International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 57, No. 5, October, 2024, pp. 1377-1386
Journal homepage: http://iieta.org/journals/jesa

Adapting Wired TCP for Wireless Ad-hoc Networks Using Fuzzy Logic Control

Yaser Ali Enaya[1*], Abdulamir Abdullah Karim[2], Suha Mohammed Saleh[3], Salam Waley Shneen[4]

[1] Department of ElectroMechanical Engineering, University of Technology, Baghdad 10066, Iraq
[2] Department of Computer Science, University of Technology, Baghdad 10066, Iraq
[3] Ministry of Education, Community of Karakh 1, Baghdad 10011, Iraq
[4] Energy and Renewable Technology Center, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: 50111@uotechnology.edu.iq

https://doi.org/10.18280/jesa.570513

Received: 1 August 2024
Revised: 4 October 2024
Accepted: 10 October 2024
Available online: 28 October 2024

Keywords:
Ad-hoc, transfer function, FLC, PID, TCP

ABSTRACT

This work presents an unprecedented resolution for the lack of harmony between wired network TCP, which transacts with stationary routers, and the Ad-hoc, which appoints mobile hosts as routers, causing failure. This is done by modifying the TCP's queue via fuzzy logic as a controller (FLC). The novelty is that when the mobile router changes location, leading to failure, the controller keeps tracking error signals, which become a behavior prediction base for the next error before occurring. The transfer function is designed with respect to the I/O parameters to optimize the system, leading to the steady with free of fluctuation and overshoot. Other than previous techniques, which depend on fixed criteria or proactive strategies such as AQM, the proposed system is distinguished by dynamic adaptation with continuous changes like mobility and signal power. Using FLC, the system response is improved by reducing time delay, packet loss, and enhancing stability and spectral efficiency. A simulation is designed to continuously monitor these parameters' results, represented by the metrics rise time, undershoot, overshoot, and steady time. For comparison purposes, the PID controller is applied, and the results show surpassing this system where the metrics rise time, undershoot, overshoot, and steady time values are for FLC 0.55, 0.017, 0, and 2.5, respectively, with fluctuation-free, while for PID 0.6, 0.526, 9.341, and 3.6533, respectively, with fluctuation. Furthermore, by dealing the FLC with the challenges of high mobility and the signal power changes, it exceeds the traditional PID controller, as the simulation results show that in terms of reducing the delay time by 28.6%, packet loss by 50%. Besides, there is a remarkable increase in the throughput by 20% and spectral efficiency by 16.7%.

