Design and simulation an optimal enhanced PI controller for congestion avoidance in TCP/AQM system

Yousra Abd Mohammed, Layla H. Abood, Nahida Naji Kadhim

1Communication Engineering Department, University of Technology, Baghdad, Iraq
2Department of Control and System Engineering, University of Technology, Baghdad, Iraq

ABSTRACT

In this paper, snake optimization algorithm (SOA) is used to find the optimal gains of an enhanced controller for controlling congestion problem in computer networks. M-file and Simulink platform is adopted to evaluate the response of the active queue management (AQM) system, a comparison with two classical controllers is done, all tuned gains of controllers are obtained using SOA method and the fitness function chose to monitor the system performance is the integral time absolute error (ITAE). Transient analysis and robust analysis is used to show the proposed controller performance, two robustness tests are applied to the AQM system, one is done by varying the size of queue value in different period and the other test is done by changing the number of transmission control protocol (TCP) sessions with a value of ± 20% from its original value. The simulation results reflect a stable and robust behavior and best performance is appeared clearly to achieve the desired queue size without any noise or any transmission problems.

Keywords:
AQM
Computer network
Congestion control
NLPI controller
SOA

1. INTRODUCTION

Avoiding high rates of packet loss in the internet is an important issue, when a packet is losing before it received from its destination point; all the data it has sent through transmission path are dissipated. In excessive states, this case will cause a congestion collapse. Congestion control has become as an urgent matter in computer and communication networks. Congestion has harmful effect on network efficiency which reflecting to noticeable packet loss, poor utilization, high delay rate, few throughputs [1], [2].

Different algorithms is used to solve the congestion problem but the best one is the active queue management (AQM) scheme, AQM algorithm maintains congestion by previously detection of incipient congestion and giving feedback signals to end-hosts to permit them minimizing their transmission rate before the router’s buffer exceeded [3]. In [4] an inclusive study is demonstrated on the AQM algorithm techniques that suggested and adopted to modify the performance efficiently, the AQM algorithms are classified based on length of the queue or its delay or both of them. Proportional-integral-derivative (PID) fuzzy-neural controller was presented; PID controller gains were tuned based on particle swarm optimization (PSO) tuning algorithm to enhance the fuzzy controller behavior [5]. An AQM scheme is controlled using fuzzy controller as a (mixing-fuzzy-PID) controller to provie best congestion solution and few delay periods, high utilization and few data drop [6]. A fuzzy proportional integral (FPI) controller with genetic tunning was suggested as an AQM for internet router [7]. An enhanced version of discrete linear quadratic optimal controller is adapted to track the desired queue size in AQM scheme and genetic algorithm (GA) are used to fix the complexity of finding the weighting matrices $Q$ and $R$ [8].

This is an open access article under the CC BY-SA license.
The adaptive suggested method was introduced as a newly adopted method, named PHAQM, by using the Hebb neural network for adjusting the variables of the model predictive control (MPC)-based AQM system [9]. Different methods that were used to maintain the congestion problem at the router in the network [10]–[13]. The algorithm that used for congestion problem is the drop tail (DT) method [14]. This method adopted the first-in first-out (FIFO) technique. DT method are used without thresholds, then if the capacity is maximized at the router buffer, the packets that arrived are directly. Maintaining the problem of congestion in AQM system based on a smart and unique snake optimization algorithm (SOA) tuning method for finding gains of the controller to fulfill the stability and robustness by fixing any problem mdropped [12], [15]. In [16] a fuzzy controlling method was used to enhance the modify AQM system behavior without need for a precise model.

In this paper an enhanced proportional integral (PI) controller is proposed for solve congestion problem that occurs in the communication networks during transmission data. The rest of the paper is: part 2 explains the modelling of AQM system. Part 3 indicateds the controller suggested scheme. Part 4, demonstrates the SOA tuning algorithm, part 5 discuss the results obtained then part 6 presents the paper conclusions.

2. MODELING OF TCP/AQM

Transmission control protocol (TCP) operation was modeled and studied using differential equations, together with a fluid-flow-based technique for ignoring TCP timeout and an investigation of stochastic relationships [17], [18]. It is assumed that each homogenous TCP flow connected to a certain bottleneck topology has an identical delay, as shown in Figure 1 [19]. The dynamic model is defined by the nonlinear deferential equations [20], [21] shown below, where the parameters of (1), (2) and (3) are shown in Table 1.

\[
W'(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} P(t - R(t)) \tag{1}
\]

\[
q'(t) = \frac{W(t)}{R(t)} N(t) - C \tag{2}
\]

\[
R = \frac{q}{c} + T_p \tag{3}
\]

