Interval type-2 and type-1 Fuzzy Logic Controllers for congestion avoidance in internet routers

Hanan M Kadhim¹, Ahmed A Oglah²

¹Control and Systems Engineering Department, University of Technology-Iraq   Email: 61276@student.uotechnology.edu.iq
²Control and Systems Engineering Department, University of Technology- Iraq   Email: 60008@uotechnology.edu.iq

Abstract. Active queue management (AQM) scheme is an important research area in congestion control. The AQM mechanism attains the congestion controller by regularize the length of queue from evaluation or dropping the packets at the moderate routers. In this paper the interval type-2 and type-1 fuzzy logic controller are designed to desist the congestion in the networks. Several evolutionary optimization algorithms are used to select the parameters of two controllers as particle swarm optimization (PSO), social spider optimization (SSO) and ant colony Optimization (ACO). The simulation of linearized model of TCP/AQM are introduced in MATLAB (R2018a). Results of the PID like fuzzy type-1controller is compared with PID like interval type-2 fuzzy controller (IT2-FLC) based on optimization algorithms in each controller and shows that IT2-FLC with SSO has been given the best result.

1. Introduction

The internet is the basic portion for running name of applications much as Web, transmission, etc. Nevertheless, because of unexpected trouble and numerate of users who hit to Net for specified clip, the congestion could be occur. This is cause longer delays in collection transmitting and often leads the queue size in the middle routers' device to overflow. One of the effective means to discover the congestion is titled an active queue management (AQM) scheme, it gives advance note of actual internet status by dropping (or scoring) the inflowing packets before router queues get full. Newly, number of (AQM) mechanism were introduced to increment the utilization of the network by regulating queues at the bottleneck links in the networks, like (random early detection (RED), adaptive RED (A-RED) and proportional-integral (PI) controller.

The earliest AQM scheme method is random early detection which proposed to manage congestion control issues. RED was proposed in 1993 by Floyd and Jacobson to solve the problems arising from the tail-drop method [1]. Same other AQM methods, the thought of RED is to estimate the congestion before buffer overflows by measurement the average queue length.

Congestion control in the network is a time varying, nonlinear and complicated problem. So, it needs an adaptive, robust, efficient AQM mechanism to gain satisfying performance. Fuzzy logic control (FLC) is rattling superior in treatment with uncertainty in the issue of congestion control by using heuristic knowledge, because it's ambitious to obtain a meticulous mathematical model by using classical methods [2].

A number of (AQMs) using a fuzzy logic are brought onward. For example, Ashtiani and Mohsen proposed an AQM theme based on fuzzy logic techniques as a (hybrid-fuzzy-PID) controller to offer efficient crowding control with less delay time, high utilization and low packets loss [3]. In 2012, (PID-fuzzy-neural controller) was introduced to be an (AQM) to the internet's routers. PID controller
parameters was tuned by using PSO as an optimization method to improve the performance of fuzzy controller [4].

A fuzzy proportional-integral (FPI) is introduced as AQM scheme in net routers to enhance the response of (PI controller) to avoid crowding in networks of computer. As an optimization method the Genetic Algorithm (GA) was used for tuning the FPI parameters [5]. Amjad J. et al., in 2014 presented two controllers’ model predictive control and PID. The two controller is based on the dynamic nonlinear model of TCP/AQM. The system based PID controller is less robust than MP controller against a variation in some parameters [6]. In 2019, a novel of control the congestion method for (TCP/AQM) scheme is proposed to overcome a tracking difficulty by using adaptive back stepping, funnel control and fuzzy logic systems. The transient and steady state performances of tracking error could meet the designing requirements [7].

In this paper, a PID-like type-1 fuzzy logic controller and PID-like IT2- fuzzy logic controller is organized for controlling the linearized model of TCP/AQM. These controllers modify congestion suppress in computer network and dedicate the optimized pressurized signaling characteristics. To adjust the gains controller, a three type of optimization methods are used with every controller which are a particle swarm optimization (PSO), ant colony optimization (ACO) and Spider social optimization (SSO). The two controllers adjust the queue window size to confirm minimum rise time, settling time, peak time and overshoot.