1. INTRODUCTION

The Transmission Control Protocol (TCP), which was developed in the 1980s, is relied upon as a technique for reducing network congestion [1, 2]. TCP was designed with the intention of performing end-to-end congestion control by giving senders access to congestion windows that have lengths with the largest possible number of unrecognized packets [3, 4]. However, due to rapid advancements in the communication industry, TCP is unable to address the congestion growth issue. As a consequence, an innovative approach called the Active Queue Management algorithm (AQM) that foresees congestion before it occurs has emerged [5]. In order to improve QoS, AQM was collaborating with the TCP approach to decrease packet loss and network delay. Early congestion identification and network status reporting are the two primary AQM tasks; they are carried out anytime the router's queue reaches its capacity via signalling received packets [5]. Next, the sender sends feedback packets by AQM as soon as it receives a notification of an overflow, enabling it to respond appropriately [6]. Although AQM is a useful tool for anticipating congestion and alerting users to current network conditions by indicating received packets before router queues are overflowed, its current algorithms have some issues. The most common issue is the parameter's complicated modification to get an optimal output [7]. In this regard, the RED algorithm becomes unstable and sensitive to the network's managing factors because of its need to understand the dynamics parameters of multi-systems [8, 9]. Next, in the case of congestion, the majority of AQMs randomly get rid of packets regardless of the type of flow pattern, which may result in a significant increase in packet delay. Since the packets must adhere to strict delay limitations, this lowers QoS [10]. As a result, academics have been working to enhance the AQM by using various expert systems and controllers, including fuzzy and proportional controllers [11-14]. Many pieces of research have been written about suggesting fuzzy logic as a new technique for TCP queues, proving its ability to solve many problems. For example, the studies [15, 16] apply fuzzy logic. Meanwhile, Bejarbaneh et al. [17] addressed AQM issues by incorporating fuzzy and PID into the networks' TCP, so congestion, packet loss, and delay time will be reduced using nonlinear systems. Similar work can be found in the study [11] where a fuzzy logic controller (FLC) is suggested to enhance AQM performance since fuzzy control improves suitability to dynamic system conditions based on PID. So, by involving optimization techniques in TCP development, such as GA [18] and particle swarm optimization techniques [19], their parameters, data retransmission, and delivery speeds (bits/sec) are managed to measure TCP performance [20]. Moreover, these optimization strategies are used in conjunction with PD [21], PI [22], or PID controllers [23]. For instance, Awad et al. [24] utilizes fuzzy on PID to design AQM in order to reduce the amount of congestion, delay time, and packet loss using nonlinear systems. Jaber et al. [25] recommended adjusting the PI's settings when it is utilized as an AQM by using Biogeography Based Optimization (BBO). Hamidian and Beheshti [26] came to the conclusion that, in the situation of changing network settings, differences in packet loss are minor. They base their research on FPID (fractional PID), AQM, and creating a strong controller that can handle TCP congestions. Additionally, the experiment shows that FPID queues exhibit less packet loss and queue length volatility than PI and PID queues. While this was going on, Sulttan et al. [27] used TCP/AQM, where AQM uses a genetic algorithm (GA) and PI controller. For another illustration [28], where the researchers improved the AQM process by using the PI to address the traffic issue between various IoT networks. As a result, they apply a packet categorization for the IoT network by designating the IP that contains the packet as a priority or non-priority. Then, the AQM keeps the queue status below its maximum capacity and only saves the priority packets, ensuring that they are not discarded. And to achieve this, nearly half of the transmission, which is a non-priority packet, is dropped because the primary function of the IoT is to send commands to distant devices. Though each of these methods improves network performance significantly by reducing packet loss and delay time through managing TCP's queues at congested links, none of them addresses the specific cause of these problems in Ad-hoc networks. So, this causative can be formulated as follows: Reaching the TCP queue steady state is severely constrained by Ad-hoc networks because of their working environment, which is lacking in infrastructure, mobile, and decentralized design, making regular network management of the TCP queue fail to reach stability in rational time [29]. On the other hand, because of their self-organization, Ad-hoc networks provide powerful connectivity. Therefore, they are utilized in critical situations such as catastrophes, force operations, and a VANET, which is a collection of access points consisting of roadsides and vehicles that collaborate through Ad-hoc networks to share information on the traffic status [30, 31]. Nevertheless, because of such environments, Ad-hoc networks have to jump between network topologies, replace QoS, and sometimes use poor resources, leading to complicated communication management. As a consequence, many AI and expert systems are suggested to improve the Ad-hoc performance. One of these suggestions is applying the fuzzy logic control to the TCP queue to provide a network steady state. The reason for choosing the TCP is its importance since it has many algorithms that are dedicated to managing congestion, errors, flow, etc., making it the best target to improve wireless Ad-hoc performance, especially it is designed for wired networks that are incompatible with Ad-hoc networks. This paper develops a new formula matching certain fuzzy logic controller parameters' characteristics to obtain an optimal path to reach into steady state in the shortest time, making fuzzy logic the perfect technique for designing TCP queues dedicated just for the Ad-hoc networks. To integrate this design, the impact of the fuzzy input variables mobility and signal power and, on the other hand, the output variable delay have been explored to improve the control's metrics such as rise time, overshoot, undershoot, and steady time, reaching the goal into steady: overshoot = 0 and fluctuation-free. This work aims to establish a strong foundation for designing an Ad-hoc TCP using fuzzy control technique to address the issue of continuously changing routes for mobile devices by using the MATLAB simulation on the fuzzy that is utilized as AQM with conventional TCP. Therefore, the study identifies the optimum method for an Ad-hoc network by comparing the simulation results of the FLC with the PID technique. According to our knowledge, there is no technique that solves the problem of the wired TCP incompatibility with the wireless Ad-hoc, but there are several attempts to improve the TCP performance in general. One of these techniques, as mentioned before, is the AQM, which is lacking the adaptation with the rapid circumstance changes of the Ad-hoc environments. Meanwhile, the FLC differentiates by its ability to deal with inaccurate givens because of continuously changing in the wireless networks. So, instead of depending on fixed criteria and linear estimates as used in the AQM and PID approaches, the FLC modifies the TCP conduct according to the inputs, such as mobility and signal power, which allows dynamically improving performance. This technique supplies solid improvements in several performance metrics, as can be noticed in this study with reducing the time delay by 28.6% and packet loss by 50% compared with PID. This paper is broken down into four sections: an introduction, a methodology, an implementation, and a conclusion.