![Bottleneck scenario](image)

Figure 1. Bottleneck scenario [19]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W)</td>
<td>TCP window size (packets) is positive, (W \in [0, W_c]), (W_c): (max. size of window)</td>
</tr>
<tr>
<td>(W(t))</td>
<td>Derivative time of (W(t))</td>
</tr>
<tr>
<td>(q)</td>
<td>Queue size (packet), (q \in [0, q_c]), (q_c): (buffer capacity)</td>
</tr>
<tr>
<td>(q(t))</td>
<td>(q(t)) derivative time</td>
</tr>
<tr>
<td>(t)</td>
<td>Time (sec)</td>
</tr>
<tr>
<td>(R)</td>
<td>Time of round trip (sec)</td>
</tr>
<tr>
<td>(N)</td>
<td>Number of TCP sessions</td>
</tr>
<tr>
<td>(C)</td>
<td>Capacity of the link (packet/sec)</td>
</tr>
<tr>
<td>(p)</td>
<td>Packet probability (mark/drop) ([0, 1])</td>
</tr>
<tr>
<td>(T_p)</td>
<td>Propagation delay</td>
</tr>
</tbody>
</table>

Table 1. Parameters of AQM model [21]
The nonlinear relations of the AQM model were linearized [8], [22] as shown in (4). Figure 2 shows the linearized AQM control system.

\[
p(s) = \left(\frac{e^{-sR_0}}{2N}\right) \left(\left(s + \frac{2N}{R_0}\right)\left(s + \frac{1}{R_0}\right)\right)
\]

(4)

In this study the values of the parameter used are: \(R_0 = 0.253\) (sec), \((C = 15\) Mbps = 3750 packets/sec) and \((N = 60\) TCP sources) [4]. Using the preceding parameters in (4), the full TCP/AQM system is obtained, and the transfer function of the model is as (5):

\[
p(s) = \left(117187.5e^{-0.253s}\right)/(s^2 + 4.5245s + 1.9759)
\]

(5)

3. PROPOSED ENHANCED CONTROLLER

The classical PID controller is considered as a flexible and traditional type of controllers, it’s used for enhancing system response. Recently, many studies are used these types with a numerous modifications such as using intelligent techniques such as combines with sliding mode controller or using with any type of neural network [23], [24], or change its structures to improve the response of the system as in [25], or use the fractional calculus method by adding an integral or differential fractiona values or together to enhance system behavior [26], the values are (\(\lambda\) for the integral part and \(\mu\) for derivative part), based on this controller gains become five variables. In this study a nonlinear formula is adopted to improve system performance and led the system to its efficient response[27], [28] as indicated in Figure 3, it is an enhanced combination between nonlinear gain \(K_n(e)\) and the original PI controller gains, this nonlinear gain is a function of system error also, as shown in (6).

\[
K_n(e) = Tanh(e) = \frac{\exp(k_0e) - \exp(-k_0e)}{\exp(k_0e) + \exp(-k_0e)}
\]

(6)

\[
e = \begin{cases} 
  e & |e| \leq e_{max} \\
  e_{max} \cdot sign(e) & |e| > e_{max} 
\end{cases}
\]

(7)

\(K_p, K_i, \) and \(K_0\) are three parameters that will tuned for achieve an optimal response for this controller.

4. SNAKE OPTIMIZATION ALGORITHM

The SOA was adopted by Hashim and Hussien [29]. It explains the snakes mating way, it describes how the snakes struggling for finding their suitable associate if there is a good place such as cold weather and a sufficient food, it’s randomly starting with population generation, and an update for its position is done in two phases: exploration and exploitation. They start with considering that the male’s numbers and the female’s numbers are equal when it starts the updating process. This action remains till complete chosen number of the iteration (7), the first phase is called exploration, it describes the state when there food is not sufficient and the thier search is done in random style. Quantity of food is calculated using (8) [30].
\[ Q = 0.5 \exp \frac{t - T}{T} \]  

And the equations of this phase are:

\[ X^t_{i,m} = X^t_{\text{rand,m}} \pm C_2 \times A_m \times (X_{\text{max}} - X_{\text{min}}) \times \text{rand} + X_{\text{min}} \]  

\[ X^t_{i,f} = X^t_{\text{rand,f}} \pm C_2 \times A_f \times (X_{\text{max}} - X_{\text{min}}) \times \text{rand} + X_{\text{min}} \]

Where \( X^t_{i,m}, X^t_{i,f} \) represent the \( i \)th places of both males and females. \( X^t_{\text{rand,m}}, X^t_{\text{rand,f}} \) represent the places that taken randomly by the two types chosen population (males and females). \( C_2 \) represents a known variable (\( C_2 = 0.5 \)), \( A_m \) is the male ability while \( A_f \) is the female ability to reach to the food place and it is calculated by (11), (12).

