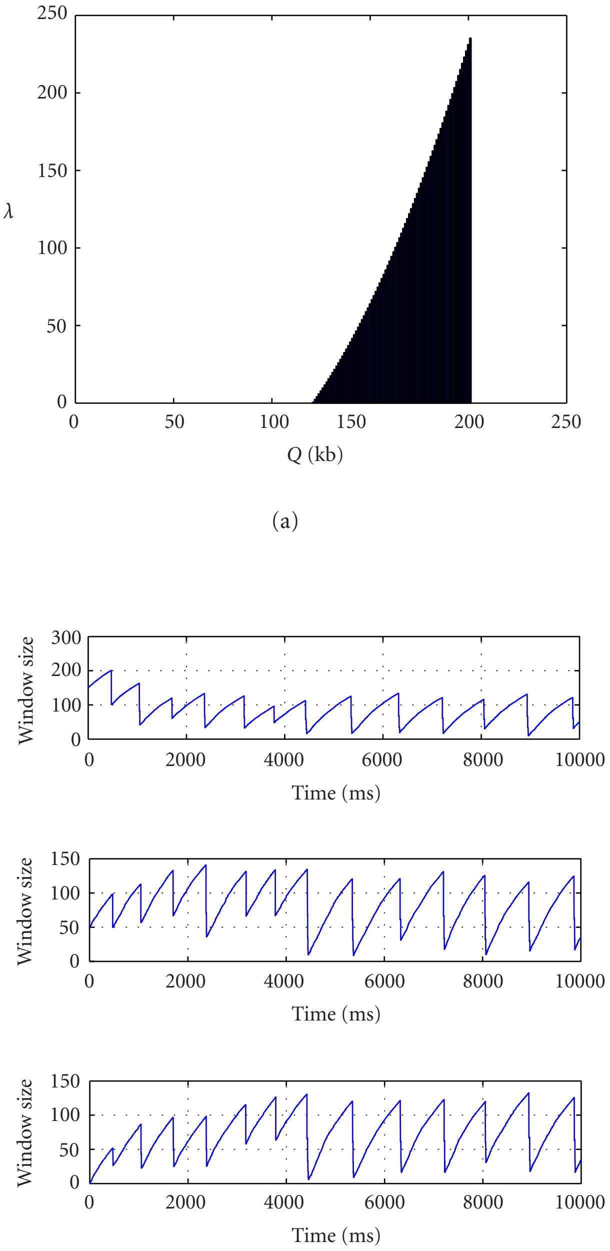
The graphs of Figures 5.4, 5.5, 5.6 show the variation of queue for the three cases. Linear case has the smallest variation. It drops packets more aggressively in the early stages of discard, and thereby it prevents congestion more efficiently. In contrast, the polynomial



(a)

(b)

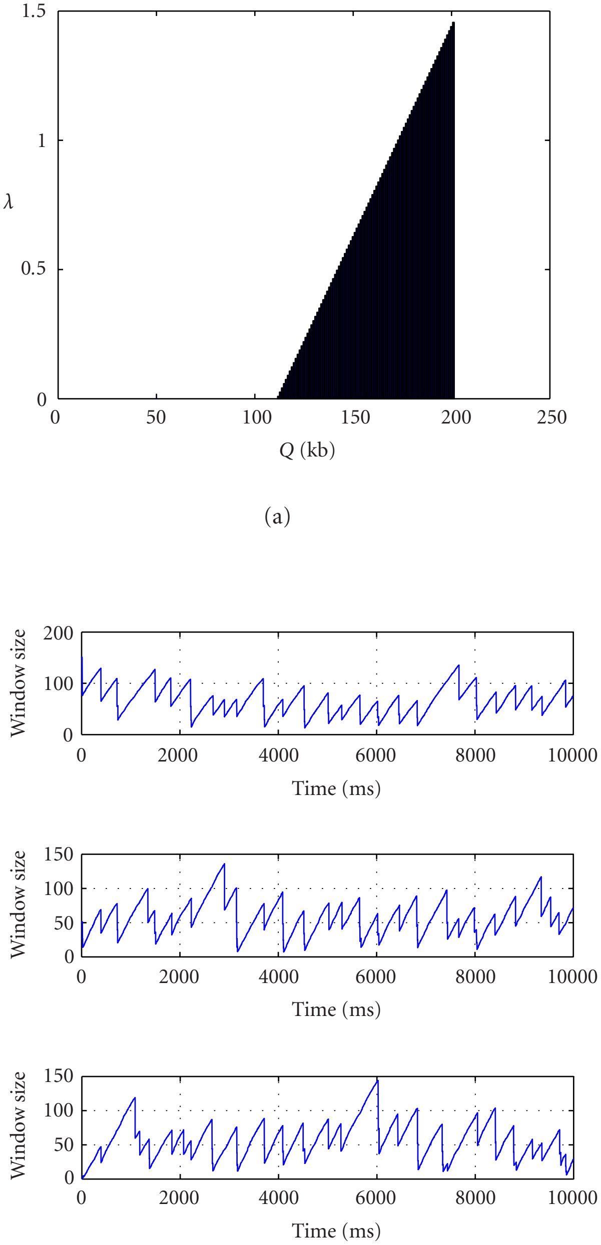
Figure 5.1. Cubic control law.