1.1 Problem statement and motivation

Ad-hoc network is a vitality technique that is used in complicated environments such as rescue operations and military communications, which make the need for reliable networks very imperative where the topology changes constantly. This technique has no fixed infrastructure, and instead it makes its own dynamic instant infrastructure from available nodes in the location. On the other hand, the TCP of wired networks is considered the backbone of data transfer in most networks. However, its design doesn't take into consideration the changing and decentralization nature of the Ad-hoc network, leading to performance degradation. Therefore, designing a new mechanism becomes a necessity to control the TCP conduct, which this study suggests the FLC to be this controller. The next challenge is that signal power changing over time effects the contact quality, causing high packet loss rates. As a consequence, the wired TCP tries lessening the sending rate when the packet loss occurs, but it doesn't differentiate between the congestion and packet loss caused by poor signal driving to unnecessary performance drooping in the wired networks. Moreover, the absence of the central infrastructure in the Ad-hoc that directs the data flow makes it more complicated, which requires a modified TCP to deal with the decentralization and to adapt to the conditions of each node. The last challenge is that the Jitter and packet loss minimize the QoS and negatively affect the time-sensitive apps such as voice and video calls. Therefore, providing a solution to enhance the TCP performance in the Ad-hoc becomes necessary to ensure continuity of communications in critical environments. Furthermore, improving the network performance amongst these challenges contributes to minimizing the delay and packet loss, which makes the suggested solutions, such as the FLC added value [32, 33].

2. METHODOLOGY

Fuzzy logic is considered an applicable technique to face Ad-hoc obstacles such as mobility, lack of infrastructure, and decentralization by integrating with the TCP queue. This can be done by reducing the fluctuating and steady state reaching time. Fuzzy logic is widely known for its power to simulate complicated, non-linear systems, which makes it perfect for designing a compatible TCP queue with the dynamic, intricate nature of Ad-hoc. The novelty of this paper can be described as providing a steady TCP for the Ad-hoc taken out from unstable TCP by using a unique transfer function that is designed especially for specific I/O fuzzy logic controller parameters. This integration between the fuzzy controller's function and parameters optimize the system by reaching directly into steady via eliminating the overshoot, overshoot = 0, and the fluctuation to improve the Ad-hoc networks' responsiveness and flexibility. These parameters are represented by Signal Power and Mobility as inputs and Delay as an output while the transfer function is represented by Eq. (1).

$$\frac{1}{s^2 + 10s + 20} \tag{1}$$

For comparison purposes and to prove the proposed system's efficiency, this work applied a PID controller to the Ad-hoc TCP queue, which is surpassed by this work as illustrated in the implementation section.

2.1 The TCP model for Ad-hoc network

The TCP model of this system is based on Eqs. (2) and (3).

$$\dot{w}(t) = \frac{1}{\frac{q(t)}{C} + T_p} - \frac{w(t)}{2} \frac{w(t - R(t))}{\frac{q(t - R(t))}{C} + T_p} p(t - R(t)) \tag{2}$$

$$\dot{q}(t) = \begin{cases} -C + \frac{N(t)}{\frac{q(t)}{C} + T_p} w(t) \ if \ q(t) > 0 \\ max\left\{0, -C + \frac{N(t)}{\frac{q(t)}{C} + T_p} w(t)\right\} \ if \ q(t) = 0 \end{cases} \tag{3}$$

where:

w and q are the size of the window and the length of the queue, respectfully, and $\dot{w}, \dot{q}$ are their derivative.

R: Transmission time of full trip ($R = \frac{q}{C} + T_p$).

p: packet sign probability.

N: number of session.

C: link capacity of number of packet per second.

T_p: delay of promulgation.

These equations rule the input to decrease the transmission ratio in the queue bottleneck case by rising 1/R via the addition and reducing w/2 via the multiplication which evaluate window size flow rate at the TCP side. Eq. (3) is the queue length overflow because of the transmission rate exceeding the capacity of the link. Eq. (4) is the saturated input equation with 0-1 boundaries. Figure 1 shows the TCP flow control at Ad-hoc network.

$$sat\left(p(t - R(t))\right) = \begin{cases} 1, p(t - R(t)) \geq 1 \\ p(t - R(t)), 0 \leq p(t - R(t)) < 1 \\ 0, p(t - R(t)) < 0 \end{cases} \tag{4}$$

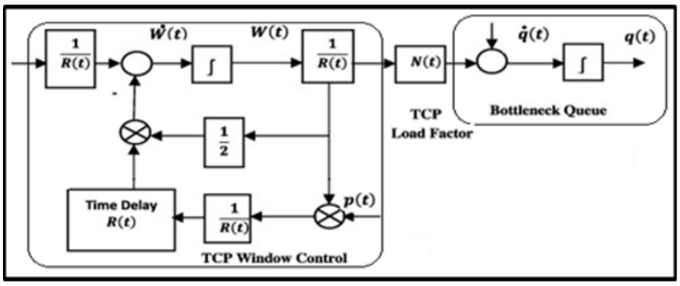

Figure 1. Schematic model of a controller with the Ad-hoc TCP network

2.2 Ad-hoc TCP queue simulation model applying fuzzy logic controller

This section illustrates the simulation model of the system, which is depicted in Figure 2. From the figure, the fuzzy logic controller manipulates the Ad-hoc TCP queue to leverage the network performance by eliminating the overshoot and the fluctuation and that lead the system into steady state in a very short time.