\[ A_m = \exp \frac{-f^t_{\text{food}}}{f^t_{i,m}} \]  

\[ A_f = \exp \frac{-f^t_{\text{food}}}{f^t_{i,f}} \]  

\( F \) explains the fitness function magnitude of males and females. When they find their food, the exploitation phase is begin, then their places is updated due to their environment temperature (TEMP) and its found as indicated:

\[ TEMP = \exp \frac{-t}{T} \]  

When the the weather becomes hot and the temperature exceeds the threshold level, the places of males and females are updated as shown:

\[ X^t_{i,f,m} = X^t_{\text{food}} \pm C_3 \times TEMP \times \text{rand} \times (X_{\text{food}} - X^t_{i,f,m}) \]  

\( X^t_{\text{food}} \) is regarded as the optimal place and \( C_3 \) equal to 2. Otherwise, the population will replaces between each other randomly in two modes either fighting or mating and reach the places as explained in (15)-(18):

\[ X^t_{i,m} = X^t_{i,m} \pm C_3 \times FM \times \text{rand} \times (X_{\text{best,f}} - X^t_{i,m}) \]  

\[ X^t_{i,f} = X^t_{i,f} \pm C_3 \times FF \times \text{rand} \times (X_{\text{best,m}} - X^t_{i,f}) \]  

\[ X^t_{i,m} = X^t_{i,m} \pm C_3 \times M_m \times \text{rand} \times (Q \times X^t_{i,f} - X^t_{i,m}) \]  

\[ X^t_{i,f} = X^t_{i,f} \pm C_3 \times M_f \times \text{rand} \times (Q \times X^t_{i,m} - X^t_{i,f}) \]

In (19), (20) is specific for fighting state and \[21, \ 22\] is specific for mating state and the \( FF, FM \) are representing the ability of fighting, \( M_m \) and \( M_f \) are considered as a mating ability for males’ and female’s as listed:

\[ FM = \exp \frac{-f^t_{\text{best,f}}}{f^t_{i}} \]  

\[ FF = \exp \frac{-f^t_{\text{best,m}}}{f^t_{i}} \]  

\[ M_m = \exp \frac{-f^t_{i,m}}{f^t_{i,m}} \]  

\[ M_f = \exp \frac{-f^t_{i,f}}{f^t_{i,f}} \]

The selection of worst male and female is happen when egg hatch then switch between them.

\[ X_{\text{worst,m}} = X_{\text{min}} + \text{rand} \times (X_{\text{max}} - X_{\text{min}}) \]  

\[ X_{\text{worst,f}} = X_{\text{min}} + \text{rand} \times (X_{\text{max}} - X_{\text{min}}) \]
For monitor system performance in reaching the optimal response one of the cost function relations will be used to monitor the queue size until reaches to its optimal level and achieve stability for the system, the integral time absolute error (ITAE) [31] was selected for checking system performance, optimal gains of suggested controller are tuned by SOA tuning method with try to minimizing the magnitude of the fitness function ITAE used [32], and it is defined in (25) by checking the system error value continuously.

\[ ITAE = \int_0^\infty t |e| \, dt \]  

Figure 4 indicates the AQM system based on SOA and, the flowchart of SOA is explained in Figure 5.

**Figure 4. AQM system based on SOA**

**Figure 5. Flowchart of SOA**
5. SIMULATION RESULTS

The simulation results for the AQM system using the suggested controller is presented in this section using Matlab/Simulink to analyze system performance based on suggested controller and the SOA then compared it with the two conventional controller (optimal PI and Classical PI controllers). The first controller is tuned using SOA and the second one is manually tuned, the SOA variables initial values are indicated in Table 2. The efficient response of the system based on nonlinear proposed controller is appeared clearly when compared with the two controllers (optimal PI, classical PI) as shown in Figure 6 and the three controllers gains is shown in Table 3. Then this comparison is analyzed based on the results of response analysis obtained for the three and it is shown Table 4.

