

Fuzzy Like Pid Controller Based on Sso Design for Congestion Avoidance in Internet Router

Mohammed I. Berbek¹, Ahmed A. Oglah²

^{1,2} Control & System Engineering Dept, University of Technology, Baghdad, Iraq.

¹ cse.19.08@grad.uotechnology.edu.iq, ² 60008@uotechnology.edu.iq

Abstract— today The topic of contention is the quality of computer network service. Efficient data handling is important. Each router offers a buffer space where packets can be held before being sent until they are processed. The theory of packet congestion control is TCP (Transmission Control Protocol) .(AQM) is a mechanism proposed for gateways to improve congestion management. AQM (active queue management) is a crucial to minimize packet loss in TCP/IP networks and improving network efficiency. This paper offers a hybrid intelligent (PID) with type1 fuzzy logic controller which is designed to decrease network congestion. As a router (AQM), social spider optimization (SSO) is used to tune the control parameters used to reduce queue size error. The designed controller based on SSO algorithm provides a good tracking performance for AQM desired value. The observational findings indicated a high-performance increase in the proposed process. In MATLAB(R2020a), a simulation of a linearized TCP/AQM model is presented. NS2 is used to perform verification for the fuzzy logic controllers for the nonlinear TCP/AQM model under a more practical simulation and to analyze and examine network behavior in different scenarios.

Index Terms—AQM controller, congestion control, fuzzy logic control, PID Controller; SSO.

I. INTRODUCTION

One of the most widely used technology is computer networking that can be extended to various areas of life. This network reliance and heavy utilization result in heavy data traffic, contributing to internet traffic jams. traffic jams Occur when overall resource demand increases the resource's usable range, more round-trip time, which creates network output distortion. The delay in delivering packets and packets loss in computer networks is due to the issue of congestion in TCP/AQM routers, and the transmission control protocol and active queue management work together to address this problem. TCP provides secure data transfer and designed to handle congestion after its occurrence. AQM predicts congestion and tries to resolve the problem before it occurs. It's been proposed that effective (AQM) give congestion warnings to senders as soon as possible, and senders may reduce transfer speeds before Overflow queue and packet loss prevent. AQM was an active research subject [1]-[5]. Several AQM techniques have been proposed for congestion avoidance. In Internet routers, AQM's first known method, the RED algorithm (Random Early Detection), was designed and implemented to reduce the problem of synchronization flow and measure traffic load using queue length measurement. Nonetheless, several studies and theoretical analyzes showed that RED's output Due to its ad hoc architecture, it responds to parameter settings and traffic volume. Consequently, some new amended schemes been proposed in the literature, including ARED, FRED and SRED. Nevertheless, these studies seem unable to sustain system efficiency in a variety of operational

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

environments, like number of connections, link capability and delay in propagation. Regulation of congestion of the network is a complex problem that varies in time and nonlinear. So, to achieve satisfying efficiency, it requires an adaptive, stable, efficient AQM mechanism. A PID controller was built with input TCP system delayed for a time to maintain a consistent queue length and low packet drop. An efficient mechanism is proposed to select control PID-controller gains based on hybrid GA-BF searching method [1]. Qin-Zhou, sn, 2011 In coping with the unpredictable case, AQM (Active Queue Management) controller with self-configuration is suggested. The Lyapunov Direct approach evaluates the predominance and stability of fuzzy theory. The algorithm is more rational, reliable and resilient compared to other conventional approaches[2]. Qiao & Lei, 2011, FAPIDNN, Suggested new algorithm for AQM , the Fuzzy controller Calculates Learning rate automatically in accordance with current network status and PID controller of the neural network calculates the likelihood of dropping the packet dependent on the rate of learning given by fuzzy controller. Simulation results indicate that the FAPIDNN algorithm is the most accurate to the queue, convergence speed and time delay of the PID controller provided [3]. M. Z. Al- Faiz introduced a The new (AQM) PID Fuzzy-Controller for Routers on the Internet. To increase the fuzzy controller's efficiency, the PID parameters were tuned using PSO as an optimization technique. [4]. Fan, X. L., Du, F. F., & Xie, Z. H. (2014), presented A new AQM system based on a PID controller with a fuzzy neuron algorithm that is adaptive and self-learning input rate implies an Fuzzy Neuron PID Adaptive Control algorithm Input rate dependent (IRAFNPID) [5]. Feng et al., 2014, They presented a minimal modification to RED called TRED stands for three-section random early detection based on (nonlinear RED), in which the likelihood function of packet falling is split into three parts to discriminate among Small, Medium and heavy loads in order to obtain a delay and low throughput Elevated traffic loads[6]. Azadegan et al., 2015. For (TCP) network congestion control, a new state feedback controller of proportional derivative type is proposed [7]. Kim Quoc et al., 2015, A paradigm integrating the fuzzy controller with the neural network (FNN) has been proposed. This increases the operating performance of the processes of active queue management at network nodes[8]. Sadek et al., 2017, Show the design, based on state input, of a PID controller. Using linear matrix inequality (LMI) technology, Our PID parameters are calculated and added to active queue management (AQM) on the router to prevent congestion [9]. In 2017, An Intelligent Fuzzy Logic Management approach has been used to be uniquely suited for complex and non-linear systems where an empirical model is difficult to access[10]. Karam, 2018, To control the length of the queue, round trip time, and PSO Based Loss of packets, Several fuzzy hybrid controllers are suggested, including controller PID like FLC Based on Particle Swarm Optimization (PSO) ,(PD) like FLC with traditional IPSO Based, and PID tuned by Fuzzy Logic with PSO Based.[11]. In 2018, A technique called ANB-AQM adaptive neural (AQM) is suggested,To contend with the nonlinearity of network networks, this strategy the parameters are fine-tuned self-learning-based. This helps to estimate the potential value likelihood of packet drop by observing the prior experience of queue length error [12]. Gomez et al., 2019, They suggested using information from the Explicit Congestion Notice (ECN) to boost AQM algorithms by incorporating Machine Learning techniques. In order to predict congestion, this smart approach Uses an artificial neural network and a reinforcement-based tuner of AQM parameters. The assessment results indicate that the approach will improve the efficiency of AQM deployed[13]. Based on a modeling principle of multi-agent systems, a new congestion control The neural networks framework is proposed to explore a class of multiple (TCP/AQM) network finite time

Received 23/3/2021; Accepted 11/6/2021

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

consensus. Wang et al., 2020 [14]. One of the conventional PID control approaches of fuzzy logic for wireless networks and congestion is discussed in this research. Diverse configurations can be managed. Oudah et al., 2021[15].

In this paper, The PID like fuzzy controller for the control of the TCP/AQM linear model is organized. the (SSO) is used to set the gains with proposed controller as approaches to optimisation. The best results were obtained through PID like fuzzy controller with SSO optimization. Furthermore, In a more realistic simulation, NS2 is used to validate the nonlinear TCP/AQM model and to simulate the fuzzy logic controllers and test network behavior in various scenarios.