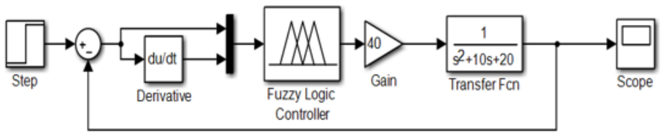

Figure 2. Simulation model of the system at using fuzzy logic controller

This is achieved by integrating the transfer function $\frac{1}{s^2 + 10s + 20}$ with specific I/O variables, which are necessary for optimizing this function.

2.3 Simulation model designing requirements

There are certain steps that have to be implemented to cooperate the fuzzy logic controller with the Ad-hoc TCP queue to design this simulation: first, defining the I/O variables that are in this model: mobility and signal power as inputs and delay as an output. Second, define fuzzy linguistic variable sets for each I/O variable, Figure 3. So, in this work, the set for the input is {Low, Medium, High}, while the output set is {Poor, Average, Excellent}. The mobility and signal power have been chosen as inputs for FLC because of their direct influence on the performance of the Ad-hoc, whose dynamic structure changes continuously as a consequence of nodes movement. This change leads to increasing the instability and delay in receiving the data. Besides, the signal power has a big impact on nodes' contact quality since the poor signal leads to packet loss and increases the delay. Therefore, choosing the delay as an output allows the FLC to improve Ad-hoc performance by reducing it by the adaptation with the inputs [34]. Third, define the membership function. This model uses a triangular membership function. Fourth, generate a fuzzy rule set that masters the controller's conduct by linking I/O variables. The fuzzy-based "if-then" rules are described in Figure 3. Figure 4 is the Rule Viewer for FLC of Ad-hoc, where these rules are used to take specific actions on the input to get the output variable and gain. Fifth, Defuzzification: This work applies the triangle membership functions and fuzzy logic, which are imperative in the Ad-hoc parameter evaluations of accuracy in degrees of difference. The aim of applying fuzzy logic is to eliminate the fogginess of the Ad-hoc queue's input (mobility and signal power) and output (delay) parameters. For this purpose, the triangular membership functions are used to model these variables by estimating the membership degree of any value within the range of the continuous scales of the linguistic variables using Eq. (5).

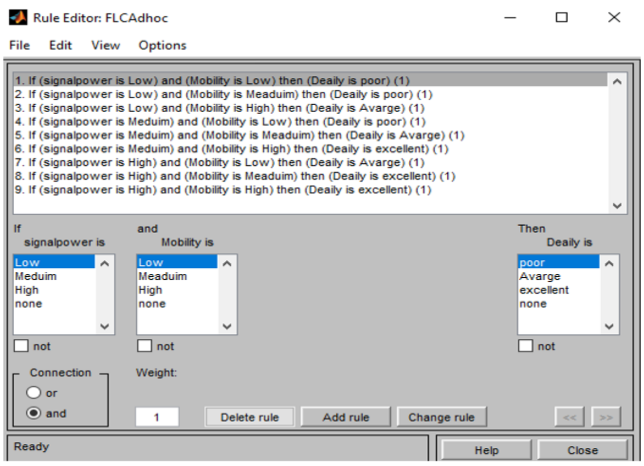

Figure 3. Rule editor for FLC of Ad-hoc

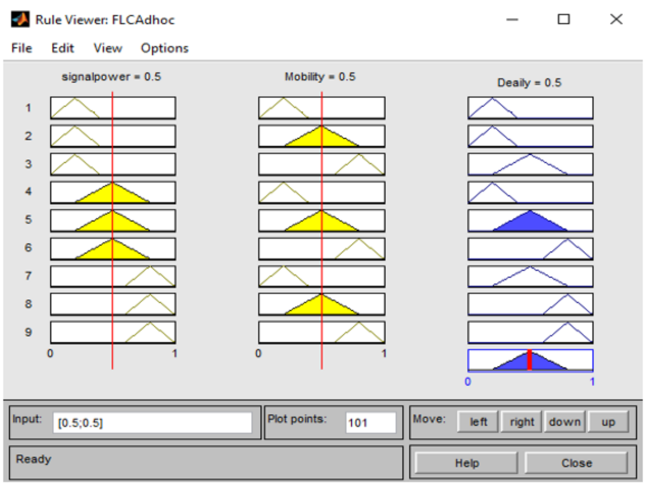

Figure 4. Rule viewer for FLC of Ad-hoc

$$\mu(x) = \max(0, \min(\frac{x-a}{b-a}, \frac{c-x}{c-b})) \qquad (5)$$

where a, b, and c are modeling the triangular function membership. This work prefers to use continuous scales rather than crisp values because they have a wider range, which enables the fuzzy to deal more accurately with systems that have a large number of input differences than the crisp. Moreover, these characteristics make the fuzzy controller adapt easily to the continuous changing of the Ad-hoc state because of its dynamic nature. At last, the feedback, as shown in Figure 2, is in control of the queue size according to the output delay by keeping and using the current output to adapt the next output; this process lets the fuzzy controller estimate the mobile and signal power periodically, reaching efficient performance.