Table 2. SOA parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population number</td>
<td>50</td>
</tr>
<tr>
<td>Maximum iteration number</td>
<td>30</td>
</tr>
<tr>
<td>Variables</td>
<td>3</td>
</tr>
</tbody>
</table>

![Figure 6. The AQM system response based on all controllers](image)

Table 3. The three controller’s gains

<table>
<thead>
<tr>
<th>Controller</th>
<th>(K_p)</th>
<th>(K_i)</th>
<th>(K_o)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical PI</td>
<td>0.0000042</td>
<td>0.00009</td>
<td>-</td>
</tr>
<tr>
<td>Optimal PI</td>
<td>0.0000083</td>
<td>0.000074</td>
<td>-</td>
</tr>
<tr>
<td>Nonlinear PI</td>
<td>0.0091</td>
<td>0.00451</td>
<td>0.00285</td>
</tr>
</tbody>
</table>

Table 4. Response analysis results for the three controllers

<table>
<thead>
<tr>
<th>Controller</th>
<th>Maximum overshoot (Mp %)</th>
<th>Peak time (Tp)</th>
<th>Rise time (tr)</th>
<th>Settling time (ts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal PI</td>
<td>18.33</td>
<td>7.05</td>
<td>2.95</td>
<td>4.4</td>
</tr>
<tr>
<td>Optimal PI</td>
<td>7.5</td>
<td>7.15</td>
<td>3.32</td>
<td>4.8</td>
</tr>
<tr>
<td>Adaptive PI</td>
<td>0</td>
<td>3.75</td>
<td>1.975</td>
<td>3.4</td>
</tr>
</tbody>
</table>

As indicated in response analysis results in Table 4 the normal PI controller and the optimal PI controller is similar in there evaluation parameters with small different in their values (355 packets and this led to overshoot with 18.33% for the normal PI and 322.5 with 7.5% overshoot) while the nonlinear PI controller is different from these two controllers by its fast response with stable behavior without any overshoot or noise during simulated time this is due to the nonlinear function and the tuned \(K_o\) variable value within the tanh hyperbolic function that is used with the PI control which is regulate and stabilize controller behavior, \(K_o\) value is tuned using SOA algorithm. The fast settling time is appeared on nonlinear PI controller that makes the system efficiently tracking the desired response, then the analysis of robustness to detect the stability of the nonlinear controller, two test is done the first one is to monitor its ability in tracking the desired system response in a stable manner by changing the value of queue size each 50 sec, the proposed controller solves this change in queue size value regularly and give a stable response in spite of changing desired queue size applied as shown in Figure 7, while the second test is done by changing the number of TCP sessions from its original value \(N = 60\) to a \(\pm 20\%\) from its original value (+20% equal 72 and -20% equal 48), it can be seen that the system response is affected when changing system TCP sessions as indicated in Figure 8, when it is increased to 72 it will suffer from slow response as compared with the original system while when it is decreased to 48 it will suffer from overshoot with value of 8.5 % and needs 10 sec to return to its stable response.
6. CONCLUSION

In this paper an intelligent snake optimization tuning algorithm is suggested to tune an enhanced PI controller with a hyperbolic function to maintain AQM system congestion problem. The controller achieve a smooth and stable performance for monitoring the system response desired value. A comparison analysis is utilized with two classical controllers (optimal PI, normal PI) to show the stable and robust response of the proposed controller based on transient response analysis (peak time, settling time, rise time and overshoot), then to test system ability to solve problems or changes which may happen during communication a robustness tests are applied to the system. The results showed an efficient response in saving the desired value of the queue size and reach to a stable and robust performance in maintaining the issue of congestion that occur in AQM system.

REFERENCES


**BIOGRAPHIES OF AUTHORS**

**Yousra Abd Mohammed** is a lecturer at the Communication Engineering Department, University of Technology-Baghdad, Iraq since 2005. She received her B.Sc. in Electronic and Communication Engineering from Technology University/Baghdad, Iraq in 1992 and her M.Sc. degree in Computer Engineering from Technology University/Baghdad in 2004. Her research interests include Control Systems, Encryption and Decryption Algorithms. She can be contacted at email: yousra.a.mohammed@uotechnology.edu.iq.

**Layla H. Abood** received her B.Eng. and M.Sc. degrees in Electronic and Communication Engineering from the University of Technology- Baghdad. She is currently an Academic staff member in the Department of Control and System Engineering, University of Technology, Baghdad, Iraq. Her research interests are Artificial Intelligent, Control Systems, System Modeling, Optimization Techniques, IoT, Image processing, Robotics, Microcontrollers, FPGA and Embedded systems. She can be contacted at email: 60066@uotechnology.edu.iq.

**Nahida Naji Kadhim** received B.Sc. and M.Sc. degrees in Control Engineering and Mechatronics Engineering from Control and Systems Engineering Department in 1993 and 2005 respectively, University of Technology-Baghdad, Iraq. She is currently an academic staff member in the Department of Control and Systems engineering, University of Technology, Baghdad, Iraq. Her researcher interests are Control System, Robotic System, Identification System and Artificial Inelegant. She can be contacted at email: nahida.n.kadhim@uotechnology.edu.iq.