(a)

(b)

Figure 5.2. Quadratic control law.



(a)

(b)

Figure 5.3. Linear control law.

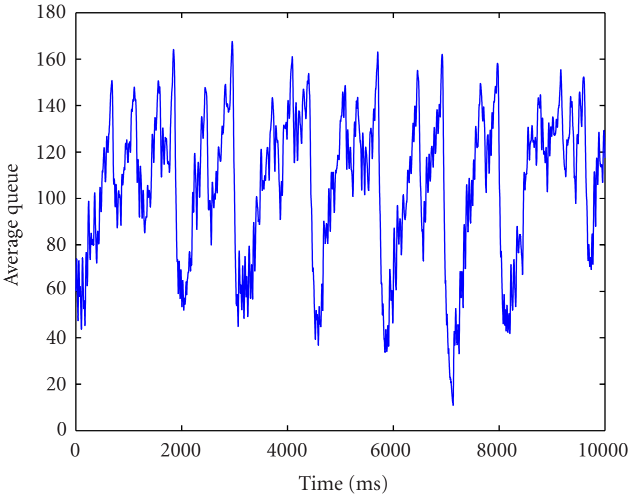


Figure 5.4. Variation of queue for cubic control law.

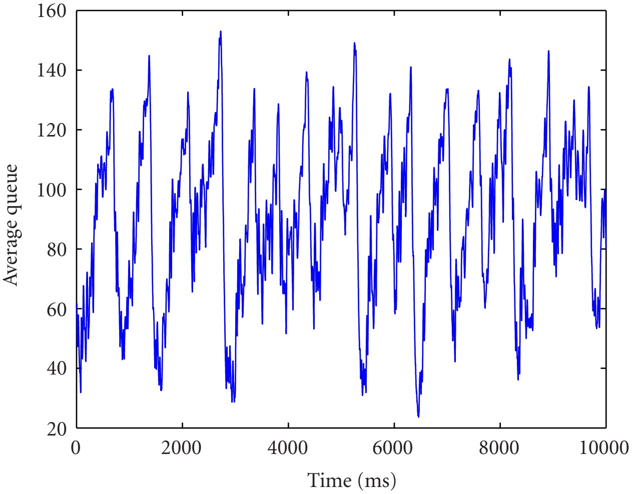


Figure 5.5. Variation of queue for quadratic control law.

controls (degree 2 and higher) drop less aggressively at the early stages, which leads to queue build up, and hence more chances for congestion.