The following parts of the paper are arranged according to: Section two describes the (TCP/AQM) model's linearization. Section three introduces the design PID-like- FLC . In Section four, A type of Evolutionary Algorithm Optimization is present. Section five, shows the simulation results PID-like- FLC in matlab and the results of NS2. Finally, Section six provides a conclusion.

II. CONGESTION CONTROL WITH TCP/AQM

The mathematical model of TCP flow includes a nonlinear differential equation and its association with active queue management see [16] for information on linearization leading to the following:

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

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

where W shows the average TCP window size (packets), q length of expected queue represents (packet), R round trip time (seconds), C link capacity (packets/sec), delay in transmission is indicated by TP (s), N indicates the load factor (TCP sessions), p Likelihood of packet marking or loss.

In (1), it was defined a multiplicative decrease formulation TCP strategy, When a sender avoided congestion, In any R , the congestive window increases in one data segment, If there is congestion, the window size is reduced to half of its initial size.

The equation (2) explains the changes in buffer space on the router; the number of packets queued in the router changes when the packet sent to a router is subtracted from the sent out a packet.

The estimated queue length q and the TCP window size W are positive and small. Also, the packet likelihood (mark/drop) p values are only in $[0,1]$. To rearrange, the Laplace transformation of (1):-

$$P_{tcp}(s) = \frac{W(s)}{P(s)} = \frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0 C}} \quad (3)$$

$$P_{queue}(s) = \frac{q(s)}{w(s)} = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \quad (4)$$

Therefore, the plant transfer function can be referred to as:

$$P(S) = P_{tcp(s)} P_{queue(s)} e^{-sR_0} \tag{5}$$

And it can be expressed as:

$$P(S) = \frac{\delta q(s)}{\delta p(s)} = \frac{\frac{c^2}{2N} e^{-sR_0}}{\left(s + \frac{2N}{R_0^2 C}\right) \left(s + \frac{1}{R_0}\right)} \tag{6}$$

A linear AQM control block diagram is shown in Fig.1. In this graph, $P_{tcp(s)}$ represents the transfer function from loss probability $\delta p(t)$ to window size $\delta W(t)$, $P_{queue(s)}$ represents the transfer function from $\delta W(t)$ to queue length $\delta q(t)$, e^{-sR_0} delay term, and $C(s)$ system controller.

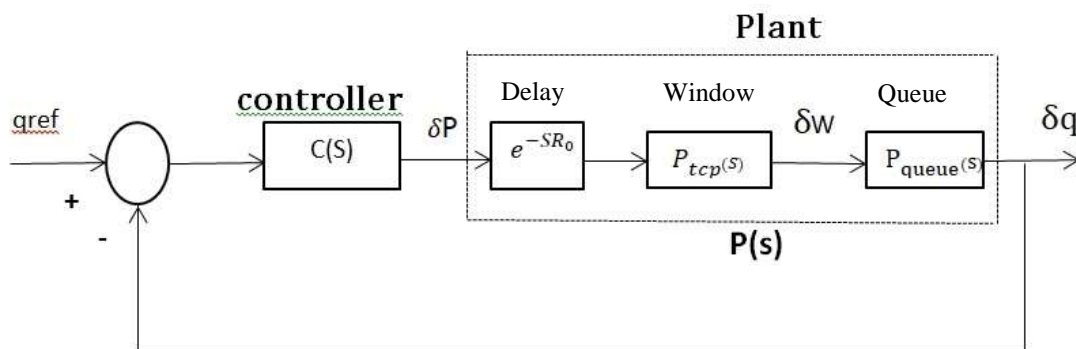


FIG. 1. ALINEARIZED AQM BLOCK DIAGRAM AS A FEEDBACK CONTROL.

After the half-time sampling of R_0 , the plant transfer function is planned specifically, and to take into account a case study with $N = 60$, $R_0 = 0.253$ sec and $C = 3750$ packets/sec. where the parameters are applied in formula (6), The obtained transfer function is in formula (7). The network topology is as shown in Fig.2.

$$P(S) = \frac{117187 \cdot 5 e^{-0.253S}}{S^2 + 4 \cdot 4524S + 1 \cdot 9759} \tag{7}$$

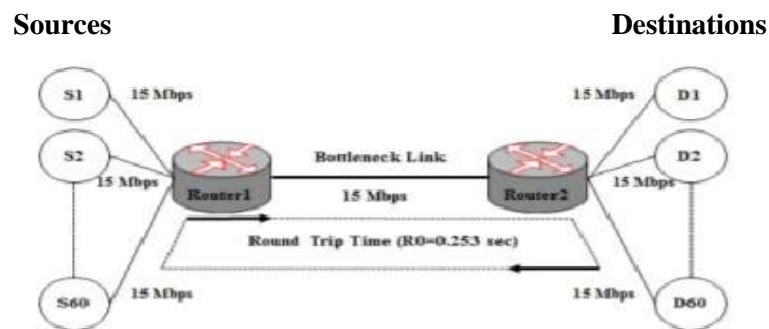


FIG. 2. THE TOPOLOGY OF THE CASE STUDY NETWORK.

III. CONTROLLER DESIGN

The key aim of the synthesis of strong control systems is to find a control law which maintains in the permitted signal a system output and error signals. The controller is designed as follows:

PID-like-FLC

The classic PID controller is a popular controller for almost every system. This sort of controller calculates the control signal according to the following formula. [17]:

$$U(t) = K_p(t) + K_d \dot{e}(t) + K_i \int e(t) dt \quad (8)$$

$$U(t) = r(t) - y(t) \quad (9)$$

Although it is often the goal of classical control theory uses empirical approaches to explain system behavior and the controller's subsequent configuration, fuzzy systems are suitable for modeling ambiguous information, such as a technological mechanism or an actual (human) controller. Fuzzy system provides a means to calculate the intermediate values between crisp values that are completely true and equal (1) or totally false and equal (0). These the range of values from 0 to 1, thus, a fuzzy set allows gray shades between black and white to be calculated (or true and false). The objective of fuzzy logic control is to imitate the human mind's mechanism while performing control acts. In its quest to imitate the behavior of human control actions, a type-1 FLC, It consists of four key parts: fuzzifier, rules, inference engine, and De-fuzzifier. The fuzzy PID type controller contains two components: the PID controller and the FLC, as seen in *Fig.3*. The proposed fuzzy PID controller is a controller to regulate the AQM size by way of confirmation minimum settling time, overshoot and rise time.