This article is organized as follows: Section 2, describes the linearized (TCP/AQM) model. Section 3, the two controllers: PID-likeT1- FLC and PID-like IT2-FLC is designed. Section 4, a type of Optimizations with Evolutionary Algorithms are presents, Section 5 presents the results of simulation for two controllers. Section 6 explains the conclusions of this article.

2. TCP/AQM model
A dynamic model of transmission control protocol (TCP) behavior was developed by using fluid-flow and the analysis of stochastic differential equation with ignoring the TCP timeout mechanism. This model can be described by the following nonlinear differential equations [8], [9]:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}P(t-R(t))$$  \hspace{1cm} (1)

$$\dot{q}(t) = \frac{W(t)}{R(t)N(t) - C}$$  \hspace{1cm} (2)

Where W is predicted TCP window size (packets), \(\dot{W}(t)\) denotes the time-derivative of \(W(t)\), q is supposed queue size (packet), \(\dot{q}(t)\) denotes the time derivative of \(q(t)\), \(t\) refer to time in (sec), R represent round trip time (sec), N is a number of TCP sessions, C is the link capacity in (packet/sec) and P is refer to a packet mark/drop probability. The marking probability takes value only in [0, 1]. Also, the queue length q and windows size W is positive and bounded values; i.e., \(q \in [0, q]\) and \(W \in [0, W]\) where \(q\) denote buffer capacity and \(W\) denote maximum window size. The nonlinear differential equations of the AQM model could be linearized around \((W_o, q_o, P_o)\) which represent an operating point for generating the following linearized model 9]:

$$P(s) = \frac{\delta q(s)}{\delta p(s)} = \frac{\frac{C^2}{2N} e^{-sR_o}}{(s + \frac{2N}{R_o e^2C})(s + \frac{1}{R_o})}$$  \hspace{1cm} (3)

Figure .1 shows the block diagram of linearized AQM control system where \(\delta p(s)\) is the loss probability, \(\delta W(s)\) is the window size, \(\delta q\) is the queue length and the \(C(s)\) is a controller’s transfer function.
As a case study we take the parameters where Ro=0.253 sec, N=60 TCP sources, (C=15) Mbps which equivalent to (3750 packets/sec) as shown in figure 2, where applying the above parameters in equation (3) gets the overall TCP/AQM system the transfer function of the model will be:

\[ P(s) = \frac{117187 \cdot 5e^{-0.253s}}{s^2 + 4 \cdot 45245s + 1 \cdot 9759} \]  

\[ (4) \]

**Figure 2.** The network topology of case study

3. **Controllers design**

Two controllers are designed as follow:

3.1. **PID-like T1-FLC**

The PID-like fuzzy logic controller is select to achieve the optimal controlled sign property by reducing the overshoot, rise time, settling and peak time. The equation of PID controller is [10]:

\[ u(t) = K_p e(t) + K_d \dot{e}(t) + K_i \int e(t) dt \]  

\[ (5) \]

Fuzzy logic controllers (FLCs), as intelligent systems, can be employed to model the making decision of human and human being experiences and making behaviors [11]. In FLC set of linguistic rules or relational expressions are used to express the input - output relationships. Fundamentally, the FLC consists of four parts are: (fuzzifier, defuzzifier, inference engine and a rule base). the input data are often crisp in many applications of fuzzy control, so a fuzzification is essential to convert the input crisp
data into group of linguistic value which are needed in inference engine. Figure 3 shows the block which diagram of PID-FLC.

Figure 3. Block diagram of PID-FLC for AQM

The membership functions of the input and output are seven triangles and Gaussian shaped and the input–output range are equal [-1 1] which tuned by $K_o$ in order to be with range [0 1] of dropping probability factor range. Center of gravity method is selected to be a defuzzification mechanism [10]. Figure 4 shows the MFs for inputs and output. The linguistic variables of membership functions of FLC and the fuzzy rule base are shown in Table 1 and Table 2 respectively.

Table 1. Linguistics description of MFs

| Item | Linguistics description | Linguistics abbreviation |
|------|-------------------------|--------------------------|
| 1    | NB                      | Negative Big             |
| 2    | NM                      | Negative Medium          |
| 3    | NS                      | Negative Small           |
| 4    | Z                       | Zero                     |
| 5    | PS                      | Positive Small           |
| 6    | PM                      | Positive Medium          |
| 7    | PB                      | Positive Big             |

Figure 4. The MFs of FLC
Table 2. Fuzzy rule base

| e   | NB | NM | NS | Z  | PS | PM | PB |
|-----|----|----|----|----|----|----|----|
| NB  | NB | NB | NB | NB | NM | NS | Z  |
| NM  | NB | NB | NB | NM | NS | Z  | PS |
| NS  | NB | NB | NM | NS | Z  | PS | PM |
| Z   | NB | NM | NS | Z  | PS | PM | PB |
| PS  | NM | NS | Z  | PS | PM | PB | PB |
| PM  | NS | Z  | PS | PM | PB | PB | PB |
| PB  | Z  | PS | PM | PB | PB | PB | PB |

3.1.1. Controller design.
The type-1 fuzzy logic controller is designed by Matlab (R 2018a) using a computer with Intel core i7 @ 2.80 GHz CPU, 4GB of RAM and 64-bit Windows 7 is shown in figure 5.

3.2. PID-like IT2-FLC
Despite having a name that holds the concept of uncertainty, T1 Fuzzy sets showed limitations in the ability to model and minimize the effect of uncertainties. That is because T1 fuzzy set is certain in the meaning of that its MF degrees are crisp values. In the other hand, the type-2 fuzzy sets were described by membership functions that are themselves fuzzy which attracted more attention. Interval type-2 (IT2) FSs is a particular case of type two Fuzzy sets, are widely utilized for their low cost of computation [12].

The membership of an (IT2 FS) is an interval which is unlike the (T1 FS) whose membership for each element is a number. The IT2 FS is bounded from below and above by two T1 FSs that named upper membership function and lower membership function. The footprint of uncertainty (FOU) is the area between upper and lower MF. IT2 FSs are effective when it is hard to define an accurate MF, or in molding the different opinions from disparate individuals. The MFs can be structured using optimization algorithms [13]. The schematic diagram of an (IT2 FLS) is correspondent to its T1 twin, the difference being that at least only one of the FSs in the rule base is an IT2 FS. Because the outputs of the inference engine are (IT2 FSs), a type-reducer is required for transforming them into a T1 FS before defuzzification can be achieved.

The controller is two input single output (MISO) with Mamdani type of fuzzy system. The two-input is the error of queue length and the change in this error and one output is the probability of packet mark (P). The input MFs are seven triangles formed with [-1 1]. Figure 6 is shown the membership function of input. The linguistic variables of membership functions of IT2-FLC and the fuzzy rule base are shown in Table 1 and Table 2 respectively.
3.2.1. Controller design:

Figure 7 shows the structure of PID like IT2-FLC.

4. Optimization with evolutionary algorithms

The EA is different from the trial and error methodology to search out the best value for the parameters of controller. The EA uses mechanisms impressed by biological evolution, like proliferation, recombination and mutation. The individuals in a population are represent a candidate solutions to the optimization problem, and the quality of the solutions is determine by fitness function. Evolution of the population then happens after the repeated application of the above operators.

In this paper, three types of optimization algorithms are used, particle swarm optimization (PSO), Ant Colony Optimization (ACO) and Social Spider Optimization (SSO).

4.1. Particle swarm optimization (PSO)

Particle swarm optimization algorithm (PSO) is a stochastic optimization method based on swarm, it was proposed by Kennedy and Eberhart in 1995. The PSO simulates animal’s social activity, like herds, insects, fish and birds. The swarms represent a collaborative way to gain food, and each member keeps dynamical the search space according to the learning experiences of its own and the additional rest members [14].

In the PSO, the potentiality solutions, named particles, fly inside the problem area by following the present best particles. The main PSO parameters are swarm size, velocity components, number of iterations and acceleration coefficients. In addition, PSO effected by velocity clamping, inertia weight and velocity constriction. Figure 8 shows the flow chart of particle swarm optimization algorithm.
4.2. Ant Colony Optimization (ACO)

Ant colony optimization (ACO) is the one amongst the recent algorithms for optimization. Realistic ant colonies are the inspiring seed of ACO algorithms. In addition, ACO is inspired by the ants’ hunting for food. This method is depending on the indirect communicating among ants by using of the chemical pheromone trails, which support these ants to choose the shortest path between food sources and their nest [15]. Figure 9 shows the implementation of the ACO [16].

![Flow chart of (PSO)](image1)

**Figure 8. Flow chart of (PSO) [14]**

![Flow chart of (ACO)](image2)

**Figure 9. Flow chart of (ACO) [16]**

4.3. Social Spider Optimization (SSO)

The social spider optimization (SSO) formula is a population based on algorithm which suggested by Cuvas et. Al, in 2013. There are two major elements of a social spider colony, the social members and communal web. There are two of social members, males and females. The proportion of male spiders approximately 30% while the proportion of female spiders almost 70% of the whole colony members. Male spiders are consisting of pair of groups, dominate and non-dominant male spiders where, the dominant male spiders possess bestead fitness features than a non-dominant. On the other hand, the female spider presents a charisma or dislike to other spiders and that correspond to their shaking dependent on the distance and weight of subscribers. To produce offspring, a dominant male mate with one or more females in a specified range and the mating operation allows the members to exchange the information [17].
5. Linearized TCP/AQM model simulation results

Consider the TCP/AQM with network parameters as set in the section 2 and the reference input which represents queue size has a rectangular form and changes every 50 seconds as shown in equation (6). The simulation is done firstly without controller as shown figure 10.

\[
q_{\text{ref}} = \begin{cases} 
300 & 0 < t < 50; \\
200 & 50 < t < 100; \\
400 & 100 < t < 150; \\
200 & 150 < t < 200; 
\end{cases} \quad (6)
\]

![Figure 10. System response without controller](image)

From the figure above we found that the response without controller is disabled to follow the queue size of the desirable value, and the system goes to the continued oscillation which exceeding the highest buffer size. In order to remove this oscillation and get best performance a PID like FLC (T1 AND IT2) are applied with using evolutionary optimization algorithms and the results changed according to the type of optimization algorithm. The result of using PID like (T1-FLC) with tuning the parameters by particle swarm optimization algorithm (PSO) shows in Figure 11. The parameters of particle swarm optimization (PSO) algorithm is chosen as shown in Table 3.

| Title                                      | Value  |
|--------------------------------------------|--------|
| Inertia weight                             | 0.1    |
| Inertia weight Damping Ratio                | 0.99   |
| Personal Learning Coefficient (c1)         | 1.5    |
| Global Learning Coefficient (c2)           | 2.0    |
| Maximum Number of Iterations               | 100    |
| Population Size (Swarm Size)               | 50     |
It worth mentioning that the fitness function use in all optimization algorithms used in this paper is the integral time absolute error (ITAE)

\[
ITAE = \int t |(e)| \, dt
\]

Where

\( e : q_{\text{ref}} - q_{\text{out}} \)

Figure 11. System response with PID like T1-FLC Based PSO

Figure 12 shows the system response of using PID like (T1-FLC) with tuning the parameters by Ant Colony Optimization (ACO). While the result of using PID like (T1-FLC) with tuning the parameters by Social Spider Optimization is shows in figure 13.

Figure 12. System response with PID like T1-FLC based ACO

Figure 13. System response with PID like T1-FLC based SSO

The comparison among the three optimization algorithms (PSO, ACO, SSO) controllers results in figure 14. It is show clearly ACO is the best one in rise time, overshoot and settling time.
For enhanced the response of the system, the PID like IT2-FLC is used where, the result of using PID like (IT2-FLC) with tuning the parameters by Particle Swarm Optimization algorithm (PSO) shows in Figure 15.

While the results of using PID like (IT2-FLC) based on Ant Colony Optimization and Social Spider Optimization SSO are shown in figure 16, figure 17 respectively.
The comparison among the three optimization algorithms (PSO, ACO, SSO) controllers results in figure 1. It is shown clearly SSO is the best one in rise time, overshoot and settling time.

Figure 1. Comparison among the three optimization algorithms (PSO, ACO, SSO) controllers results in Figure 1. It is shown clearly SSO is the best one in rise time, overshoot and settling time.

The steady state characteristics comparison for all controllers with optimization algorithms including in Table 4, and the comparison of PID gains parameters for all algorithms including in Table 5.

The table 4 shows that Ant Colony Optimization (ACO) algorithm in T1-FLC is outperform the rest of the algorithms with minimum rise time and settling time. While the Social Spider Optimization (SSO) algorithm in IT2-FLC is outperform the rest of the algorithms with minimum rise time and settling time.
Table 4. Steady state response comparison

| Title                        | Rise Time (Sec) | Settling Time (Sec) | Overshoot (%) | Peak Time (Sec) |
|------------------------------|-----------------|---------------------|---------------|-----------------|
| T1-FLC Based PSO             | 4.046           | 10.23               | -             | -               |
| T1-FLC Based ACO             | 2.081           | 6.546               | 0.505         | 4.5694         |
| T1-FLC Based SSO             | 3.041           | 6.812               | -             | -               |
| IT2-FLC Based PSO            | 1.758           | 9.673               | 4.737         | 4.353          |
| IT2-FLC Based ACO            | 1.846           | 9.743               | 0.505         | 5.915          |
| IT2-FLC Based SSO            | 1.366           | 3.763               | -             | -               |

Table 5. Comparison of PID gains parameters for all algorithms

| Title                        | Kp              | Ki              | Kd              | Ko              |
|------------------------------|-----------------|-----------------|-----------------|-----------------|
| T1-FLC Based PSO             | 0.0037          | 0.0058          | 0.0058          | 0.0027          |
| T1-FLC Based ACO             | 0.0070173       | 0.0039262       | 0.0023746       | 0.0066034       |
| T1-FLC Based SSO             | 0.007896        | 0.003110        | 0.004163        | 0.005019        |
| IT2-FLC Based PSO            | 0.006585        | 0.001661        | 0.006845        | 0.006845        |
| IT2-FLC Based ACO            | 0.0043009       | 0.0015991       | 0.0016189       | 0.0029711       |
| IT2-FLC Based SSO            | 0.00387         | 0.00210         | 0.00250         | 0.00425         |

So, comparison is made between T1FLC based ACO and IT2FLC based SSO algorithms in figure 19 and it shows that system response based IT2-FLC is a better than the response based on T1-FLC with less rise time and settling time.

Figure 19. comparison between T1-FLC and IT2-FLC
6. Conclusion
From the design and the simulation results, it can be concluded that:

1. The designed Fuzzy-PID controller can deal with congestion problem with a good tracking performance about the desired queue size with high link utilization and faster system response observed as compared with routers.
2. The evolutionary optimization algorithms helped to optimally select the best fuzzy-PID parameters.
3. When we made comparison in a system response among the three optimization algorithms, we found that ACO is the best algorithm for tuned parameters in T1-FLC, while the SSO is the best algorithm for tuned parameters in IT2-FLC with minimum rise time, settling time and maximum overshoot.
4. When we compared the two controllers, the results show that the system response in IT2-FLC is perform better than the T1-FLC with 34% percentage of enhancement in rise time.
5. The modeling and linearization of window and queue dynamics of the TCP/AQM model about an operating point to gain insight for the purpose of feedback control in design and analysis of AQM schemes.

References
[1] S. Floyd and V. Jacobson, 1993 Random early detection gateway for congestion avoidance, *IEEE /ACM Transactions on Networking* 1 (4).
[2] C. Chrysostom, A. Pitsillides and Y.A. Sekercioglu, 2009 Fuzzy explicit marking: A unified congestion controller for Best-Effort and Diff-Serv networks, *Computer Networks, Elsevier*, 53.
[3] H. Ashtiani, H. Moradi and Mohsen, 2012 Active Queue Management in TCP Networks Based on Fuzzy-PID Controller, *Journal of Applied computer science & Mathematics* 6 (12), pp. 9-14.
[4] M. Z. Al-Faiz and S. A. Sadeq, 2012 Particle Swarm Optimization Based Fuzzy-Neural Like PID Controller for TCP/AQM Router, *Intelligent Control and Automation* 3 (1), pp.71-77.
[5] M. Z. Al-Faiz and A. M. Mahmood 2011 Fuzzy-Genetic Controller for Congestion Avoidance in Computer Networks, *IJCCCE* 11 (2), pp. 22-30.
[6] A. J. Humaid, H.M. Hasan and F. A. Raheem, 2014 Development of Model Predictive Controller for Congestion Control Problem, *IJCCCE* 14 (3), pp. 42-51.
[7] K. Wang, Y. Liu, X. Liu, Y. Jing and S. Zhang, 2019 Adaptive fuzzy funnel congestion control for TCP/AQM network, *Elsevier, ISA Transactions*.
[8] V. Misra, W.-B. Gong, and D. Towsley, 2000 Fluid-Based Analysis of a Network of AQM routers Supporting TCP Flows with an Application to RED, in *Proc. ACM SIGCOMM*, pp. 151-160.
[9] C.V. Hollot, V. Misra, D. Towsley and W. Gong, A Control Theoretic Analysis of RED, in *Conf. Anc. IEEE Int. Conf. Computer and Communications Societies*, pp. 1510 – 1519.
[10] L. Reznik, Fuzzy Controllers, Australia, Newnes, 1997, Ch. 4.
[11] C. Wang, Bo Li, K. Soharby, and Y. Peng, 2003 AFRED: An Adaptive Fuzzy-based Control Algorithm for Active Queue Management, *IEEE LCN*.
[12] M. Nie and W. W. Tan, 2008 Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in *Proc. IEEE Int’l Conf. on Fuzzy Systems, Hong Kong*, pp. 1425–1432.
[13] J. Zeng, 2005 Type-2 fuzzy hidden Markov models and their application to phoneme classification, in *IEEE Region 10 Postgraduate Student Paper Contest*.
[14] D. Wang, D. Tan and L. Liu, 2018 Particle swarm optimization algorithm: an overview, *Soft Comput, Springer*.
[15] A. Akhtar, 2019 Evolution of Ant Colony Optimization Algorithm — A Brief Literature Review, *Neural and Evolutionary Computing*.
[16] K. Khurshid, S. Irateza and A. A. Khan, 2010 Application of Ant Colony Optimization based algorithm in MIMO Detection,” *IEEE Congress on Evolutionary Computation, CEC*.
[17] E. Cuevas, M. Cienfuegos, D. Zaldívar and M. Pérez-Cisneros, 2013 A swarm optimization Algorithm inspired in the behavior of the social-spider, *Expert Systems with Applications* **40** (16), pp. 6374-6384.