6. Conclusion

In this paper, we have developed a dynamic model that simulates the interaction of TCP sources with active queue management system (AQM). We have proposed a modified version of an earlier model called RED. This is governed by a system of stochastic differential

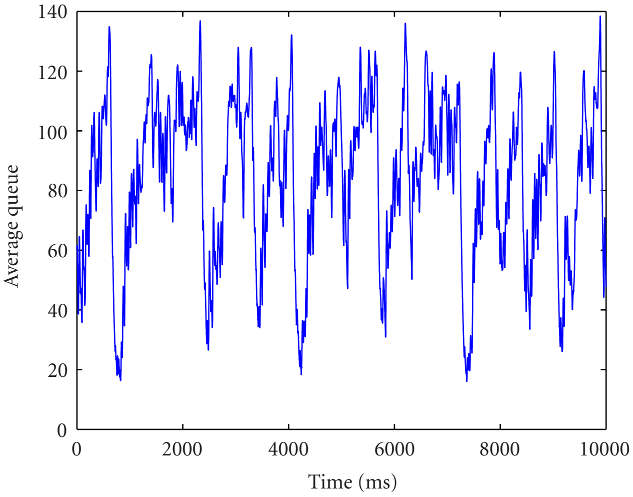


Figure 5.6. Variation of queue for linear control law.

equations driven by doubly stochastic point process with intensity as the control. This proposed model contains a feedback control law that observes the router (queue) status and controls the intensity by sending congestion signals (warnings) to the sources for adjustment of their transmission rates according to the router load. The control laws are optimized by use of genetic algorithm (GA) and random recursive search (RRS) technique. It is observed from our simulation results that the linear control law is better than the polynomial control laws. More precisely, the higher the degree of the polynomial control law is the lower is the performance. The numerical results demonstrate that the proposed model and the method can improve the system performance significantly. As mentioned in the introduction, in previous studies, the choice of the critical parameters such as $\{\min_{th}, \max_{th}, P_{\max}\}$ describing important thresholds for making discarding decision was considered to be an unsolved problem. According to our approach, these parameters or their equivalents are automatically determined by the optimization process itself. Thus this paper provides a technique for determination of critical parameters.

Acknowledgment

This work was partially supported by the National Science and Engineering Research Council of Canada under Grant no. A7109.

References

[1] J. Aikat, J. Kaur, F. D. Smith, and K. Jeffay, *Variability in TCP roundtrip times*, Internet Measurement Conference, Florida, 2003.

[2] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, *Recommendations on queue management and congestion avoidance in the Internet*, RFC 2309.

[3] M. Crovella and A. Bestavros, *Self-similarity in World Wide Web traffic: evidence and possible causes*, IEEE/ACM Trans. Networking **5** (1996), no. 6, 835–846.

[4] S. De Cnodder, K. Pauwels, and O. Elloumi, *A rate based RED mechanism*, The 10th International Workshop on Network and Operating System Support for Digital Audio and Video, Chapel Hill, North Carolina, 2000.

[5] W. Feng, D. Kandlur, D. Saha, and K. Shin, *A self-configuring RED gateway*, Proceedings of INFOCOM '99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, New York, 1999, pp. 1320–1328.

[6] S. Floyd, R. Gummadi, and S. Shenker, *Adaptive RED: an algorithm for increasing the robustness of RED's active queue management*, preprint, 2001.

[7] S. Floyd and V. Jacobson, *Random early detection gateways for congestion avoidance*, IEEE/ACM Trans. Networking **1** (1993), no. 4, 397–413.

[8] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, *On the self-similar nature of ethernet traffic (extended version)*, IEEE/ACM Trans. Networking **2** (1994), no. 1, 1–15.

[9] V. Misra, W. B. Gong, and D. Towsley, *Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED*, Proceedings of ACM SIGCOMM Computer Communication Review **30** (2000), no. 4, 151–160.

[10] V. Paxson and S. Floyd, *Wide-area traffic: the failure of Poisson modeling*, IEEE/ACM Trans. Networking **3** (1994), no. 3, 226–244.

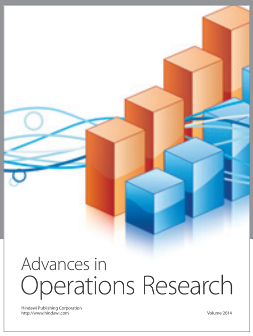
[11] T. Ye and S. Kalyanaraman, *Adaptive tuning of RED using on-line simulation*, GLOBECOM 2002—IEEE Global Telecommunications Conference, vol. 3, Taipei, 2002, pp. 2210–2214.