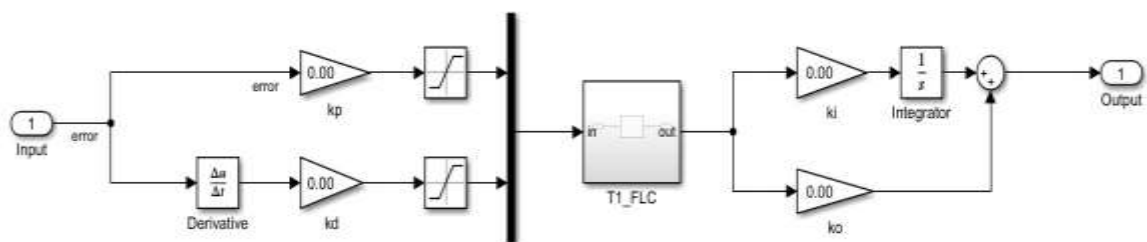


FIG. 3. THE STRUCTURE OF A PID-LIKE-FUZZY LOGIC CONTROLLER.

The membership functions input and output form like seven triangles and Gaussian in shape, which are distributed across their universes of discourse for input and output membership functions. The fuzzy system is normalized which means that the effective universes of discourse are all given by (-1,1). Tuned by K_o to fall inside the [0 1] range of the drop probability factor. As a De-fuzzification Method, the center of gravity is chosen [17]. The MFs for inputs and outputs are shown in *Fig.4*. Table I shows the fuzzy rule base of the FLC.

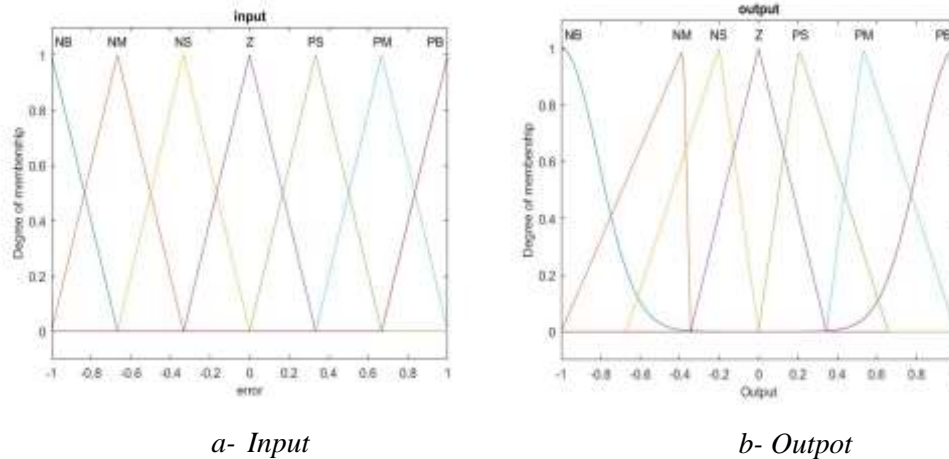


FIG. 4. THE FLC'S MEMBERSHIP FUNCTIONS.

TABLE I. FUZZY RULE'S BASE

e \ 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

Table I is built first on the system's characteristics, then fine-tuned through repeated trials, and this table reflects abstract information about how to control the process provided the error and its derivative as input.

where PB Positive Big, PM Positive Medium, PS Positive Small, Z Zero, NS Negative Small, NM Negative Medium and NB Negative Big are the abbreviations in the Table. In addition, Use this number of MFs to obtain an accurate and high quality Fuzzy system, thus obtaining an accurate response. Created using Matlab (Mamdani Fuzzy Inference Systems). The rules are also included through the AND operation.

IV. EVOLUTIONARY ALGORITHMS FOR OPTIMIZATION

The EA differs from the trial-and-error method of determining the correct value for the controller parameters. An optimization algorithm is a process or algorithm used to calculate the maximum and minimum of a function has Certain restrictions. [18].

SSO Technique

Erik Cuevas et al. presented the Social Spider Optimization (SSO) in 2013, which is a demographic algorithm that simulates the social spider's cooperative behavior. SSO takes two quest agents into account: men and women. Based on their gender, Each individual receives different evolutionary operators to Simulation of different cooperative conduct in a

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

colony. This type of categorisation individuals makes it possible to the reduction of crucial flaws present in a variety of SI methods, such as inappropriate discovery exploitation equilibrium and Convergence prematurely. In contrast to other common, genetic algorithm (GA), Particle Swarm Optimization (PSO) algorithms, Artificial Bee Colony (ABC), and Social Network Optimization (SNO), Evades particulate accumulation at the best positions; prevention of crucial defects, such as a minimal balance between exploration and exploitation or a premature convergence to suboptimal solutions. SSO was widely modified and implemented in many fields of engineering [19].

Social spider algorithm formulates the search area for the problems of optimization as a web spider in super dimensions. Each web position corresponds to a feasible approach to the problem of optimization and all viable problem solutions have the Web positions corresponding. The web also acts as a means of disseminating the vibrations that spiders make. Every spider has a particular position on the Web, and the solution's consistency (or fitness) is determined by its objective function, which reflects the spider's ability to locate a food source in a specific location. Spiders have complete freedom of movement on the internet. They cannot, however, leave the site because off-web positions are insufficient solutions to the optimization problem. When a spider travels to a new spot, it sends out a pulse that can be felt across the web. Each vibration stores information about a single spider, and other spiders can learn how to get the vibration. [20]. Fig.5 is a complete flow chart of the algorithm.

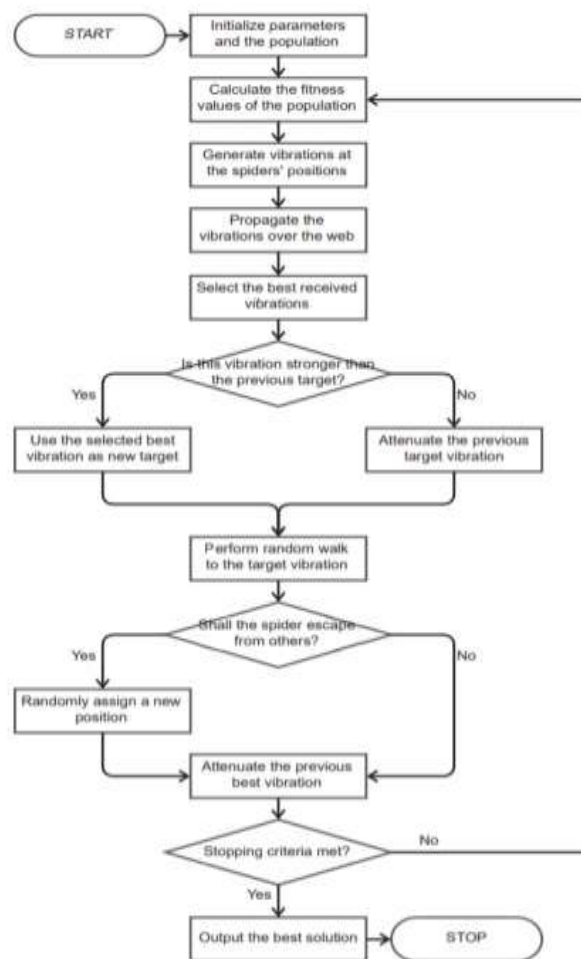


FIG. 5. SOCIAL SPIDER ALGORITHM FLOW CHART.