2.4 TCP/PID controller

The PID is a controller that contributes in the industrial field to control speed, flow, and pressure. There are two control methods used by the PID; first, there is the proportional D controller, which uses a feedback loop to make the process parameters' limits stay within the plan. Second, the proportional I controller helps the D controller to improve the PID controller's prediction and manages buffer overflow [24]. Figure 5 provides an illustration of these two methods. The PID's Eq. (6) below is utilized by this work's TCP for the ad-hoc network [35]:

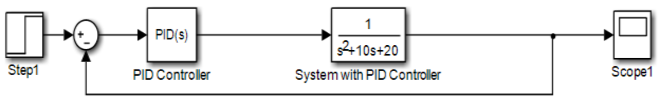

Figure 5. Simulation model of PID controller for Ad-hoc TCP network

$$u(t) = kpe(t) + ki\int e(t)dt + kd\frac{de(t)}{dt} \qquad (6)$$

3. IMPLEMENTATION

The implementation is applied by using the MATLAB, Simulink package, for designing the simulation system and controlling it. The Fuzzy Logic Toolbox is also used for designing and controlling the FLC. This simulation model is designed by the calculation of Laplace equation, Eq. (7), using MATLAB to transform the I/O of the FLC into diagrams to observe and modify them by monitoring the metrics overshoot, undershoot, rise time, and steady time [36, 37].

The factors of this system comprise fuzzy, transfer function, and mobility, which make Laplace the most convenient equation to transform the system domain from time to frequency because of its characteristics [38, 39]. This transforming is represented by Figures 6-11, where Figure 6 depicts the comparison results between the FLC (pink curve) and PID controller (black curve) when applying the two controllers to the TCP queue of the Ad-hoc network; Figure 7 shows the FLC of Ad-hoc for signal power input; Figures 8 and 9 show the Membership Function Editor (MFE) of the FLC for the Ad-hoc TCP queue of the inputs Signal Power and Mobility at low. On the other hand, Figure 10 shows the MFE of the FLC for the Ad-hoc TCP queue of the output Delay at poor. Moreover, 3-D plots of the output delay variable (y-axis) with respect to the input variables mobility (x-axis) and signal power (z-axis) are provided in Figure 11.

The simulation shows the FLC efficiency of controlling the TCP's queue and designing together a system specified to fulfill Ad-hoc's requirements through a comparison with another controller, P|ID. The results are shown in Figure 6, where the pink and black curves represent TCP/ FLC and PID controllers, respectively. The simulation model Information is listed below:

1- 10 movable nodes work as senders, receivers, and routers upon Ad-hoc requests.
2- Mobility Model: using Random Waypoint Mobility Model to simulate the nodes' movement within a specified area.
3- Mobility Pattern: the nodes move at a speed 0-10 m/s within 1000 × 1000 m with random pauses.
4- Traffic Pattern: depending on the TCP, the nodes generate communication between them to transfer the packets.

The other information is mentioned in the below example:

Now, suppose these Ad-hoc conditions: 20 Mbit/s bandwidth bottleneck, 30 ms latency between two mobile routers that are linked to number of senders and receivers. According to Eqs. (2) and (3), link capacity = 20 Mbit/sec; load factor = 65; propagation delay = 0.25 sec; round-trip time = 0.30 sec; input queue size = 300 packets; and overflow queue length = 800 packets (sender side). Figure 6 shows the results of setting these variables in the two system models, FLC and PID. Moreover, Figure 6 depicts the comparison results between the FLC and the PID controllers, which show the superiority of the FLC. Besides, the figure proves that FLC achieves the optimal into steady access without redundancies, fast Rise and steady times, and the ultimate score is overshoot = 0, making the system Fluctuation-free by applying the developed transfer function, Eq. (1), with respect to the fuzzy I/O variables (Mobility, Signal Power, Delay). So, from the figure, the comparison process between FLC and PID metrics includes: for FLC, the rise time = 0.55, which is the time that the controller travels 3/4 the distance into steady state, undershoot = 0.017, and overshoot =0, while for PID, the rise time = 0.6, undershoot = 0.526, and overshoot =9.341, which indicates that all FLC metrics are superior to PID metrics. It is very clear when the x-axis (time) is taken with respect to the above metrics (y-axis), where the FLC takes up 2.5 ms without overshooting or fluctuating to reach the "Into Steady" state, while the PID needs to 3.6533 ms to access the steady state because of its high overshoot and fluctuation.

$$F(s) = L\{f(t)\} = \int_0^{\infty} e^{-st} f(t)dt \tag{7}$$

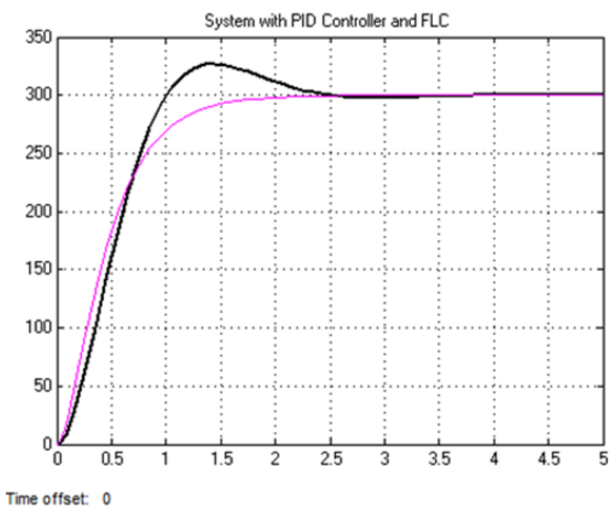

Figure 6. Simulation result of the system at using comparative PID controller with fuzzy logic controller

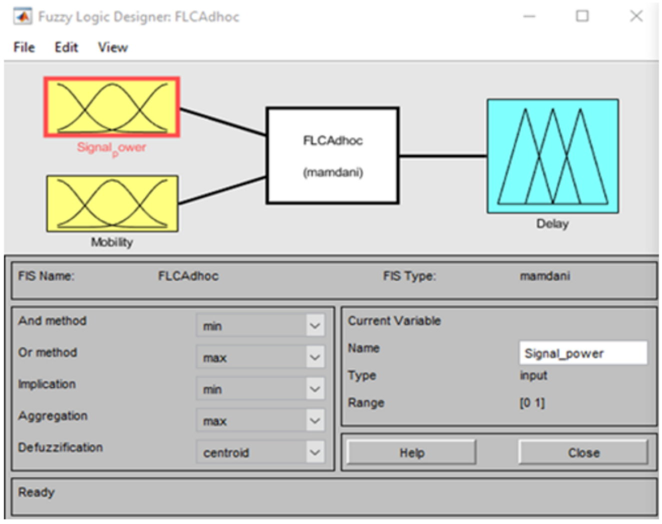

Figure 7. FLC of Ad-hoc for input1 (signal power)

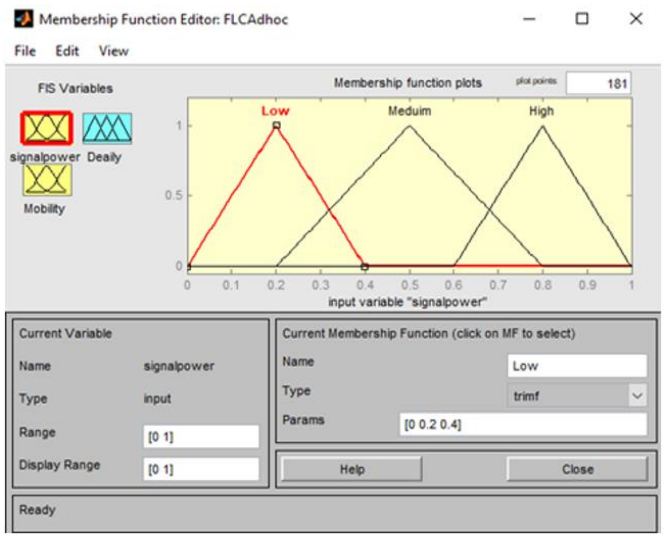

Figure 8. MFE-FLC of Ad-hoc of input1 (Signal Power) at Low

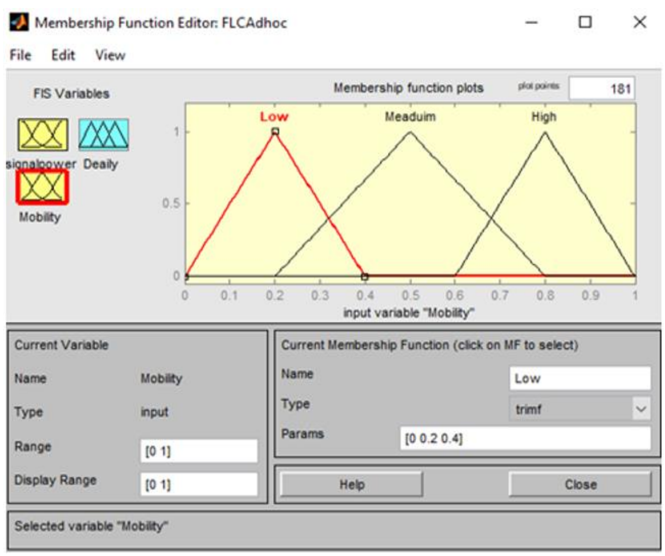

Figure 9. MFE-FLC of Ad-hoc of input2 (Mobility) at Low

Figure 10. MFE-FLC of Ad-hoc of output (Delay) at Poor

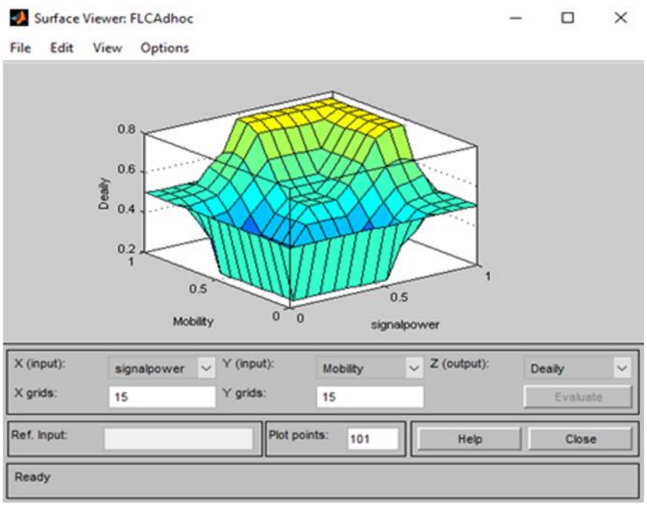

Figure 11. 3D Surface view for FLC of Ad-hoc

3.1 Fuzzy sets

Table 1 lists the fuzzy sets, the Linguistic Variable, continuous scale, which is the upper and lower limits of the values that the linguistic variable may take, and the membership function values for each variable in this work. The triangular membership function values are often selected by the system designer or domain expert and are based on the properties of the Low, Poor, Medium, High, etc. [40]. Example 1 explains how Triangular Membership functions provide useful information for control techniques and decision-making by calculating the degree of membership for particular values.

Table 1. I/O variables and their respective triangulate membership function values

Variable	Linguistic Variable	Continuous Scale	*a*	*b*	*c*
Mobility	Low	0 - 100	0	20	40
	Medium		30	50	70
	High		60	80	100
Signal Power	Low	-100 - 0	-100	-80	-60
	Medium		-70	-50	-30
	High		-40	-20	0
Delay	Poor	0 - 100	0	20	40
	Average		30	50	70
	Excellent		60	80	100

Example 1: Let's consider the linguistic variable degree of the Mobility value = 45 by applying Eq. (5):

$$\text{Low } \mu(45) = \max\left(0, \min\left(\frac{45-0}{20-0}, \frac{40-45}{40-20}\right)\right) = \max\left(0, \min\left(\frac{45}{20}, \frac{-5}{20}\right)\right) = 0$$

Medium: μ (45) = triangular (45, 30, 50, 70) = 0.75, and High: μ (45) = triangular (45, 60, 80, 100) = 0.

For Signal Power with value = -45, Low: μ (-45) = 0, Medium: μ (-45) = 0.75, High: μ (-45) = 0. At last, Delay = 75, Poor: μ (75) = 0, Average: μ (75) = 0, Excellent: μ (75) = 0.75.

From all of these, it can be concluded that "Medium" is predominant for both Mobility and Signal Power, indicating a mild and well-stable value that decreases the overshoot and undershoot. In contrast, the delay parameter has a significant degree of membership in the "Excellent", indicating a larger delay in relation to the low and medium ranges. This implies that longer rise times and more overshoot fluctuations could be caused by higher delays. Mobility is frequently linked to control system engineering, practically with overshooting and undershooting, which promote stability. Meanwhile, the Signal Power correlates to the communication systems, in particular to the rise time and overshooting.

3.2 Integration with the transfer function

To get the best performance in this experiment, the transfer function is integrated with the FLC output by multiplying. For example, from Table 1, assume the delay's continued scale value is 75, so:

$$\text{Integral Transfer Function} = 75 \times \frac{1}{s^2 + 10s + 20} = \frac{75}{s^2 + 10s + 20}$$

Next, from Figure 2, the feedback loop saves the current delay output to use it with the Integral Transfer Function in the next inputs to modify the mobility and signal power in order to get a lesser delay in the next round. By constantly repeating this process, the delay disappears in the shortest time and rises without redundancies and free overshoot, as shown in Figure 6.

3.3 Feedback loop

The feedback loop works as a supervisor in this FLC system taking the integration step results from multiplying the current output (delay) by the Integral Transfer Function to make an adjustment to the input in order to decrease the output time after each loop reaching the Into Steady state. So, this system reacts not only to the Ad-hoc dynamic environment represented by the output but also to the transfer function.

