

Bacterial Foraging Optimization Robust-RED for AQM/TCP Network

Amir Esmaeili Abharian, Ehsan Shakeri

Abstract—The congestion control problem of the intermediate nodes in the Internet has received extensively attention in control community. In this paper, a novel intelligent robust early detection controller based on a bacterial foraging optimization algorithm is presented to solve the problem of robust active queue management. However in This paper, design approach of robust BFO-RED active queue management with rejection disturbance in TCP network is proposed using bacterial foraging optimization based robust algorithm. To design disturbance traffic rejection tuning, disturbance rejection conditions based on H_2/H_∞ are illustrated and the performance of response based on the bacterial foraging is computed for the designed RED as Integral of time weighted squared error (ITSE). Hence, parameters of classical RED are selected by bacterial foraging based robust H_2/H_∞ algorithm to obtain the required response. We illustrate our results with NS2 simulations.

Index Terms—Active queue management (AQM), random early detection (RED), Bacterial foraging optimization (BFO).

I. INTRODUCTION

The transmission control protocol (TCP) congestion control mechanism has adopted the end-to-end window-based flow control to prevent congestion, which has been a basic operation of the current Internet. However, the Internet expects the network itself to cooperate with end hosts in congestion control to enhance TCP throughput and to fairly share network resources. Routers need large buffers to absorb network traffic bursts. Consequently, with passive drop-tail queue management, the traffic is subjected to high values of delay and delay jitter.

In the last decade, evolutionary computation based approaches have received increased attention from the engineers dealing with problems which could not be solved using conventional problem solving techniques [1-3]. A typical task of a GA in this context is to find the best values of a predefined set of free parameters associated with either a process model or a control vector. One of the active areas of research in GA is system identification [4]-[6]. A recent survey of evolutionary algorithms for the evaluation of improved learning algorithm and control system engineering can be found in [4, 6, 5]. The general problem of evolutionary algorithm based engineering system design has been tackled

in various ways. GA has also been used to optimize nonlinear system strategies. Among them, a large amount of research is focused on the design of fuzzy controllers using evolutionary algorithm approaches. GA could be used for developing the knowledge base about the controlled process in the form of linguistic rules and the fine tuning of fuzzy membership function [5].

Bacterial Foraging Optimization (BFO) is a population-based numerical optimization algorithm. Until date, BFO has been applied successfully to some engineering problems, such as optimal control [7], harmonic estimation [8], transmission loss reduction [9] and machine learning [10]. However, experimentation with complex optimization problems reveals that the original BFO algorithm possesses a poor convergence behavior compared to other nature-inspired algorithms and its performance also heavily decreases with the growth of the search space dimensionality.

It should be noted that even the most successful nature-inspired optimization techniques, such as Genetic Algorithm (GA) [9, 11], and Particle Swarm Optimization (PSO) [10, 11], are also sensitive to the increase of the problem complexity and dimensionality, due to their stochastic nature [12]. Mixed H_2/H_∞ control design approaches are useful for robust performance for systems under parameter perturbation and uncertain disturbance. However, the conventional output feedback designs of mixed H_2/H_∞ optimal control is very complicated and not easily implemented for practical industrial applications [13].

The first well-known AQM scheme, random early detection (RED) [14] algorithm, was developed and introduced into Internet routers for reducing the flow synchronization problem and calmed the traffic load via measurement of average queue length. Nevertheless, several studies and theoretical analyses have shown that the performance of RED is sensitive to its parameter settings and traffic load due to it being designed in an ad hoc architecture [15]. Therefore, a number of new modified schemes including ARED [16], FRED [17], and SRED [18], have been proposed in the literature. However, those studies are unable to maintain the system performance in a wide range of operational conditions such as the number of connections, propagation delay, and link capacity. A fluid-based model of the dynamics of the TCP and RED was developed by the stochastic theory [18]. This model represents the behavior of the characteristic variables of the network and shows that it accurately captured the qualitative evolution of TCP traffic flows. Based on this TCP model, several congestion control schemes have been developed to improve the performance of communication networks [19].

In this paper, the proposed algorithm will bridge the gap

Manuscript received March 4, 2011.

A. E. Abharian is with the Islamic Azad University, Garmsar branch, Garmsar, Iran (corresponding author to provide phone: 98-21-88457260; fax: 98-21-88410179; e-mail: aeabharian@srbiau.ac.ir).

E. Shakeri, was with Shiraz University, Shiraz, Iran. (e-mail: ehsan.sh1@gmail.com).

between the theoretical mixed optimal H_2/H_∞ control and classical AQM/RED. The proposed H_2/H_∞ RED design consists in finding an internally stabilizing RED that minimizes an H_2 integral performance index subject to an inequality constrained on the H_∞ norm of the closed loop transfer function. That means the solving of the two problems: stability robustness constraint and external disturbance attenuation constraint. This optimization is based on the bacterial foraging (BF) search method, an efficient mechanism for selecting the optimal RED parameter is proposed.

The rest of this research is organized as follows. Section II presents the TCP/ARED model and the AQM objective. A BF-ARED design for AQM is illustrated in Section III. Section IV shows the results of simulations to demonstrate the performance of the proposed control scheme. Finally, brief conclusions are provided in Section V.

II. TCP/RED MODEL

This section describes a brief review for mathematical models of AQM based on a control theoretical approach. A mathematical dynamic model of TCP behavior is developed by Misra *et al.* [11] using fluid-flow and stochastic differential equations. The model becomes as following (1):

$$\begin{cases} \dot{W}(t) = \frac{1}{R(t)} + (1-Q(W)) \left(-\frac{W(t)W(t-R(t))}{2R(t-R(t))} \right) \times \\ p(t-R(t)) + (1+W(t))Q(W) \frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)) \\ \dot{q}(t) = \sum_{i=1}^N \frac{W(t)}{R(t)} - C \end{cases} \quad (1)$$

where $\dot{W}(t)$ denotes the time-derivative of $W(t)$, $\dot{q}(t)$ denotes the time-derivative of $q(t)$, and

W : Expected TCP window size (packets)

q : Expected queue length (packets)

R : Round-trip time (seconds)

C : Link capacity (packets/second)

N : Load factor (number of TCP sessions)

p : Probability of packet mark/drop

t : Time

The function $Q(W)$ determines the probability that one loss is caused by a timeout, given that the window size W at the time of the loss. We use a simplified version of (1) which ignores the TCP timeout mechanism. The non-linear differential equations are following as:

$$\begin{cases} \dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)) \\ \dot{q}(t) = \frac{W(t)}{R(t)} N(t) - C \end{cases} \quad (2)$$

The expected queue length q and the expected TCP window size W are positive value and bounded quantities. And also, the probability of packet mark/drop p takes value only within $[0, 1]$. Fig. 1. show these differential equations in the block diagram which highlights TCP window-control and queue dynamics.

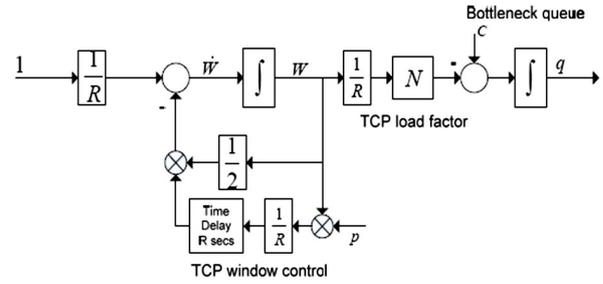


Fig. 1. Block-diagram of TCP model.

For the control theoretical analysis, (2) was approximated as a linearized constant model by small-signal linearization about an operating point W_0, q_0, p_0 . Brief reviews for the linear model are shown below, and see [11] for linearization details. The operating point is defined by $\dot{W} = 0, \dot{q} = 0$ so that it is accomplished by conditions:

$$\begin{cases} \dot{W}(t) = 0 \Rightarrow W_0^2 p_0 = 2 \\ \dot{q}(t) = 0 \Rightarrow W_0 = \frac{R_0 C}{N} \end{cases} \quad (3)$$

And take the first chance of RTT as an input delay:

$$R_0 \cong \frac{q_0}{C} + T_p \quad (4)$$

where T_p is propagation delay (seconds). We finally get the dynamics of TCP/AQM router:

$$\begin{cases} \delta \dot{W}(t) = -\frac{2N}{R_0^2 C} \delta W(t) - \frac{R_0 C^2}{2N^2} \delta p(t - R_0) \\ \delta \dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \end{cases} \quad (5)$$

where

$$\delta W(t) \equiv W - W_0$$

$$\delta q(t) \equiv q - q_0 \quad (6)$$

$$\delta p(t) \equiv p - p_0$$

Thus, the block diagram of linearized AQM control system is shown in Fig. 2. In this diagram, $P_{tcp}(s)$ is the transfer function of the TCP behavior, $P_q(s)$ is transfer function of the queue dynamics.

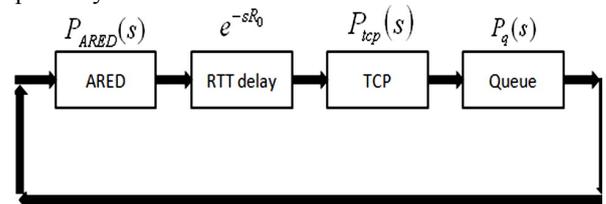


Fig. 2. Block diagram of a linearized AQM as feedback control.

The transfer functions of $P_{tcp}(s)$ and $P_q(s)$ become as following, respectively:

$$\begin{cases} P_{tcp}(s) = \frac{R_0 C^2}{2N^2 \left(s + \frac{2N}{R_0^2 C} \right)} \\ P_q(s) = \frac{N}{R_0 \left(s + \frac{1}{R_0} \right)} \end{cases} \quad (7)$$

where R is Round Trip Time (RTT), N is number of active TCP session and C is link capacity (packet/sec). And also, the

plant transfer function which is denoted as $P(s)$ can be expressed as:

$$P(s) = \frac{C^2 e^{-sR_0}}{2N \left(s + \frac{2N}{R_0^2 C} \right) \left(s + \frac{1}{R_0} \right)} \quad (8)$$

Let the state variable $x(t)$ of equation (5) be defined as:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix}$$

Equation (8) can be represented with the state-space model:

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c p(t - R_0) \\ y(t) = C_c x(t) \end{cases} \quad (9)$$

Where

$$A = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, \quad B_c = \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix} \\ C_c = [0 \quad 1] \quad (10)$$

The RED algorithm has a set of five tunable parameters which are minimum threshold (\min_{th}), maximum threshold (\max_{th}), Maximum Mark/Drop Probably \max_p , queue weight w_q , and Inter Packet Mark/Drop Probably function $p(q)$. It randomly marks/drops packets according to probability curve shown in Fig. 3.

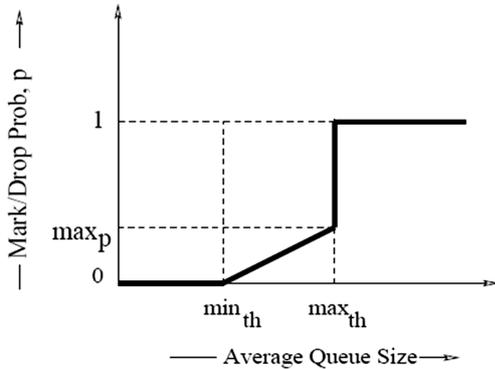


Fig. 3. Marking/ Dropping Curve for RED.

In RED the Exponential Weighted Moving Average (EWMA) \hat{q} of the queue length is computed by the filter $\hat{q} \leftarrow (1 - w_q)\hat{q} + w_q q$. The filter's bandwidth, denoted by K in [9], is given by $K = -C \ln(1 - w_q)$, where C is the router link's bandwidth. The inverse of K is the filter's "averaging time" denoted by T_{red} [16].

$$T_{red} = n \times R_0 \Rightarrow n = \frac{-1}{C \ln(1 - w_q)} \approx \frac{1}{R_0 \times C \times w_q}$$

where R_0 is round-trip time. And transfer function of RED is:

$$P_{ARED}(s) = \frac{L_{red}}{s + \frac{1}{n}} \quad (11)$$

Hence the RED gain, denoted by L_{red} is given by [19]:

$$L_{red} = \frac{\max_p}{\max_{th} - \min_{th}}$$

where \max_p is tunable parameter and \min_{th} and \max_{th} are constant parameters of RED mechanism.

III. BACTERIAL FORAGING OPTIMIZATION

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemotaxis, reproduction, and elimination-dispersal. We briefly describe each of these processes as follows [22].

A. Chemotaxis

In the original BFO, a unit walk with random direction represents a "tumble" and a unit walk with the same direction in the last step indicates a "run". Suppose $\theta^i(j, k, l)$ represents the bacterium at j th chemotactic, k th reproductive, and l th elimination-dispersal step. $C(i)$ is the chemotactic step size during each run or tumble (i.e., run-length unit). Then in each computational chemotactic step, the movement of the i th bacterium can be represented as:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Gamma(i)}{\sqrt{\Gamma^i(i)\Gamma(i)}} \quad (12)$$

where $\Gamma(i)$ is the direction vector of the j th chemotactic step. When the bacterial movement is run $\Gamma(i)$ is the same with the last chemotactic step; otherwise, $\Gamma(i)$ is a random vector whose elements lie in $[-1, 1]$. With the activity of run or tumble taken at each step of the chemotaxis process, step fitness, denoted as $J(i, j, k, l)$, will be evaluated.

B. Reproduction

The health status of each bacterium is calculated as the sum of the step fitness during its life, that is, $\sum_{j=1}^{N_c} J(i, j, k, l)$,

where N_c is the maximum step in a chemotaxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

C. Elimination and Dispersal

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemotaxis and reproduction are not enough for global optima searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, the dispersion event happens after a certain number of reproduction processes. Then some bacteria are chosen, according to a preset probability P_{ed} to be killed and moved to another position within the environment.

D. Step-By-Step Algorithm

In what follows we briefly outline the original BFO algorithm step by step.

Step1. Initialize parameters $n, S, N_C, N_S, N_{re}, N_{ed}, P_{ed}, C(i), (i=1,2,\dots,S), \theta^i$

where

n : dimension of the search space,

S : the number of bacteria in the colony,

N_C : chemotactic steps,

N_S : swim steps,

N_{re} : reproductive steps,

N_{ed} : elimination and dispersal steps,

P_{ed} : probability of elimination,

$C(i)$: the run-length unit (i.e., the size of the step taken in each run or tumble).

Step2. Elimination-dispersal loop: $l = l + 1$.

Step3. Reproduction loop: $k = k + 1$.

Step4. Chemotaxis loop: $j = j + 1$.

Substep4.1. For $i=1,2,\dots,S$, take a chemotactic step for bacterium i as follows.

Substep4.2. Compute fitness function, $J(i, j, k, l)$.

Substep4.3. Let $J_{last} = J(i, j, k, l)$ to save this value since we may find better value via a run.

Substep4.4. Tumble. Generate a random vector $\Gamma(i) \in R^n$ with each element $\Gamma_m(i), m=1,2,\dots,n$, a random number on $[1 \ -1]$.

Substep4.5. Move, let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Gamma(i)}{\sqrt{\Gamma^T(i)\Gamma(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i .

Substep4.6. Compute $J(i, j+1, k, l)$ with $\theta^i(j+1, k, l)$.

Substep4.7. Swimming.

I: Let $m = 0$ (counter for swim length).

II: While $m < N_s$ (if has not climbed down too long), the following hold.

- Let $m = m + 1$.

- If $J(i, j+1, k, l) < J_{last}$, let $J_{last} = J(i, j+1, k, l)$, then another step of size $C(i)$ in this same direction will be taken as (2.2) and use the new generated

- $\theta^i(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$.

- Else let $m = N_s$.

Substep4.8. Go to next bacterium ($i+1$). If $i \neq S$, go to *Substep4.2.* to process the next bacterium.

Step5. If $j < N_C$, go to Step 3. In this case, continue chemotaxis since the life of the bacteria is not over.

Step6. Reproduction

Substep6.1. For the given k and l , and for each $i=1,2,\dots,S$ let

$$J_{health}^i = \sum_{j=1}^{N_C+1} J(i, j, k, l)$$

be the health of the bacteria. Sort bacteria in order of ascending values J_{health} .

Substep6.2. The S_r bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split and the copies that are made are placed at the same location as their parent.

Step 7. If $k < N_{re}$, go to Step2. In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

Step8. Elimination-dispersal: for $i=1,2,\dots,S$ with probability P_{ed} eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. to do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to Step2; otherwise end.

IV. H_2 / H_∞ - RED ROBUST AQM DESIGN PROBLEM FORMULATION

Let consider the AQM system in Fig. 4. The TCP network nominal model $P_n(s)$ to be controlled undergoes perturbation $\Delta P_n(s)$ and the RED, is of the classical type. We define nominal TCP model as:

$$P_n = \frac{C^2}{2N \left(s + \frac{2N}{R_0^2 C} \right) \left(s + \frac{1}{R_0} \right)}$$

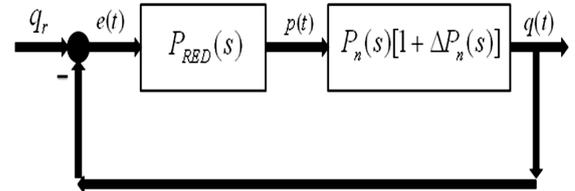


Fig. 4. AQM/RED system with TCP perturbation.

Suppose ΔP_n is bounded according to the relation:

$$|\Delta P_n(j\omega)| \leq |\xi(j\omega)|, \quad \forall \omega \in [0 \ \infty) \quad (13)$$

where the function $\xi(s)$ is stable and known. The robust stability reveals that if a controller $P_{RED}(s)$ is chosen so that nominal closed loop system (free of $\Delta P_n(s)$) in Fig. 4. is asymptotically stable, and the following inequality holds,

$$\left\| \frac{P_n(s)P_{RED}(s)\xi(s)}{1 + P_n(s)P_{RED}(s)} \right\|_\infty \leq 1 \quad (14)$$

then the closed loop system in Fig. 4. is also asymptotically stable under plant perturbation (13), where the H_∞ norm in (14) is defined as:

$$\|P_n(s)\|_\infty = \sup_{\omega \in [0 \ \infty)} |P_n(j\omega)| \quad (15)$$

However, often robust stability alone is not enough in control system design. Optimal tracking performance is also appealing in many practical control engineering applications. Therefore, the mixed H_2 / H_∞ control problem is formulated as follows:

$$\min \left(\int_0^{\infty} e^2(t) dt \right) \quad (16)$$

for the nominal closed loop system in Fig. 1., subject to the robust stability constraint (14), where $e(t)$ is the tracking error. That means, under the constraint (14), the error energy (integral squared error criterion) (16) must be as small as possible. From the above analysis the control design problem involve how to specify a robust RED to achieve the optimal tracking (16) subject to the robust stability constraint (14). Tracking error signal is given by

$$e(s) = \frac{q_r(s)}{1 + P_{RED}(s)P_n(s)} = \frac{A(s)}{B(s)}$$

Using Parseval's theorem, one obtains the expression of the integral criterion:

$$\begin{aligned} J &= \min \left(\int_0^{\infty} e^2(t) dt \right) = \min \left(\frac{1}{2\pi j} \int_{-j\infty}^{j\infty} e(-s)e(s) ds \right) = \\ &= \min \left(\frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{q_r(-s)q_r(s) ds}{[1 + P_n(-s)P_{RED}(-s)][1 + P_n(s)P_{RED}(s)]} \right) = \\ &= \min \left(\frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{B(s)B(-s)}{A(s)A(-s)} ds \right) \end{aligned} \quad (17)$$

where $A(s)$ and $B(s)$ are Hurwitz polynomials of s with appropriate degree. The minimization problem in the equation (17) can be solved with the aid of the residue theorem. Let

$$A(s) = \sum_{k=0}^4 a_k s^k, \quad B(s) = \sum_{k=0}^3 b_k s^k$$

where in AQM/RED system, $A(s)$ and $B(s)$ define as

$$\begin{cases} A(s) = s^4 + \left(\frac{1}{n} + \frac{R_0 C + 2N}{R_0^2 C} \right) s^3 + \left(\frac{2N}{R_0^3 C} + \frac{R_0 C + 2N}{n R_0^2 C} \right) s^2 + \\ \quad \left(\frac{2N}{R_0^3 C n} \right) s \\ B(s) = \frac{q_r L_{red}}{2N} \end{cases} \quad (18)$$

In (18) we suppose that $q_r(s) = \frac{q_r}{s}$. We have

$$\begin{cases} a_0 = 0, a_1 = \frac{2N}{R_0^3 C n}, a_2 = \frac{2N}{R_0^3 C} + \frac{R_0 C + 2N}{n R_0^2 C}, \\ a_3 = \frac{1}{n} + \frac{R_0 C + 2N}{R_0^2 C}, a_4 = 1 \\ b_0 = \frac{q_r L_{red}}{2N}, b_1 = 0, b_2 = 0, b_3 = 0 \end{cases}$$

The form of J_4 define as:

$$J_4(\max_p) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{\sum_{k=0}^3 b_k s^k \sum_{k=0}^3 b_k (-s)^k}{\sum_{k=0}^4 a_k s^k \sum_{k=0}^4 a_k (-s)^k} \quad (19)$$

Then, based on the residue theorem, the robust

performance in (19) must be of the following form

$$J_4 = \min(J_4(\max_p)) \quad (20)$$

From the definition (15), the constraint in (14) can be expressed by:

$$\begin{aligned} \left\| \frac{P_n(s)P_{RED}(s)\xi(s)}{1 + P_n(s)P_{RED}(s)} \right\|_{\infty} &= \\ \sup_{\omega \in [0, \infty)} \sqrt{\frac{P_n(-j\omega)P_n(j\omega)P_{RED}(-j\omega)P_{RED}(j\omega)\xi(-j\omega)\xi(j\omega)}{[1 + P_n(-j\omega)P_{RED}(-j\omega)][1 + P_n(j\omega)P_{RED}(j\omega)]}} &\leq 1 \end{aligned}$$

or

$$\begin{aligned} \left\| \frac{P_n(j\omega)P_{RED}(j\omega)\xi(j\omega)}{1 + P_n(j\omega)P_{RED}(j\omega)} \right\|_{\infty} &= \sup_{\omega \in [0, \infty)} \sqrt{\frac{w(\omega)}{v(\omega)}} = \sqrt{\sup_{\omega \in [0, \infty)} \frac{w(\omega)}{v(\omega)}} = \\ \sqrt{\sup_{\omega \in [0, \infty)} \Phi(\omega)} &\leq 1 \end{aligned}$$

where $w(\omega)$ and $v(\omega)$ are some appropriate polynomials of ω . The physical meaning of the above relation is that if the largest peak of $\phi(\omega) = w(\omega)/v(\omega)$ is less than 1, then the system in Fig. 4. is stable under TCP model perturbation. Generally speaking, to scan $\omega \in [0, \infty)$ to find the peaks of $\phi(\omega)$ is not an easy task. Actually, the peaks of $\phi(\omega)$ occur at the points which must satisfy the following equation:

$$\begin{aligned} \frac{d\phi(\omega)}{d\omega} = 0 &\Rightarrow w(\omega) \frac{dv(\omega)}{d\omega} - v(\omega) \frac{dw(\omega)}{d\omega} = 0 \\ &\Rightarrow \prod_{i=1}^n (\omega - \alpha_i) = 0 \end{aligned}$$

Therefore, only the real roots α_i of the above equation need to be found. So, the robust is equivalent with stability constraint in (14)

$$\sqrt{\max_{\alpha_i} \frac{w(\alpha_i)}{v(\alpha_i)}} \leq 1 \quad (21)$$

The design procedure is, therefore, a minimization problem (20) under the inequality constraint (21). The algorithm follows the steps:

Step1. Initialize parameters $n, S, N_C, N_S, N_{re}, N_{ed}, P_{ed}, C(i), (i=1,2,\dots,S), \theta^i$ and random values of RED parameter. where, n : Dimension of the search space (Each Parameter of RED AQM), S : The number of bacteria in the population, N_C : chemotactic steps, N_{re} : The number of reproduction steps, N_{ed} : the number of elimination-dispersal events, P_{ed} : elimination-dispersal with probability, $C(i)$: the size of the step taken in the random direction specified by the tumble. The RED parameter is searched in the range of stabilizable space determine by stability criteria.

Step2. Elimination-dispersal loop: $l = l + 1$.

Step3. Reproduction loop: $k = k + 1$.

Step4. Chemotaxis loop: $j = j + 1$.

Step5. If $j < N_C$, go to *step 3*. In this case, continue chemotaxis, since the life of the bacteria is not over.

Step6. Reproduction.

Step7. Check if the relation (21) is fulfilled.

Step8. If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

Step9. Elimination-dispersal: for $i = 1, 2, \dots, S$ with probability P_{ed} , eliminate and disperse each bacterium. To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to step2; otherwise end. The applied flowchart illustrates in Fig. 5. and 6.

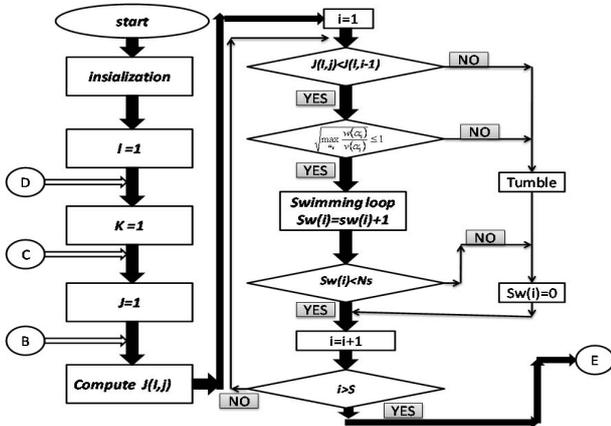


Fig. 5. flowchart (part1).

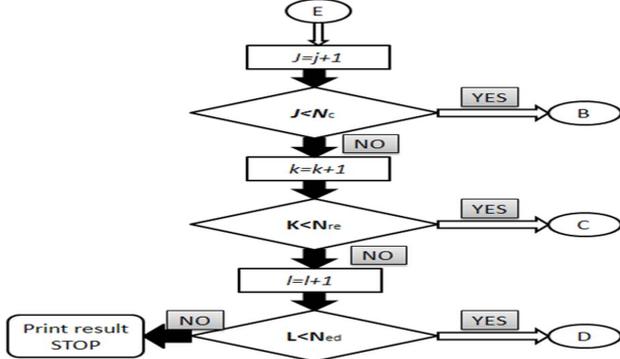


Fig. 6. flowchart (part2).

V. NUMERICAL EXAMPLE

In this section, we illustrate the robust BF-RED design for TCP/AQM. For a TCP/AQM network modeled by equation (2), it is assumed that $N=100$ homogeneous TCP connections and shares one bottleneck link with a capacity of 10 Mbps, i.e. $C=1250$ (packets/second). Furthermore, the desired queue size was $q_r = 150$ packets and, therefore $0 \leq \max_p \leq 1$.

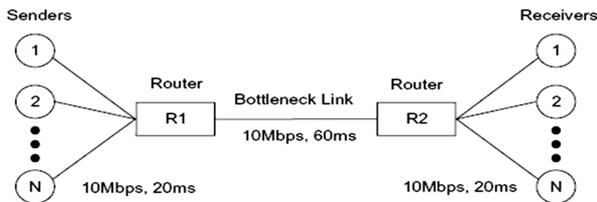


Fig. 7. Dumbbell network topology.