Received 23/3/2021; Accepted 11/6/2021

V. RESULTS AND DISCUSSION

A. Linearized TCP/AQM Model Simulation Results

Network topology simulation results parameters shown in Fig.2 are as follows: The bottleneck link capacity of 15 Mbps is between router1 and router2, with a propagation delay of 0.2 second, $N = 60$, and a packet size of 500 bytes, and the desired input queue size has rectangular shape per 50 seconds inconstancy. The AQM Router 1's maximum queue length is 800 packets. Router1 configures the technique of AQM and drop Tail is used in other gateways.

$$q_{ref} = \begin{cases} 300 & 0 < t < 50 \\ 200 & 50 < t < 100 \\ 400 & 100 < t < 150 \\ 200 & 150 < t < 200 \end{cases}$$

Fig.6 depicts the simulation without the use of a controller.

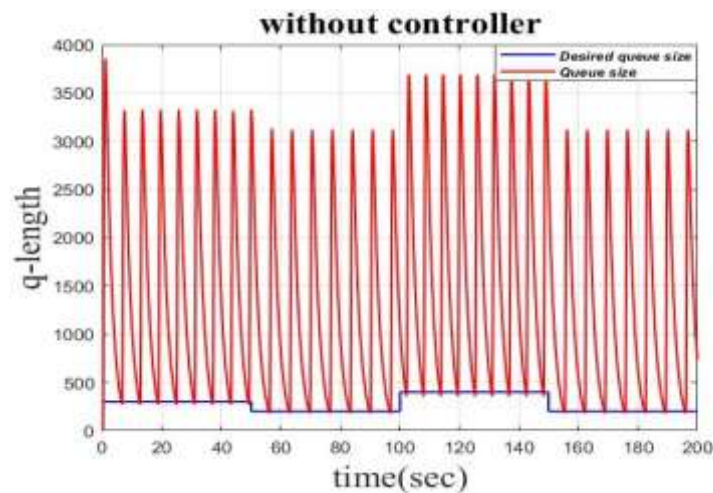


FIG. 6. SYSTEM RESPONSE WITHOUT CONTROLLER.

The system without the controller is seen to be unable to track queue length to desired level through queue length where the device enters a long-term oscillation due to high congestion that exceeds the Maximum size buffer. To reduce this continuous Oscillating and to improve tracking efficiency, a PID like FLC is implemented using evolutionary optimization algorithm (SSO) . The consequence of the use (PID, like FLC) of the social optimization algorithm (SSO) parameters is shown in Fig.7.

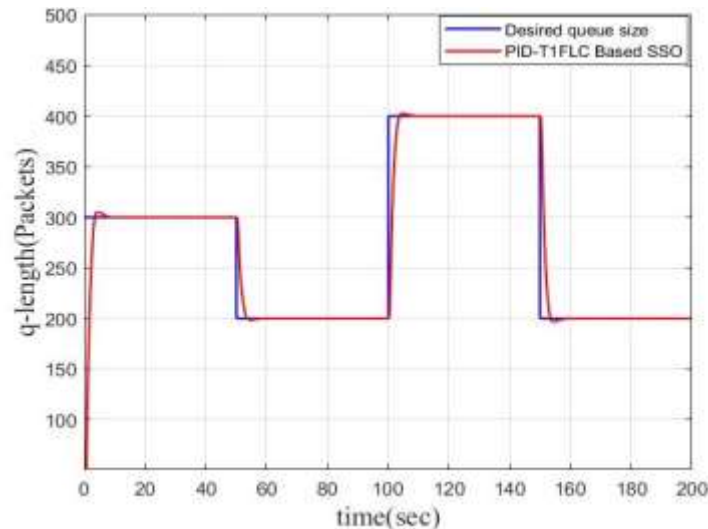


FIG. 7. BASED ON SSO,THE SYSTEM RESPONDS WITH A PID LIKE FLC.

The SSO is a suitable tuning optimization tool (PID like FLC) Parameters, the SSO parameters are: spiders number 50, Lower female percentage 0.65, High female percentage 0.9, iterations number 50, and the integral time absolute error is the fitness function (ITAE).

$$ITAE = \int t |(e)| dt \quad (10)$$

where

$$e = q_{ref} - q_{ou}$$

From Fig.7., it is shown clearly, that the controller suggested based on SSO is able to track the desired value of AQM in router. And achieve the best rise time, settling time, and overshoot.

The computational results of FLC-PID with SSO optimization algorithm are shown in the Table II. And PID gain parameters for SSO algorithm included in Table III.

TABLE II. FLC WITH SSO OPTIMIZATION ALGORITHM NUMERICAL RESULTS

Controller	Rise time (sec)	Settling time (sec)	Overshoot %
SSO_FLC	2	7.3	1.666

Raise time and settling time are significant factors, Because a lower value means that an optimum queue level is being maintained and congestion is being reduced more rapidly.

TABLE III. GAINS OF SSO ALGORITHM'S FLC-PID CONTROLLER

Title	Kp	Ki	Kd	Ko
SSO_FLC	0.004018	0.008345	0.002814	0.007536

This section shows that the system without the controller is unable to track desired queue length, and when the PID like FLC controller is applied to the system, the response is good and highly efficient, and the system is able to track desired queue length value. This indicates that the efficiency and ability of the controller in the stability of the system and

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

tracking AQM required value. Thus it can be reduce the dropping packets rate and improve the networks performance in general.

When comparing the results with the search [21], where it was robust PID controller that was proposed in order to design robust AQM algorithm. This controller is designed based on H_{∞} technique with Ant Colony Optimization (ACO) method is used to tune the parameters of PID controller to achieve the required robustness of the network, as shown in the Table IV.

TABLE IV. TCP/AQM SYSTEM RESPONSE PERFORMANCE OF SSO FLC, AND H_{∞} ACO PID

Controller	Rise time(sec)	Settling time (sec)	Overshoot%
SSO_FLC	2	7.3	1.666
H_{∞} _ACOPID[21]	0.4807	0.9220	-

B. Results of Nonlinear TCP/AQM Model Simulation in NS2

The simulation non-linear was done via the Network Simulator 2 [NS-2] software version (NS— 2.35) for testing the built-in (PID-like-FLC) performances in a more complicated traffic scenario. Table V displays AQM-controller parameter settings (PID-like-FLC).

TABLE V. AQM-CONTROLLER PARAMETER

Title	Kp	Ki	Kd	Ko
FLC Based on SSO	0.00119	0.01	0.022	0.001

Fig.8. illustrates the behavior of the suggested controllers to keep the queue length near to the target 200 packet value depending on the network parameters in Section (A). The mean and standard deviation values of FLC are Mean= 196.6450 and standard deviation= 38.9276.

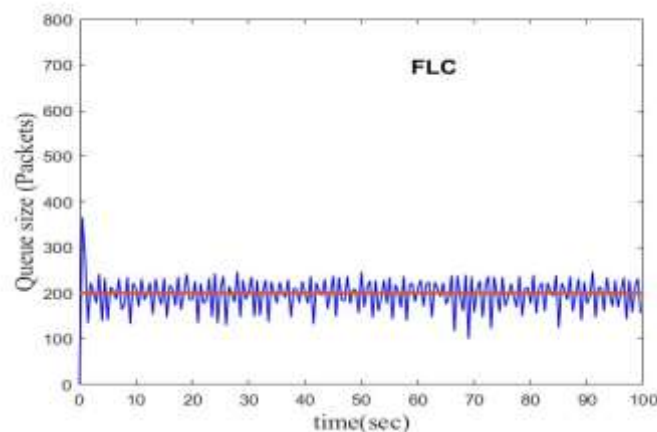


FIG. 8. (C=15Mbps- N=60- Tp=5ms) THE RESPONCE OF THE SYSTEM

Table VI shows a comparison with Search [21] when (C=15Mbps - N=60 - Tp=5ms - and Queue size=200 packets)

TABLE VI. PERFORMANCE COMPARISSON BETWEEN SSO_FLC AND H_{∞} _ACOPID

controller	Mean	Standard deviation
SSO_FLC	196.6450	38.9276
H_{∞} _ACOPID[21]	195.6	33.09

C. Robustness and Efficiency Test

Many studies have been performed to test the efficiency and robustness of the FLC by modifying the different networks. Values of parameters, multiple values of the size of the queue, and additional disruptions are as follows:

C.1 Change the TCP Session Number (N)

FLC stability and robustness were evaluated in FTP Flows. The link capacity (C) and the propagation time (T_p) were both set at 5 ms and 5 Mbps respectively, while the TCP sessions (N) were chosen to increase their original values by 10% and 66% and 100% respectively (66, 100 and 120). The FLC system response was shown in Fig.9 These statistics showed that as the number of sessions increased, the FLC's ability to track the queue size decreased at the target level, and that in order to boost the controllers' response with N greater than 100 increase, The parameters of the controller needed to be Re-adjusted. Table VII showed results mean and standard deviation of the proposed controllers' length of the queue.

TABLE VII. MEAN AND DEVIATION VALUES IN DIFFERENT NUMBER OF TCP SESSION(N)		
TCP Sources	Mean FLC	Standard Deviation FLC
66	211.755	24.5087
100	225.405	19.7927
120	230.77	19.5012

Fig.9. shows the effects (mean) and (standard deviation) of the controllers' length of the queue. It can be deduced that the FLC controller could produce a more suitable solution with a change in standard deviation equal to (20.5%) and an increase in the mean value by (8.3%) with an increase in N to (66, 100, 120), respectively.

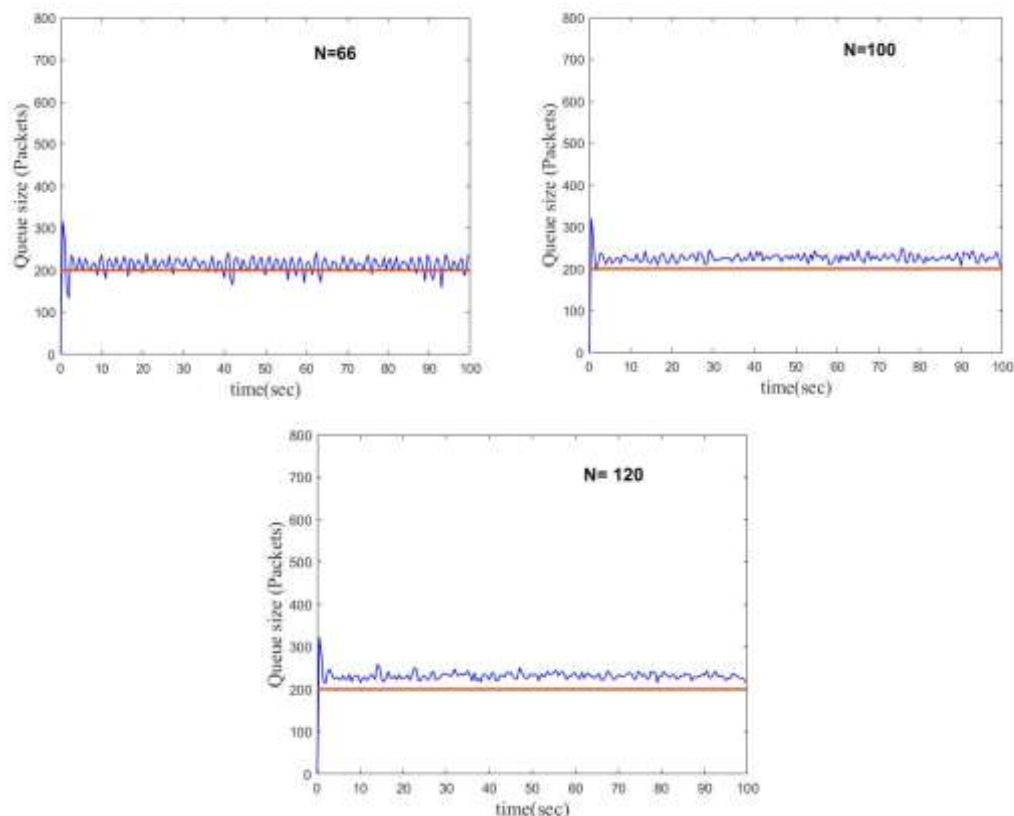


FIG. 9. SYSTEM RESPONSE USING FLC FOR DIFFERENT TCP SESSIONS.

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

C.2 Sudden Differences in TCP Flows

The robustness of The controller suggested was tested in a more complicated scenario, where long-lived TCP (N) flow numbers have been modified abruptly. The number of TCP flows used is 60, 120, 80 and 60 at times 0-30, 30-50, 50-70 and 100, respectively, as seen in Fig.10. The illustration figure shows that the FLC can hold the queue length near the desired value. The mean and standard deviation for the length of an FLC queue are: mean = 206.3050, standard deviation = 38.2348.

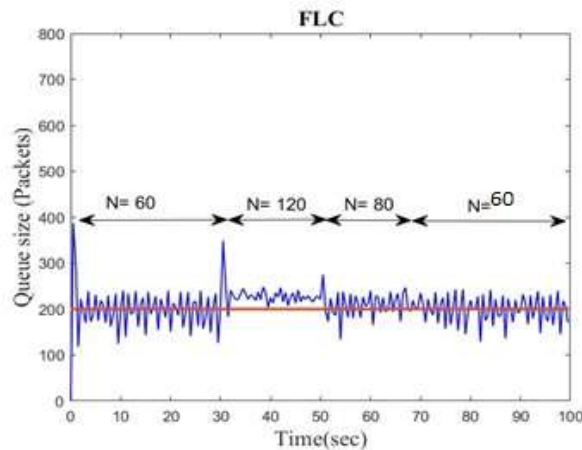
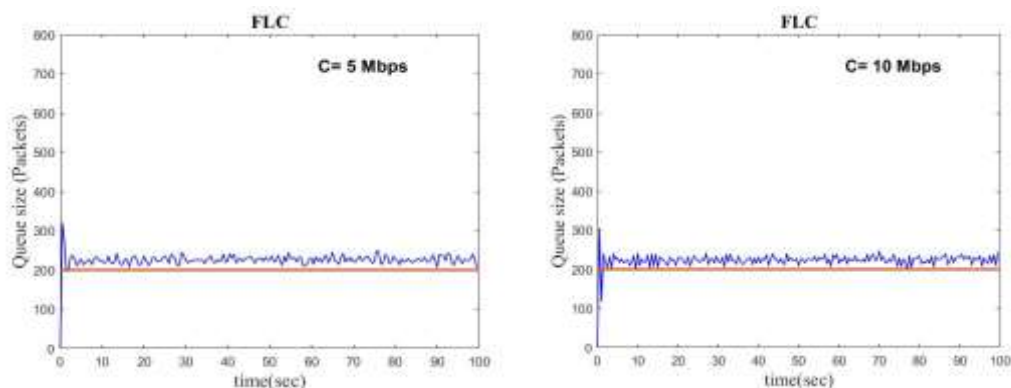


FIG. 10. SYSTEM RESPONSE USING THE CONTROLLER FOR SUDDEN TCP SESSIONS.

C.3 Change of the Value of the Link Capacity (C)

The next test is to see how the FLC responses when the link's ability is changed. The Fig.11. demonstrated the output of the device using the FLC by decreasing the link capacity from its original value to (5, 10) Mbps respectively, and then increasing the link capacity from its original value to 25 Mbps. Other parameters were set as ($T_p=5ms$) and ($N=100$). The Figures showed that even with changes in link capacity values, the queue size tracked the target level (200 packets) Because of the FLC controller . Table VIII shows effects of (mean) and (standard deviation) FLC queue length.



Received 23/3/2021; Accepted 11/6/2021

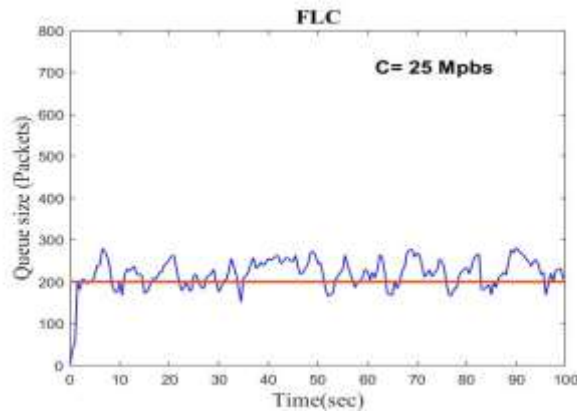


FIG.11 . SYSTEM RESPONSE USING FLC FOR VARIOUS VALUES OF C.

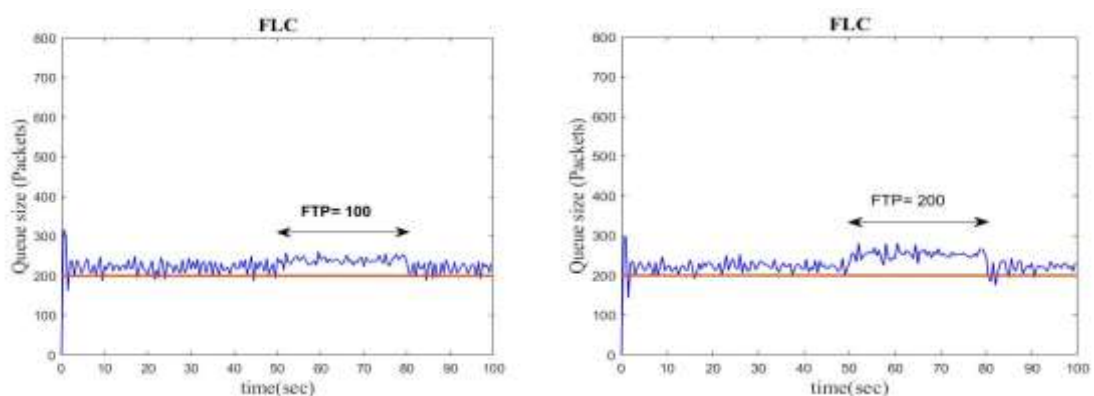
TABLE VIII. MEAN AND DEVIATION VALUES IN DIFFERENT LINK CAPACITY VALUE (C)

C(Mbps)	Mean FLC	Standard Deviation FLC
5	225.405	19.7927
10	222.8	20.7993
25	219.13	37.9315

where by increasing the C, the value of the mean will decrease by (2.8%) to approach the desired value, With an increase in the standard deviation value by (47.8 %), When increase in C to (5, 10, 25) respectively.

C.4 Disturbance Caused by Various FTP Connections

File transfer protocol (FTP) is used to transfer server data and pass client files to the server. An external FTP connection was used as a disruption to assess the proposed controllers' robustness. TCP sessions were set to 100 and additional FTP contacts (100, 200, 500) between 50 and 80 seconds. Fig.12 shows Using the FLC, the system's response. The figures show system response and the controller's ability to track the desired value (200 packets) even with different FTP connections. This can be clearly seen in Table IX.



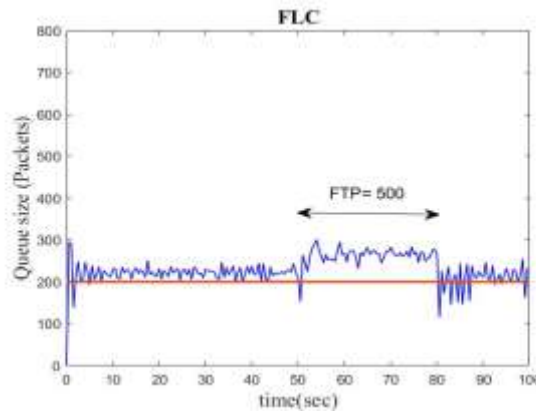


FIG.12 . SYSTEM RESPONSE USING FLC CONTROLLER TO DISTURBANCE FTP CONNECTIONS.

TABLE IX. MEAN AND DEVIATION FOR LENGTH QUEUE DISTURBANCE OF VARIOUS FTP CONNECTION

FTP Connections	Mean FLC	Standard deviation FLC
100	225.595	24.6907
200	230.175	26.9185
500	230.715	34.7319

Although the (mean) and (standard deviation) are increased by increasing the FTP connection, it remains within the acceptable value due to the proposed control FLC.

C.5 Disturbance with Extra UDP

The Particular disruption identified by a non-responsive UDP Flows was used to verify the proposed controllers' robustness. At intervals of 30 and 60 seconds, additional UDP flows (10 and 20) were used. Fig.13. shows the system response when using FLC with additional UDP flows. When the additional UDP streams are 10, Simulation results indicate that the controller can tracking the desired response. Even so, when UDP flows rise to 20, the proposed controller response oscillated away from the desired length. however, It was below and not above the appropriate value of the queue. Table X shows FLC's queue results (mean) and (standard deviation).

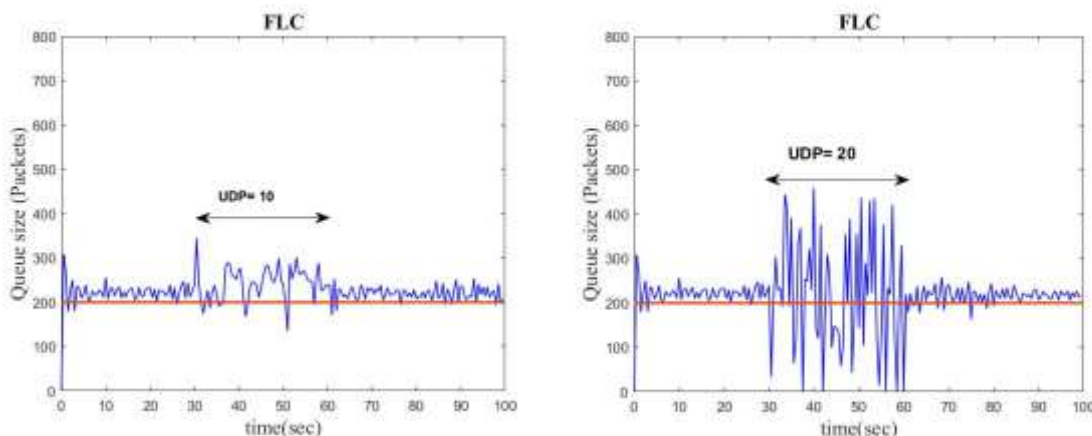


FIG.13 . FLC SYSTEM RESPONSE WITH DISTURBANCE AS MORE UDP FLOWS.

TABLE X. MEAN AND DEVIATION VALUES WITHIN DISTURBANCE OF FURTHER UDP FLOWS

UDP Connections	Mean FLC	Standard deviation FLC
10	227.80	31.3769
20	218.18	27.9741

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

C.6 Change the Queue Size

The proposed controller's robustness was evaluated using various queue sizes. when the length of the queue increased by 25%, 100% and 150% of the initial value (200 packets) equal to (250, 400 and 500) each. Fig.14. showed the simulation results of the system using FLC. The controller made the queue size followed its target level. The Table XI indicates the mean and standard deviation values.

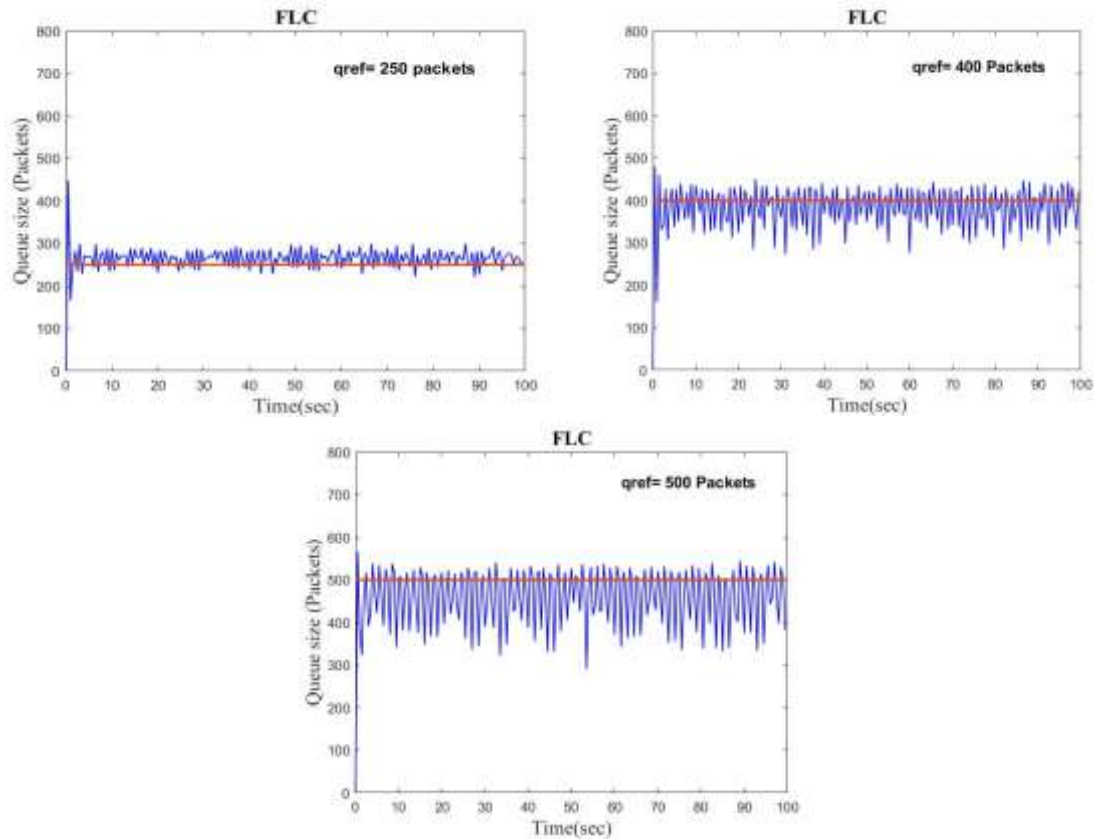


FIG.14 . SYSTEM RESPONSE USING THE FLC WITH DIFFERENT QUEUE SIZES

TABLE XI. MEAN AND DEVIATION VALUES OF DIFFERENT QUEUE SIZE

Queue Reference	Mean FLC	Standard deviation FLC
250	264.045	30.4979
400	381.98	56.0308
500	457.63	73.4366

C.7 Change the Value for Propagation Delays (T_p)

To measure FLC's ability and performance as an AQM when the propagation delay takes different values. Fig.15 indicates the simulation effects of the proposed controller for T_p equals to (15 ms and 20 ms). The FLC as AQM was succeeded by setting queue size for all T_p values near the packet reference value (200). Mean and standard deviation values shown in Table XII.

TABLE XII. MEAN AND DEVIATION VALUES WITHIN DIFFERENT T_p

T_p (msec)	Mean FLC	Standard deviation FLC
15	201.27	38.4506
20	192.395	43.5718

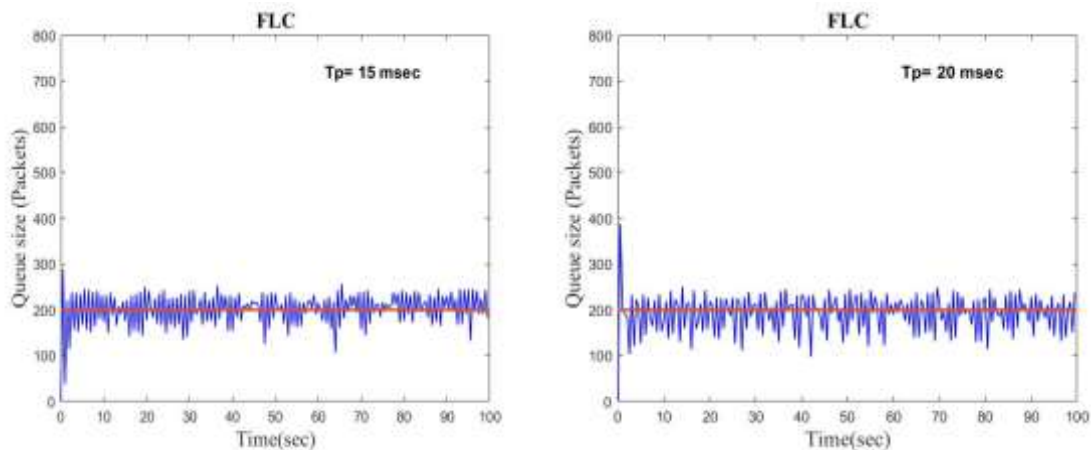


FIG.15 . SYSTEM RESPONSE USING FLC/AQM WITH DIFFERENT TP-VALUES.

This section shows the application of the controller in a real environment via simulation non-linear done via the Network Simulator 2 [NS-2], The PID-like-FLC controller has proven its ability to stay close to desired queue length value 200 packet, It also demonstrated high performance by staying close to the desired value when applying different types of Disturbance in multiple scenarios.

VI. CONCLUSION

The efficiency of FLC was evaluated in this paper when it was used to control congestion in matlab and ns2. The following points can be concluded from the results of the design and simulation:

The Fuzzy like PID controller designed can handle problems of congestion by good queue size monitoring efficiency, high link usage and quicker device response compared with routers. The system response is improved using the optimization algorithm SSO to optimally pick the best Fuzzy-like-PID parameters, as shown in table (3), demonstrating the usefulness of SSO as an optimization process. NS2 is used to perform the nonlinear simulation of the system, It encourages verification of the durability and performance of the proposed controller easily by adjusting the network topology parameters, The proposed controller showed a strong ability to verify the robustness of Dynamic changes in the network as TCP Session, Propagation Delay, and link capacity. MATLAB simulation and the NS2 simulator were used to test the proposed controllers, and compare it with H_{∞} -ACOPID as shown in the tables IV and VI.

REFERENCES

- [1] A. E. Abharian and M. Alireza, "Hybrid GA-BF based intelligent PID active queue management control design for TCP network," ICECT 2011 - 3rd Int. Conf. Electron.
- [2] Niu, Q., Hong, F., & Yan, F. (2011). Fuzzy logic-based AQM controller design research. 2011 IEEE Power Engineering and Automation Conference.
- [3] Y. Qiao and Q. Lei, "A new active queue management algorithm based on self-adaptive fuzzy neural-network PID controller," 2011 Int. Conf. Internet Technol.
- [4] M. Z. Al-Faiz and S. A. Sadeq, "Particle Swarm Optimization Based Fuzzy-Neural Like PID Controller for TCP/AQM Router," *Intell.* pp. 71–77, 2012.

DOI: <https://doi.org/10.33103/uot.ijccce.21.2.7>

- [5] X. L. Fan, F. F. Du, and Z. H. Xie, "Input-rate based adaptive fuzzy neuron PID control for AQM," 2014.
- [6] Feng, C.-W., Huang, L.-F., Xu, C., & Chang, Y.-C. Congestion Control Scheme Performance Analysis Based on Nonlinear RED. IEEE Systems Journal, 2014.
- [7] M. Azadegan, M. T. H. Beheshti, and B. Tavassoli, "Design of proportional-derivative-type state feedback controllers for congestion control of transmission control protocol networks," . pp. 1774–1779, 2015.
- [8] N. Kim Quoc, V. Thanh Tu, and N. Thuc Hai, "Model Integrating Fuzzy Argument with Neural Network Enhancing the Performance of Active Queue Management," EAI Endorsed Trans. p. 150042, 2015.
- [9] B. A. Sadek, T. El Houssaine, and C. Noredidine, "A robust PID controller for active queue management framework in congested routers," 2017 Intell.
- [10] L. A. THEBO, M. A. UNAR, A. Q. K. RAJPUT, and A. I. WAGAN, "Design and comparison of a fuzzy logic controller for Active Queue Control," . pp. 869–874, 2017.
- [11] Z. A. Karam, "Hybrid Fuzzy Congestion Controllers for Computer Networks Tuned by Modified Particle Swarm Optimization," . p. 17, 2018.
- [12] S. K. Bisoy, P. K. Pandey, and B. Pati, "Design of an active queue management technique based on neural networks for congestion control," 11th IEEE Int. Conf. pp. 1–6, 2018.
- [13] C. A. Gomez, X. Wang, and A. Shami, "Intelligent active queue management using explicit congestion notification," 2019.
- [14] C. Wang, X. Chen, J. Cao, and J. Qiu, "Neural Network-Based Distributed Adaptive of Multiple TCP / AQM Networks," pp. 1–9, 2020.
- [15] M. K. Oudah, M. Q. Sulttan, and S. W. Shneen, "Fuzzy type 1 PID controllers design for TCP/AQM wireless networks," .pp. 118–127, 2021.
- [16] C. V. Hollo, V. Misra, D. Towsley and W. B. Gong, "A Control Theoretic Analysis of RED," Proceedings of IEEE INFOCOM, Anchorage, 24-26 April 2001, pp. 1510- 1519.
- [17] L. Reznik, Fuzzy Controllers, Australia, Newnes, 1997, Ch. 4.
- [18] S. M. GirirajKumar, D. Jayaraj, and A. R. Kishan, "PSO Based Tuning of a PID Controller for a High-Performance Drilling Machine," pp. 12–18, 2010.
- [19] A. Luque-Chang, E. Cuevas, F. Fausto, D. Zaldívar, and M. Pérez, "Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives," 2018.
- [20] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," Appl. Soft Comput. J., vol. 30, pp. 614–627, 2015.
- [21] H. I. Ali and K. S. Khalid, "H-infinity Based Active Queue Management Design for Congestion Control in Computer Networks," Second Eng. Conf. Control. Comput. Mechatronics Eng. (ECCCM2, 2014) H-infinity, no. June, 2014.