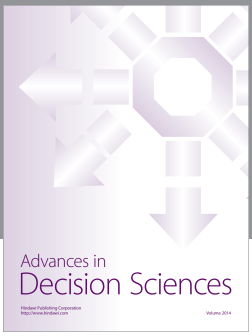
N. U. Ahmed: School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5

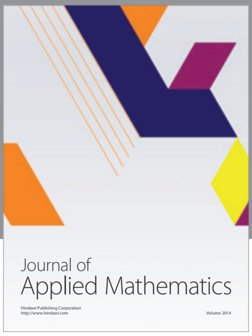
E-mail address: ahmed@site.uottawa.ca

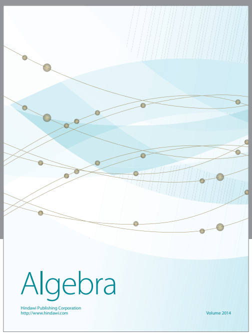
Cheng Li: School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5

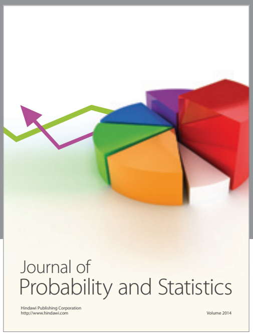
E-mail address: lichengca@yahoo.com


Advances in Operations Research

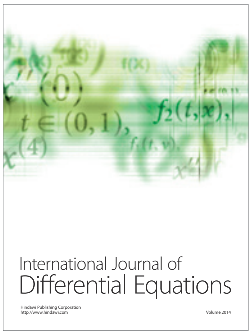

Advances in Decision Sciences

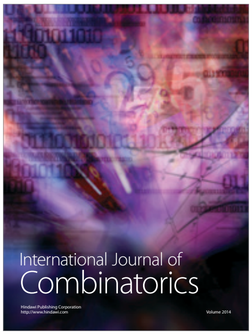

Journal of Applied Mathematics

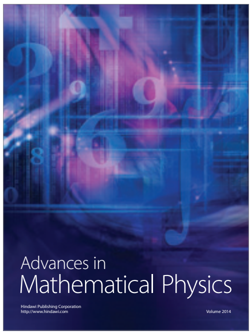

Algebra

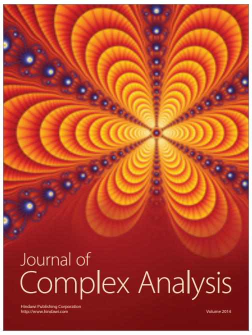

Journal of Probability and Statistics

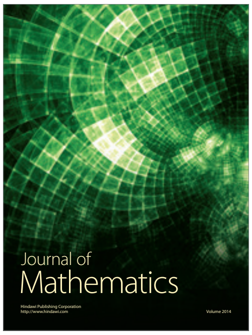
The Scientific World Journal

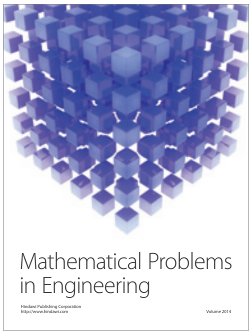

International Journal of Differential Equations

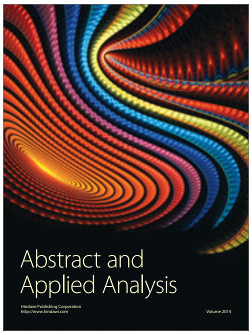

International Journal of Combinatorics

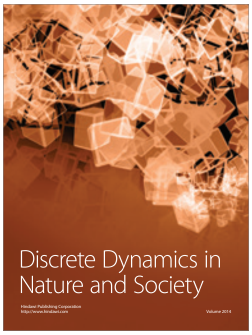

Advances in Mathematical Physics

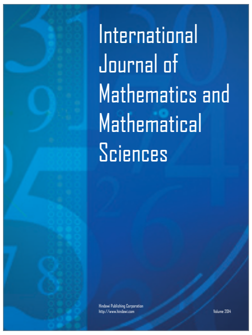

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

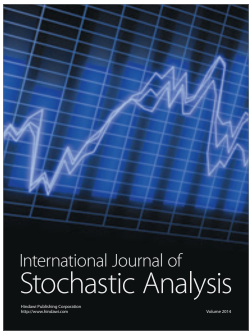

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization