A Particle Swarm Optimization Approach for Optimum Design of First-Order Controllers in TCP/AQM Network Systems

Sana Testouri
Department of Electric Engineering, National School of Engineering of Tunis

Karim Saadaoui
Department of Electric Engineering, National School of Engineering of Tunis

Mohamed Benrejeb
Department of Electric Engineering, National School of Engineering of Tunis

ABSTRACT
This paper proposes a Particle Swarm Optimization (PSO) method for determining the optimal parameters of a first-order controller for TCP/AQM system. The model TCP/AQM is described by a second-order system with time delay. First, the analytical approach, based on the D-decomposition method and Lemma of Kharitonov, is used to determine the stabilizing regions of a first-order controller. Second, the optimal parameters of the controller are obtained by the PSO algorithm. Finally, the proposed method is verified and compared with the PI controller using the Network Simulator, NS-2.

General Terms
Time Delay, Stability, First-Order Controller, TCP/AQM.

Keywords
Time delay, TCP/AQM, PSO.

1. INTRODUCTION
During the last few years, the number of users in internet has grown rapidly, which leads problems in network in communication (because high packet loss rates, increased delays ...), indeed in network, the packet loss indicates congestion which happens when the packet flow is greater than the link capacity. In fact, the congestion-control mechanism becomes indispensable in an over-charged network. TCP (Transmission Control Protocol), has been the basis of control congestion. It adopts the end-to-end window-based flow control to avoid congestion [1]. Recently, we assist to a growing interest of designing AQM (Active Queue Management) using control theory. The goal of AQM is to maintain shorter queuing delay and higher throughput by dropping packets at intermediate nodes. It has therefore attracted attention in the research for transmission control protocol (TCP) of end-to-end congestion control. Random early detection (RED) [2] is the first well known AQM algorithm, which aims to drop packets with a certain probability a function of the average queue size. Furthermore, it is difficult to obtain adequate values of RED parameters to provide satisfactory performance in terms to provide of overall quality of service (QoS). Therefore, feedback control principles appear to be an appropriate tool in the analysis and design of AQM strategies. Some controllers for AQM based on feedback control theory have been developed, such as Integral (I) controller and PI controller in [3], Proportional-Derivative (PD) controller in [4], and PID controller in [5, 6]. The first-order controllers derived of phase-lead and phase – lags, have been used in the last decades. In fact, the problem of determining all stabilizing first-order controllers with analytical methods for delay free linear time-invariant systems has been recently solved in [7, 8, 9] Generally, dynamic performance of PI/PID and first-order controllers are unable to satisfy some performance specification of the transient performance (including small rise time, small settling time and small overshoot) and small steady state error simultaneously in some situation. For resolved this problem an algorithm of improving the performance, based on Particle Swarm Optimization (PSO) is proposed, to determine the optimal parameters of the first–order controller for TCP /AQM system. Indeed, the model of TCP/AQM is described by a second-order system with time delay [10]. Nevertheless, several approaches have been proposed to determine the stabilizing region of controller parameters for TCP/AQM model without take into account the delay in the closed loop [11, 12, 13] Our objective is to determining stabilizing optimal parameters of a first-order controller of the TCP/AQM model with time delay, using the PSO algorithm, for guarantee some performance for a high performance, this algorithm is named PSO/first-order controller. This paper has been organized as follows: in section 2, we introduce first the linear control system model. Next, we describe the AQM control law using a first-order controller with a mathematical formulation of its digital implementation. The stabilizing regions in the parameter space of a first-order controller for TCP/AQM system with time delay are determined with the analytical method in section 3. In section 4, the PSO method is proposed to obtain the optimal parameters of the controller. Simulation results both in Matlab and NS-2 are given in section 5. Finally, the conclusion is drawn in section 6.

2. MODEL TCP/AQM
2.1 TCP flow Control Model
The dynamic model of TCP/AQM is developed in [14] using a fluid flow and stochastic differential equation analysis. In [3, 10] the model is simplified and ignores the time out mechanism and slow start phase of TCP. This model is described by the following non-linear differential equations:

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

where $W(t)$ denotes theTCP window size (packet), $q(t)$ denotes the queue length in the router (packet); $\dot{W}(t) = \frac{dW(t)}{dt}$; $\dot{q}(t) = \frac{dq(t)}{dt}$

$p(t)$ denotes the probability packet marking/dropping $\left(p(t)\in[0,1]\right)$; $R(t)$ denotes the round-trip time; $C(t)$ denotes the link capacity (packet/s); T_p denotes the propagation delay (s); $N(t)$ denotes the load factor (number of TCP sessions).

The first differential equation in (1) describes the TCP window control dynamic and the second equation models the bottleneck queue length. The queue length and window size are positive, bounded quantities, i.e., $q \in \left[0,\bar{q}\right]$, $W \in [0,\bar{W}]$ where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size, respectively. Also, the marketing probability p takes value only in $[0,1]$.The dynamic model of TCP/AQM (1) is linearized in [3, 10], we illustrated the linear TCP/AQM dynamics in the linear TCP/AQM dynamics in a block diagram in Fig. 1. According to Fig. 1, the TCP/AQM model can be expressed by the transfer function $G(s)$.

$$G(s) = G_{TCP}(s)G_{queue}(s)$$

$$G_{TCP}(s) = \frac{\frac{R_0C^2}{2N}}{(s+\frac{2N}{R_0^2C})}e^{-sR_0} \qquad G_{queue}(s)=\frac{\frac{N}{R_0}}{(s+\frac{1}{R_0})} \tag{2}$$

where $G_{TCP}(s)$ denotes the transfer function from loss probability δp to window size δW and $G_{queue}(s)$ relates δW to queue length δq. The term e^{-sR_0} is the Laplace transform of the time delay in the delayed loss probability $\delta p(t-R_0)$ notes the queue's dynamic. The network parameters $\{N,C,R_0\}$ are positive, and $R_0>0$ is the time delay [10].It is clear that the linearized TCP/AQM model is a second order plant with time delay.

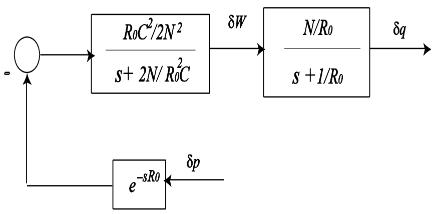


Fig1: Block diagram of the linerarizd TCPflow- control model

2.2 AQM Control System Design

In this section, we present first the AQM control law using the first-order controller. Then we demonstrate the implemented digital of this controller.

2.2.1 First-Order Controllers in AQM system

In fig 2, we give a closed-loop feedback control system depiction of AQM, where $C(s)$ is the AQM controller, $G(s) = G_{TCP}(s)G_{queue}(s)$ is the plant dynamics, q_0 is to the desired queue length around which the controller should stabilize q

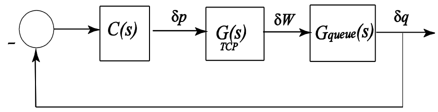


Fig 2: Block diagram of AQM control system

Transfer function of first order controller for AQM is described as follows

$$C(s)=\frac{\alpha_2 s+\alpha_3}{s+\alpha_1} \tag{3}$$

2.2.2 Digital implementation of the first-order controller

The objective of an AQM controller is to mark packets with a probability p .The marking probability is calculated according to the first-order controller and it is a function of the difference between the instantaneous queue length and the desired queue length to which we want to regulate, where δq is given by $\delta q = q - q_0$ and, we assume $p_0 = 0$, which makes $\partial p = p$, The first order controller transfer function is in the form (1.1) , we can write

$$\frac{p}{\delta q} = \frac{\alpha_2 s+\alpha_3}{s+\alpha_1} \tag{4}$$

In order to evaluate the effectiveness and performance of the proposed first order controller by simulation, we use the ns 2 simulator which presents a discrete event simulator. In fact, the first order controller is not implemented in the core of Network Simulator, NS-2[15]. Hence, for the digital implementation of the first-order controller, we need to convert the transfer function (4) describe in the s–domain (Laplace Transform) into a z- transform and choose sampling frequency f_s as 10-20 times the loop bandwidth. In our case, we choose $f_s = 160Hz$ [3, 10].

The first-order controller transfer function is in the form (5), and in z-domain it becomes

$$\frac{p}{\delta q} = \frac{A\left(1-\alpha z^{-1}\right)+B\left(1-z^{-1}\right)}{1-\alpha z^{-1}} \tag{5}$$

where $A=\alpha_3$ $B=\alpha_2\alpha_1-\alpha_3$ and $\alpha = e^{-\alpha_1 T_s}$

This transfer function (5) can be converted to the following difference equation for $t = kT_s$, where $T_s = 1/f_s$,

$$p(kT) = a_1\delta_q(kT_s) - b_2\delta_q((k-1)T_s) + p((k-1)T_s)*\alpha \tag{6}$$

The digital implementations of a first-order controller tested in NS-2 can be described by the following pseudo code called at every sampling time.

$$p = a_1(q-q_0) - b_1(q_{old}-q_0) + p_{old}\alpha$$
$$p_{old} = p$$
$$q_{old} = q$$
$$\alpha = e^{-\alpha_1 T} \tag{7}$$

where $a_1 = \alpha_2$, $b_1 = \alpha\alpha_3 + \alpha_2\alpha_1 - \alpha_3$ *et* $\alpha = e^{-\alpha_1 T}$

It is clear that the pseudo code (7) dependent of the first-order controllers parameters $(\alpha_1,\alpha_2,\alpha_3)$.

3. STABILIZING FIRST-ORDER CONTROLLERS

In this section, the aim is to determine the stabilizing regions of first-order controllers for TCP/AQM model with time delay via parametric methods. We consider the closed–loop AQM system in fig2, where $G(s)$ denotes the function transfer of the TCP/AQM plant and $C(s)$ denotes the transfer function of the first-order controller (3)

$$G(s) = G_{TCP}(s)G_{queue}(s) = \frac{B}{Q(S)}e^{-R_0 s} \qquad (8)$$

where $B = \frac{C^2}{2N}$, $Q(s) = (s + \frac{2N}{R_0^2 C})(s + \frac{1}{R_0})$

The network parameters $\{N,C,R_0\}$ are positive, and R_0 is the time delay. The closed-loop AQM system is a second -order system with time delay, whose characteristic equation is

$$1+C(s)G(s)=0 \qquad (9)$$

which leads to the following characteristic quasi-polynomial.

$$V^*(s) = (s+\alpha_1)Q(s) + B(\alpha_2 s + \alpha_3)e^{-sR_0} \qquad (10)$$

Multiplying both sides of by $e^{R_0 s}$ yields

$$V(s) = (s+\alpha_1)Q(s)e^{R_0 s} + B(\alpha_2 s + \alpha_3) \qquad (11)$$

As $e^{R_0 s}$ does not have any finite zeros [16], the zeros of $V^*(s)$ are identical to those of $V(s)$. The characteristic quasi-polynomial $V^*(s)$ of the closed-loop AQM system is stable if and only the zeros of $V(s)$ are in open left hand plane (LHP).Then, $V(s)$ is defined as Hurwitz or stable. Determining stabilizing controller parameters $(\alpha_1,\alpha_2,\alpha_3)$ of will be done in the next section.

3.1 Determining the Admissible Ranges of α_1

The characteristic quasi-polynomial (11) dependents of three $(\alpha_1,\alpha_2,\alpha_3)$ parameters, in fact to finding the stabilizing regions of first-order controllers present difficulty to determine analytical, for these reason, our aims is to determine the admissible ranges of the first parameters α_1 then to determine the remaining two parameters (α_2,α_3). Therefore, for calculating the admissible values of α_1, the following Lemma 1 is given, which allow give a condition for the stability of $V(s)$, where $V'(s)$ denotes the derivative of $V(s)$.

Lemma 1. [17] Consider the quasi-polynomial

$$\Delta(s) = \sum_{i=0}^{n}\sum_{l=1}^{r} h_{il} s^{n-i} e^{\tau_l s} \qquad (12)$$

such that $\tau_1 < \tau_2 < \ldots < \tau_r$, with main term $h_{0r} \neq 0$ and $\tau_1 + \tau_r > 0$ .If $V(s)$ is stable then $V'(s)$ is also a stable quasi-polynomial.

Now, using Lemma 1, if $V(s)$ is stable then $V'(s)$ is also a stable quasi-polynomial, where

$$V'(s) = \left[\left(R_0(s+\alpha_1)+1\right)Q(s) + (s+\alpha_1)Q'(s)\right]e^{R_0 s} + B\alpha_2 \qquad (13)$$

Note that only two parameters (α_1,α_2) appear in the expression of $V'(s)$.

Repeating the same reasoning: if $V'(s)$ is stable, then is also stable, $V''(s)$ given by

$$\Delta''(s,\alpha_1) = \Big\{\left[sQ''(s) + (2R_0 s+2)Q'(s) + (R_0^2 s + 2R_0)Q(s)\right] + \alpha_1\left[Q''(s) + 2R_0 Q'(s) + R_0^2 Q(s)\right]\Big\}e^{R_0 s} \qquad (14)$$

Note that only one controller parameter α_1 appears in the expression of $V''(s)$, moreover the term $e^{R_0 s}$ has no finite roots, so stability of $V''(s)$ is equivalent to stability in expression (13) without term $e^{R_0 s}$.To sum up, using the condition of Lemma 1, we can get an admissible stabilizing range for the controller parameter α_1 [18].

3.2 Stabilizing Regions in the plane of (α_2,α_3)

Once the admissible values of α_1 is fixed within the range determined the above procedure, the set of the stabilizing regions in the plane of the parameters (α_2,α_3) is determined by using the D-decomposition method [19], which is described in what follows

- Substitute s by $j\omega$ and set the real and imaginary parts of $V'(s)$ to zero.
- The (α_2,α_3) plane can be partitioned into root-invariant regions.
- Stabilizing regions in the (α_2,α_3) plane can be determined by choosing a point inside the root-invariant regions and applying classical methods.

Evaluating the characteristic function at the imaginary axis is equivalent to replacing s by $j\omega$, $\omega \geq 0$ in (11), which gives

$$V(j\omega) = \left[(j\omega+\alpha_1)\left(R(\omega) + jI(\omega)\right)\right]\left(\cos(R_0\omega) + j\sin(R_0\omega)\right) + B(\alpha_2 j\omega + \alpha_3) \qquad (15)$$

where $R(\omega)$ and $I(\omega)$ are the real and the imaginary part of $Q(j\omega)$

Three cases will be investigated.

Case1. Setting $\omega = 0$, this leads to the following equations

$$\alpha_3 = -\frac{1}{B} R(0)\alpha_1$$

(16) Case2. For $\omega > 0$, the following pair of (α_2, α_3) is be calculated for each fixed value of α_1

$$\begin{cases} \alpha_2 = \frac{1}{B}\left[I(\omega) - \alpha_1 \frac{R(\omega)}{\omega} \right] \sin(R_0\omega) - \left[R(\omega) + \alpha_1 \frac{I(\omega)}{\omega} \right] \cos(R_0\omega) \\ \alpha_3 = \frac{1}{B}\left[\omega I(\omega) - \alpha_1 R(\omega) \right] \cos(R_0\omega) + \left[\omega R(\omega) + \alpha_1 I(\omega) \right] \sin(R_0\omega) \end{cases} \quad (17)$$

By sweeping over all values $\omega > 0$, the (α_2, α_3) plane can be partitioned into root–invariant regions; therefore stabilizing regions can be determined. The stabilizing region determined by expression (16) and (17) guaranteed only that stability of the TCP/AQM model for closed loop, thus it is unable to satisfy some performance. In order to achieve good control performance of the TCP/AQM model, we will propose, in next section, a method Particle Swarm Optimization (PSO) to search efficiently the optimal parameters of the first -order controller.

4. OVERVIEW OF PARTICLE SWARM OPTIMZATION

4.1 Particle Swarm Optimization Algorithm

PSO algorithm was developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer), which is a robust stochastic optimization technique based on the movement and intelligence of swarms. PSO relies on the exchange of information between particles of the population called swarm moving in the search space looking for the best solution [20]. PSO technique can generate a high-quality solution within shorter calculation time and stable convergence characteristic than other stochastic methods [21]–[22]. It guides searches using a population constructed by many particles rather than individuals. Generally, PSO is characterized as a simple concept, easy to implement, and computationally efficient. In the PSO algorithm, each particle, candidate solution to the optimization problem, is characterized by a random position and velocity. During flight, each particle updates its own velocity and position, by moving its trajectory towards its best solution (fitness) and by leaving a track of its coordinates in the problem space which are associated with the best solution that is achieved so far. This value is called $pbest$. Each particle also modifies its trajectory towards the best previous position attained by any member of its neighborhood [23]. Each particle also modifies its trajectory towards the best previous position attained by any member of its neighborhood, which represent another best value called $gbest$. The PSO concept consists of considering a population (swarms) of the n_p particle moving randomly in the search space looking for the best solution. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $pbest_{i,g}$ to $gbest_g$ as shown in the following formulas:

$$\begin{cases} v_{i,d}^{(t+1)} = \omega.\, v_{i,d}^{(t)} + c_1 r_1(\,)\,(pbest_{i,d} - x_{i,g}^{(t)}) + c_2 r_2(\,)\,(gbest_{i,d} - x_{i,d}^{(t)}) \\ x_{i,d}^{(t+1)} = x_{i,d}^{(t)} + v_{i,d}^{(t+1)} \end{cases} \quad (18)$$

$d = 1, 2, ..., m$; $i = 1, 2, ..., n_p$

where n_p is the number of particles in a group; m is the number of members in a particles; t is the pointer of iterations (generations); $V_{i,d}^{(t)}$ is the velocity of particle i at iteration t such as $V_d^{\min} \le v_{i,d}^{(t)} \le V_d^{\max}$; W is the inertia weight factor; c_1, c_2 is the acceleration constant; $r_1(\,), r_2(\,)$ is the random number between 0 and 1; $x_{i,d}^{(t)}$ is the current position of particle i at iteration t; $pbest_i$ is the best position discovered by the particle until the iteration t; $gbest$ is the global best particle position of the entire population. Each particle $P_i\ (i = 1, 2, ...n_p)$ is characterized by the current position $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,d})$ of P_i particle in the d-dimensional space; its velocity $v_i = (v_{i,1}, v_{i,2}, ..., v_{i,d})$; the previous position $pbest$ of the P_i particle is recorded and represented as $pbest_{i,...,d}$;

the index of best particle among all of the particles in the group is represented by the $gbest_d$. To damp the velocity and to reduce uncontrollable oscillations of the particles, a method is incorporated into the system [24] limiting the velocity to a maximum value predetermined $V^{\max}$. This constraint is defined so that the particles do not move too quickly, for which regions searched be between the present position and the target position. In fact, if $V^{\max}$ is too high, particles might explore the good solutions, but if $V^{\max}$ is too small, particles may not explore sufficiently beyond local solutions. The $V^{\max}$ parameter thus improves the resolution of the search and arbitrarily limits the velocities of each particle $V^{\max}$. Much research which employed PSO algorithm $V^{\max}$ was often set at 10–20% of the dynamic range of the variable on each dimension [23].The constant c_1 and c_2 represents the weighting of the stochastic acceleration terms that pull each particle toward $pbest$ and $gbest$ positions. In some works, these parameters are determined from the following equation

$$0 \le c_1 + c_2 \le 4 \quad (19)$$

In our case, we adopt $c_1 = c_2 = 2$ which verify equation (19) [25].The inertia weight factor w is used to defined the exploration capacity of each particle, hence to improve the convergence of the PSO algorithm, in general, w is according to the following equation

$$w = \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (20)$$

where $iter_{\max}$ is the maximum number of iterations (generations), and $iter$ is the current number of iterations.

4.2 Implementation of PSO/first-order controller

In this part, we used the new performance criterion in the time domain [23] include the overshoot M_p, rise time t_r ,settling time t_s and steady state error E_{ss} or determining the optimal parameters of the first-order controllers for TCP/AQM network systems. The first-order controller using the PSO algorithm is developed to improve a good step response that will result in performance criteria minimization in the time domain. Therefore, a new performance criterion $W(\alpha)$ is defined as [23]

$$W(\alpha)=\left(1-e^{-\beta}\right).\left(M_p+E_{ss}\right)+e^{-\beta}.\left(t_s-t_r\right) \tag{21}$$

where $\alpha=\left(\alpha_1,\alpha_2,\alpha_3\right)$ are three parameters of the first-order controller to compose an individual and β is weighting factor. For used the PSO method, we adopt the term "individual" to replace the "particle" and the "population" to replace the "group" in this paper. The members $\alpha=\left(\alpha_1,\alpha_2,\alpha_3\right)$ are assigned as real values. If there are individuals in a population, then the dimension of a population is $n\times 3$. The performance criterion $W(\alpha)$ can satisfy the designer requirements using β . In fact, if $\beta>0.7$, the overshoot and steady states error are reduced, but if $\beta<0.7$, the rise time and settling time are reduced [23]. In general, the β is defined in the range $[0.8,1.5]$ [26]. In our case, due to trials, β is set to 1.5 to optimum the step response of TCP/AQM network systems. Now, we define the reciprocal of performance criterion $W(\alpha)$ by the fitness function f , as being the evaluation value of each individual in population. It implies the smaller $W(\alpha)$ the value of individual $\alpha=\left(\alpha_1,\alpha_2,\alpha_3\right)$, the higher its evaluation value

$$f(\alpha)=\frac{1}{W(\alpha)} \tag{22}$$

In many works, the Routh–Hurwitz criterion was employed to test the closed-loop system stability to limit the evaluation value of each individual of the population within a reasonable range [23]. If the individual satisfies the Routh–Hurwitz stability test applied to the characteristic equation of the system, then it is a feasible individual and the value of is small. In the opposite case, the value of the individual is penalized with a very large positive constant. In our case it is not necessary to test the stability because the stabilizing regions of parameters $\left(\alpha_1,\alpha_2,\alpha_3\right)$ are determined in the previous section. The proposed PSO method each particle contains three members $\left(\alpha_1,\alpha_2,\alpha_3\right)$. It means that the search space has three dimension and particles must 'fly' in a three. Our objective here is to minimize the performance criteria such as the overshoot, rise time, settling, and steady- state error. We calculate the step response of the system and out of which we calculate the performance criteria. The Iterations are run till the performance criteria minimize. The flowchart of the PSO is shown in Fig .4.

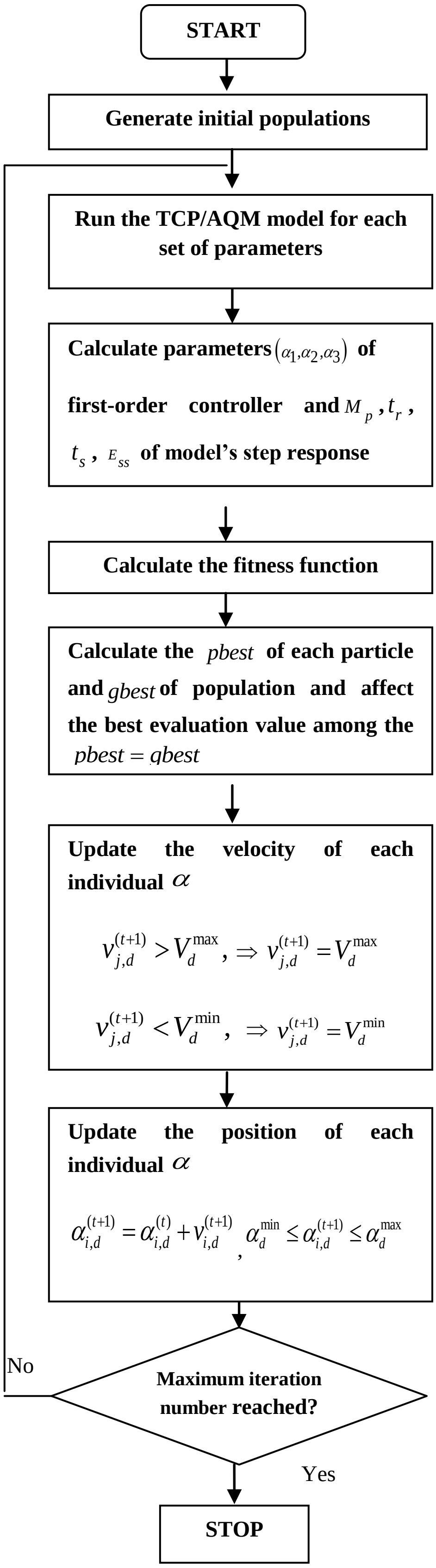


Fig 4: The flowchart of the PSO/first-order controller for TCP/AQM system

5. SIMULATIONS AND DISCUSSIONS

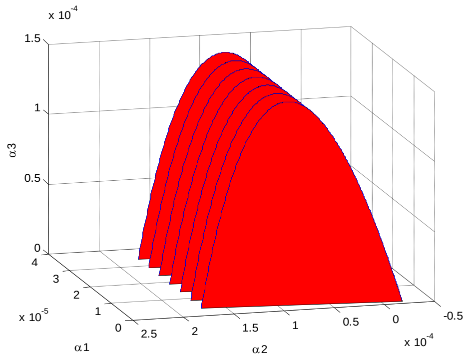
This section validates the effectiveness and performance by simulation, of a first-order controller using PSO algorithm, named PSO/first -order controller.

5.1 Simulations in Matlab

In the first simulation, we will conduct simulation by Matlab. Thus, we consider determining the stabilizing regions in the parameter space of a first order controller applied to the system given in equation (6). According to the network parameter $\mathrm{N} = 60, \mathrm{C} = 3750 \mathrm{packets} / \mathrm{s}$ and $R_0 = 0.25s$, it follows from (23) that

$$P(s) = \frac{117187.5}{(s+0.512\ )(s+0.4)} e^{-0.246s} \tag{23}$$

The admissible range of α_1 is $\alpha_1\,(-2,+\infty)$, obtained by applying the procedure given in section 3, Then fixing $\alpha_1 = 1e-5$ the interval, the stabilizing region in the plane of the remaining of the two parameters (α_2, α_3) derived from equations (16) and (17) is determined. The stabilizing regions in the plane $(\alpha_1, \alpha_2, \alpha_3)$ in fig 5.



.

Fig 5: Stabilizing regions in the plane $(\alpha_1, \alpha_2, \alpha_3)$

Fig6 represents the step response of the closed loop system with $(\alpha_1, \alpha_2, \alpha_3) = \{1e-5, 1e-4, 0.5e-4\}$

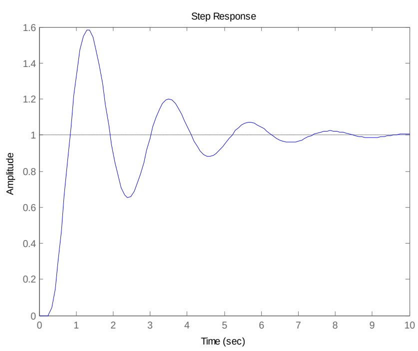


Fig 6: Step response of the TCP/AQM closed-loop system

It is seen that response of the closed-loop control system is stable, but it doesn't has good performances. Hence, in order to guarantee the good performance of the TCP/AQM system for closed loop, we applied PSO/ first-order controller, presented in section 4. According to the trials, the following PSO parameters are used to verify the performance of the PSO/first-order controller parameters.

- The lower and upper bounds of the three controller parameters are chosen of the stabilizing regions of the fig5 follows as $\left[\alpha_1^{\min} = -0.5e-5, \alpha_1^{\max} = 3e-5\right]$

  $\left[\alpha_2^{\min} = 0, \alpha_2^{\max} = 1.8e-4\right]$, $\left[\alpha_3^{\min} = 0, \alpha_3^{\max} = 1.2e-4\right]$
- population size= 50
- iteration =50
- acceleration constant $c_1 = c_2 = 2$
- $w_{\min} = 0.4$, $w_{\max} = 0.9$
- the limit of change in velocity [24]

$$V_{\alpha_1}^{\max} = \frac{\alpha_1^{\max}}{2}, V_{\alpha_2}^{\max} = \frac{\alpha_2^{\max}}{2}, \text{and} V_{\alpha_3}^{\max} = \frac{\alpha_3^{\max}}{2}$$

Using the procedure presented in the flowchart, we obtain three optimal parameters of the first-order controller

$$(\alpha_1, \alpha_2, \alpha_3) = \{2.4414\text{e-}5,\ 5.4999\text{e-}5, 1.9718\text{e-}5\}$$

Fig 7 represents the step response of the closed loop system with $(\alpha_1, \alpha_2, \alpha_3) = \{2.4414\text{e-}5,\ 5.4999\text{e-}5, 1.9718\text{e-}5\}$

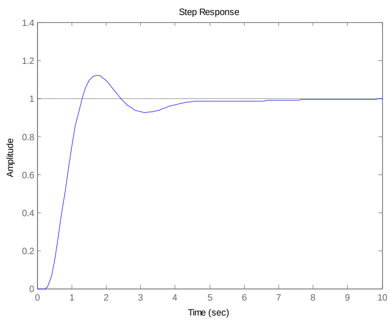


Fig 7: Step response of the TCP/AQM closed-loop system using of PSO/first-order controller

To show the effectiveness of the proposed approach, a comparison is made with the designed first order controller using PSO and the first order controller without optimization the performance of the two approaches are present in Table 1.

Table 1. Performance of the PSO/first-order controller

Performance Criteria	First-order controller	PSO/first-order controller
Overshoot $M_p(\%)$	58.7063	12.0897
Rise time $t_r(s)$	0.3986	0.3298
Settling time $t_s(s)$	8.0655	4.3947
Steady- state error E_{ss}	0	0

As can be seen in table1, the PSO/first-order controller allows to obtain better evaluation value, therefore, it can achieve good performance criterion (no overshoot, minimal rise time, Steady state error = 0).

5.2 Simulation in NS

In order to verify the effectiveness and performance of the proposed PSO first order controller by simulation we used the NS-2 simulator .The network topology is shown in fig8.

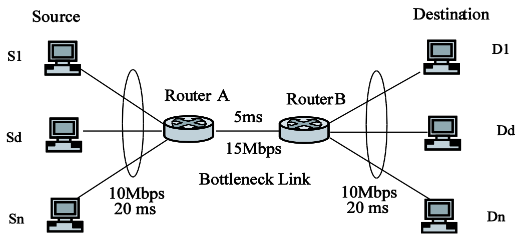


Fig 8: Simulation of network topology

We introduced 60 TCP flows and the simulation time is 80 s. $S_i(i=1,...,n)$ are TCP senders with average packet size 500 Bytes. S_d is a FTP sender which has 10 Mbps capacity and 20ms propagation delay, the traffic scenario. The only bottleneck Link lies between Router R_A and R_B, which has 15Mbps capacity and 5ms propagation delay. Router R_A uses the PSO First-order controller (or PI controller), others use the Drop Tail. The sampling is $160Hz$. The buffer has a maximum capacity of 800 packets and the desired queue length is 200 packets. The parameters of the PI controller defined in [3] are $a = 1.822e-5$ and $b = 1.816e-5$, the parameters of the first-order controller are chosen to the stabilizing region in fig.5 $(\alpha_1, \alpha_2, \alpha_3) = \{1e-5, 1e-4, 0.5e-4\}$ and the parameters optimal of the PSO/first-order controller determined in the section 4 are $(\alpha_1, \alpha_2, \alpha_3) = \{2.4414e\text{-}5,\ 5.4999e\text{-}5, 1.9718e\text{-}5\}$. We will use the network configuration presented in fig.8, we make a comparison between PSO/ first-order controller with first order controller and PI controller. The desired queue is fixed at 200 packets in both controllers, their respective instantaneous queue lengths are plotted in figs 9, 10 and 11, respectively. We noticed that the PI controller and first -order controller have taken a long time to regulates the queue to reference value compared the PSO/ first-order controller which quickly keeps the queue length.

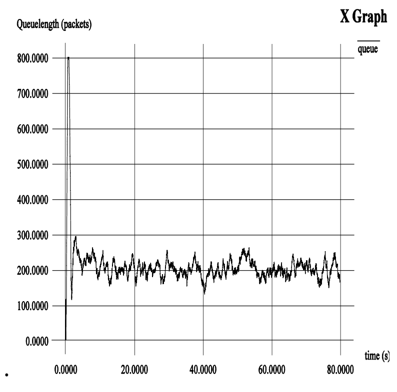


Fig 9: Instantaneous queue size, PSO/first-order controller

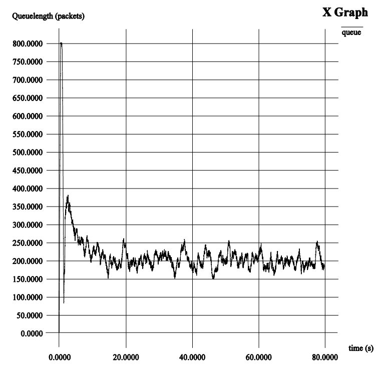


Fig 10: Instantaneous queue size, first-order controller

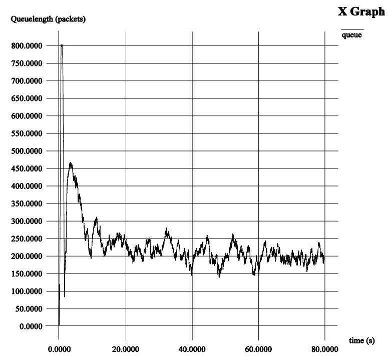


Fig 11: Instantaneous queue size, PI controller

6. CONCLUSION

This paper discusses the stability characteristic of TCP/AQM systems with time delay using the first-order controllers. First, The D-decomposition method and Lemma of Kharitonov is used for determining the stabilizing regions in the plane of $(\alpha_1, \alpha_2, \alpha_3)$. Second, a Particle Swarm Optimization (PSO) method is proposed for determining the optimal parameters of first-order controllers. The results show that the proposed controller can perform an efficient search for the optimal parameters. The simulation with NS-2 simulator shows that the proposed PSO/first-order controller has better performance than first-order controller and Hollot's PI control scheme.

7. REFERENCES

[1] Jacobson, V "Congestion avoidance and control" Proceeding of SIGCOMM'88, ACM, 1988.

[2] S. Floyd and V. Jacobson,"Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol. 1, pp 397-413, August 1993.

[3] C.V Hollot, V.Misra D. Towsley, W. Gong "Analysis and design of controllers for AQM routers supporting TCP flows". IEEE Trans on Automatic Control, vol.47, pp 945-959, June 2002.

[4] J. S. Sun, G.C.K.T.Ko, S.Chan and M. Zukerman, "PD-controller: A New Active Queue Management Scheme," In Proc. IEEE Global Telecommunications Conference GLOBECOM 2003), December 2003.

[5] Kim K B. Design of feedback controls supporting TCP based on the state-space approach. IEEE Trans. Automatic Control, 2006, 51(7): 1086-1099.

[6] Yang Ji-Wen, Gu Dan-Ying, Zhang Wei-Dong. An analytical design method of PID controller based on AQM/ARQ, Journal of Software, 2006, 17(9): 1989, 1995 (in Chinese).

[7] R. N. Tantaris, L. H. Keel, and S. P. Bhattacharyya, "Stabilization of continuous-time systems by first order controllers" Proceedings of the 10th Mediterranean Conference on Control and Automation, Lisbon, Portugal, July 9-12, 2002.

[8] K. Saadaoui and A. B. ¨ Ozg¨ uler," A new method for the computation of all stabilizing controllers of a given order ", International Journal of Control, vol. 78, pp. 14-28, 2005

[9] R.N.Tantaris, L.H. Keel and S. P. Bhattacharyya,"Stabilizationof discrete-time systems by first-order controllers", IEEE Trans. Automat. Cont., vol. 48, pp. 858-860, 2003.

[10] C .V. Hollot, V. Misra, D. Towsley, W. Gong "A Control Theoretic Analysis of RED" Proceedings of IEEE Infocom 2001.

[11] S. Kunniyur and R. Srikant, "Analysis and design of an Adaptive Virtual Queue (AVQ) algorithm for Active Queue Management", Proceedings of ACM SIGCOMM, pp. 123-134, August 2001

[12] Fan. Y., Ren. F. et Lin. C., "Design a pid controller for active queue management", In IEEE International Symposium on Computers and Communication (ISCC), volume 2, pages 985–990, 2003.

[13] Agrawal. D et Granelli. F, "Redesigning an active queue management system" In IEEE Globecom, volume 2, pages 702–706, December 2004.

[14] Misra. V., Gong. W., et Towsley. D., "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED". In ACM SIGCOMM, pages 151–160, August, 2000.

[15] Online Available: NS-2 Simulator, available at http://www.isi.edu/nsnam/ns/

[16] G. J. Silva, A. Datta, and S.P Bhattacharyya, "PI stabilization of first– order systems with time delay," Automatica, vol. 37, pp. 2025- 2031, 2001.

[17] Kharitonov. V. L. Niculescu S, Moreno J et Michiels W. "Static outputstabilization: Necessary conditions for multiple delay controllers", IEEE Trans Automat. Control, 2005.

[18] K. Saadaoui, S. Testouri and M. Benrejeb " Robust stabilizing first- order controllers for a class of time delay systems "ISA Transactions vol. 49, pp. 277-282, 2010.

[19] Gryazina EN, Polyak BT. Stability regions in the parameter space: D-decomposition revisited. Automatica 2006;42:13-26

[20] J.Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[21] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in Proc. IEEE Int. Conf. Evol. Comput., Anchorage, AK, May 1998, pp. 611–616.

[22] 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.

[23] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," IEEE Trans. Energy Conversion, vol. 19, pp. 384-391, June 2004.

[24] Kennedy, J. and Eberhart, R., Swarm Intelligence, Academic Press, 1st ed., San Diego, CA, 2001.

[25] Y. Shi and R. Eberhart. Empirical Study of Particle Swarm Optimization. In the IEEE Congres on Evolutionary Computation, pages 1945–1950, Washington, 1999.

[26] Zwe L G., « A Particle Swarm Optimization approach for optimum design of PID controller in AVR system » IEEE Transactions on Energy Conversion, vol.19, p. 384-391 june, 2004.