The parameters selected for the proposed bacterial foraging algorithm are as follows and the flow chart of the algorithm is presented in Fig. 5. and 6.

- Number of bacteria $S : 20$
- Depth of attractant: 0.01
- Number of chemotactic steps $N_C : 30$

- Width of attractant: 0.04
- Swimming length $N_S : 4$
- Height of repellent: 0.01
- Number of reproduction steps $N_{re} : 10$
- Width of repellent: 10.0
- Number of elimination and dispersal events $N_{ed} : 50$
- Probability of elimination and dispersal $P_{ed} : 0.02$
- Run length vector initial value $C(i) : 0.05$

After a series of bacterial foraging manipulations, the convergence curve of the ISEC value vs. iteration is described in Fig. 7., and its final value is $J_4(\max_p^*) = 0.345$. Correspondingly, the robust BF-RED gain were obtained

$$L_{red}^*(\max_p^*) = 2.3456 \times 10^{-8}$$

In this simulation experiment, the results obtained with different AQM schemes are shown in Fig. 9. It is clear that the presented BF-RED and ARED successfully achieve the desired queue length and rapidly stabilize the network at the operating point. Although the transient responses of PI can maintain the queue length around the desired value, too inactive and serious overshoots occurred in the PI schemes. However, the REM scheme cannot maintain the desired queue length because there is no parameter of the target queue length in its control mechanism.

Fig. 10. shows the queue length as a function of the network load. As the load increases, the queue length also increases as expected. The PDF-AQM produced longer queues than RED. Under loads of 0.35, the queue produced by robust AQM was 41.55% longer than the one produced by RED. While under loads of 1.0, the queue length produced by robust AQM was almost three times that of the one produced by RED.

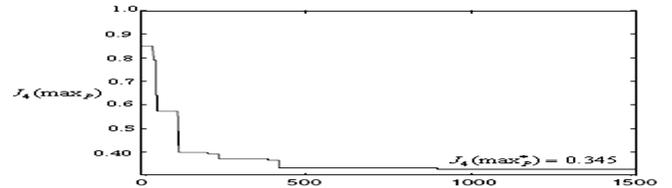


Fig. 8. objective function.

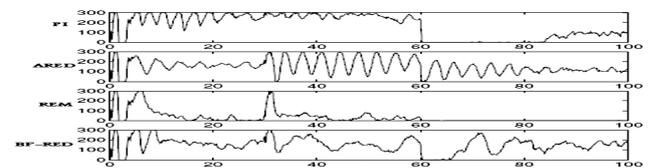


Fig. 9. Responses of queue length (in packets) on dumbbell network topology subject to dynamic traffic changes.

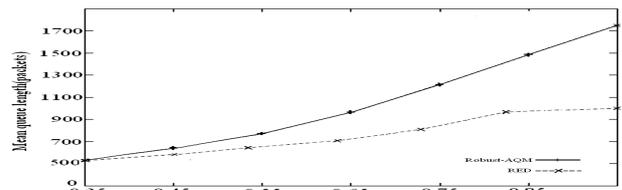


Fig. 10. The mean queue length as a function of the network load.

Fig. 11. shows, respectively, the mean goodput per connection given as both the percentage of the link capacity and as a function of the network load. The per connection goodput given by PDF-AQM is 38.2% higher under loads of

0.34 than those given by RED. Under loads of 0.85, such a difference can be on the order of 100%. These results show the advantage of using PDF-AQM for individual connections despite larger queues.

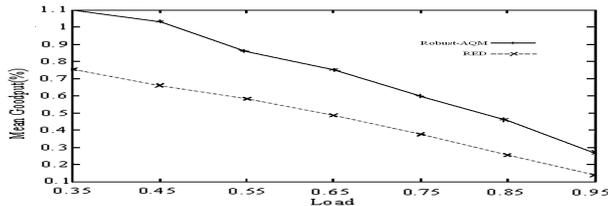


Fig. 11. The mean goodput per connection as a function of the network load.

VI. CONCLUSION

A robust mixed H_2/H_∞ -RED for AQM has been presented to ensure both high utilization and low packet loss rate by regulating the queue length in an Internet router. Based on the bacterial foraging optimization algorithm, a simple and effective robust BF-RED has been proposed. RED parameter can be directly obtained by solving the specified optimization problem via bacterial foraging algorithm. The performance of the proposed AQM congestion control scheme has been evaluated in various network scenarios via the numerical simulations. The simulation results reveal that the proposed scheme is superior to the existing AQM schemes. Results derived from simulation showed that higher goodput per connection was achieved when the robust AQM controller was used as compared to RED. Moreover, lower numbers of retransmission timeout and RTT values were obtained when robust AQM was used. These results indicate that robust AQM is a potential candidate for adoption as a conventional AQM policy in networks that use a TCP protocol.

REFERENCES

- [1] C. L. Lin, H. W. Su, "Intelligent control theory in guidance and control system design: an Overview," Proc. Natul. Sci., Counc. ROC(A), vol. 24, no. 1, pp. 15-30, 2000.
- [2] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part I: A unified formulation;" "-Part II: Application example," IEEE Trans. System, Man, Cybern. A: Systems and Humans, vol. 28, no. 1, pp.26-47, Jan. 1998.
- [3] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control system engineering: A survey," Control Eng. Practice, vol. 10, pp.1223-1241, 2002.
- [4] Z. Michalewicz, Genetic Algorithms+Data Structures=Evolution Programs. New York: Springer-Verlag, 1999.
- [5] H. Yoshida, K. Kawata, and Y. Fukuyama, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment", IEEE Trans. Power Syst., Vol. 15, pp. 1232-1239, Nov. 2000.
- [6] S. Tsutsui and D. E. Goldberg, "Simplex crossover and linkage identification: Single-stage evolution vs. multi-stage evolution," in Proc. IEEE Int. Conf. Evolutionary Computation, HI, 2002, pp. 974-979.
- [7] D. H. Kim and J. H. Cho, "Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization," in Proceedings of the 3rd International Atlantic Web Intelligence Conference (AWIC '05), vol. 3528 of Lecture Notes in Computer Science, pp. 231-235, Lodz, Poland, June 2005.
- [8] J. H. Holland, Adaptation in Nature and Artificial System, MIT Press, Cambridge, Mass, USA, 1992.
- [9] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, NY, USA, 1989.
- [10] D. H. Kim and C. H. Cho, "Bacterial foraging based neural network fuzzy learning," in Proceedings of the Indian International Conference on Artificial Intelligence, pp. 2030-2036, Pune, India, December 2005.
- [11] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," IEEE Transactions on Evolutionary Computation, vol. 9, no. 1, pp. 61-73, 2005.
- [12] A. W. Mohammed and N. C. Sahoo, "Efficient computation of shortest paths in networks using particle swarm optimization and noising metaheuristics," Discrete Dynamics in Nature and Society, vol., Article ID 27383, 25 pages, 2007.
- [13] lin Nicolae Calistru, "Mixed H2/Hinf PID Robust Control via Genetic Algorithms and MATHEMATICA Facilities", "-----", ".
- [14] Floyd, S., Gummadi, R., & Shenker, S. (2001). Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. Available: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>.
- [15] W. Feng, K. G. Shin, D. D. Kandlur and D. Saha, "The BLUE active queue management algorithms," IEEE/ACM Transactions on Networking, 10(4), 89-102. 2002.
- [16] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, 1(4), 397-413. 1993.
- [17] D. Lin, and R. Morris, "Dynamics of random early detection," In Proceedings of ACM SIGCOM'97 (pp. 127-137). Cannes, 1997.
- [18] T. J. Ott, T. V. Lakshman and L. H. Wong, "SRED: Stabilized RED," In Proceedings of IEEE INFOCOM'99 (pp. 1346-1355). NewYork, 1999.
- [19] 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," In Proceedings of ACM/SIGCOM (pp. 151-160). Stockholm, 2000.
- [20] C. Kuo .Chen, H. H. Kuo, J. J. Yan and T. L. Liao, "GA-based PID active queue management control design for a class of TCP communication networks", Expert Systems with Applications 36 (2009) 1903-1913.
- [21] D. H. Kim, A. Abraham and J. H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177 (2007) 3918-3937.
- [22] D. H. Kim and Z. Wu, "Robust Tuning of Embedded Intelligent PID Controller for Induction Motor Using Bacterial Foraging Based Optimization," ICESS 2004, LNCS 3605, pp. 137-142, 2005.

Amir Esmaili Abbarian is a P.hD student with specialization in Control, Department of Engineering at Islamic Azad University, Science and Reserch branch. He received his B.Sc degree from Islamic Azad University, Science and Reserch branch, Tehran in 2003 and M.Sc. from Islamic Azad University Science and Research branch in 2005. His research areas of interest are Stochastic Control, Advanced Automation and Computer Networks.

Ehsan Shakeri received his B.Sc degree from Shahid Sattari Air Force University, Tehran in 2007 and M.sc. degree with specialization in Control from faculty of e-Learning at Shiraz University, Shiraz in 2010. His research interests include iterative learning control, Kalman filtering and nonlinear control.