3.4 Comparing the compatibility of wired TCP to Ad-hoc with different controllers

Assume these parameters: - Propagation delay: 0.22 sec, Full-trip time: 0.26 sec, Demand queue size (DQ): 300 packets, Maximum queue length (at sender): 740 packets, Incoming Traffic Rate (ITR): 20 Mbit/s, Packet Size: 1000 bytes, Ffactor (Medium Congestion Level (CL) - Initial): 0.9 (10% reduction), Proportional Gain (Kp) = 0.1, Integral Gain (Ki) = 0.01, and Derivative Gain (Kd) = 0. For comparison purposes, the wired TCP is applied to the Ad-hoc with three scenarios to figure out which one is the best developer for the TCP compatibility to work with wireless networks via tracking the queue status in different intervals.

Scenario 1: Applying the TCP directly to the Ad-hoc without any controller. Converting ITR to packets per sec = (ITR in Mbit/s × 106 bits/Mb) / (packet size in bytes × 8 bits/byte) = (20 Mbit/s × 106 bits/Mb) / (1000 bytes × 8 bits/byte) = 250 packets/sec.

By applying Eq. (8):

$$Q\ (t) = ITR \times t \tag{8}$$

We will calculate queue length (Q (t)) at different intervals (t) as shown below:

Q (0.1 sec) = 250 packets/sec × 0.1 sec = 25 packets

Q (0.2 sec) = 250 packets/sec × 0.2 sec = 50 packets

Q (0.3 sec) = 250 packets/sec × 0.3 sec = 75 packets

By continuing the calculations for the t steps, we get the result as shown in Figure 12 which demonstrates the system performance on the queue. It shows that the demand queue length is unable to be fulfilled as a consequence of the extreme crowding.

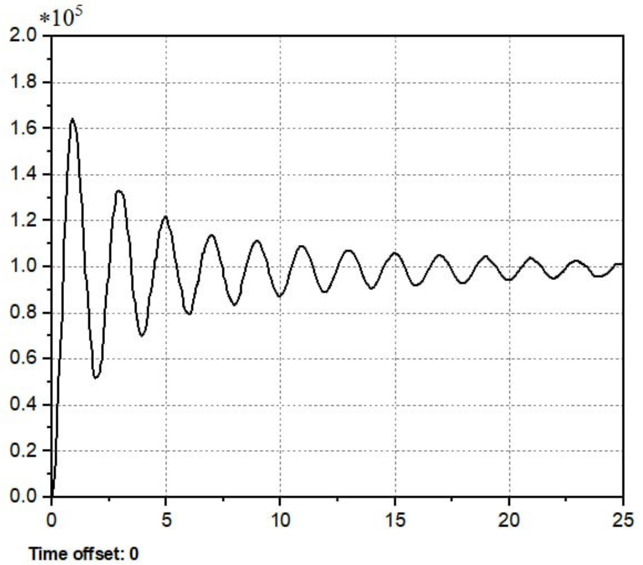

Figure 12. The response of TCP applying to Ad-hoc without a controller

Scenario 2: Applying the TCP to the Ad-hoc with FLC. By applying Eq. (9):

$$Q\ (t) = (ITR \times t) - (SR \times t) \tag{9}$$

Where SR is Effective Sending Rate = ITR ×Ffactor.

Calculate Q (t) at different t:

Q (0.1) = (250 × 0.1) - (250 × 0.9 × 0.1) = 2.5 packets (SR adjusted by Ffactor)

Q (0.2) = (250 × 0.2) - (250 × 0.9 × 0.2) = 5 packets

By continuing the calculations for the t steps, we get the result (pink curve) as shown in Figure 6, which shows a lower and more stable queue length compared to the non-control scenario.

Scenario 3: Applying the TCP to the Ad-hoc with PID. The PID controller adjusts the sending rate (SR) based on the error (e (t)), Eq. (10), and the Change in SR (ΔSR), Eq. (11).

$$e\ (t) = DQ - Q\ (t) \tag{10}$$

$$\Delta SR = Kp \times e\ (t) + Ki \times \Sigma\ (e\ (t)) \tag{11}$$

At t = 0.1 sec (Initial), Assume Q (0.1) = 0 (empty), then error is the DQ since e (0.1) = DQ - Q (0.1) = 300 - 0 = 300 packets, and Σ (e (0.1)) = 0 (at initial). From Eq. (11), ΔSR = 0.1 × 300 + 0.01 × 0 = 30 packets/sec (increase). Update SR with Initial Adjustment based on below equation, which is better representative for PID:

SR (0.1) = Initial SR (assumed close to ITR) + ΔSR = 250 + 30 = 280 packets/sec.

At t = 0.2 sec using the adjusted sending rate:

Q (0.2) = (ITR × t) - (SR (0.1) × t) = (250 × 0.2) - (280 × 0.2) = -6 packets (Negative value: SR too high initially). Therefore, e (0.2) = 300 + 6 = 306; this impractical negative value scenario implies to further adjustment and this is done by minimizing SR. So, updating SR to minimum according to Eq. (12):

$$SR\ (t) = \max\ (ITR + \Delta SR,\ SRmin) \tag{12}$$

SR (0.1) = max (250 + 30, 100) = 280 packets/sec. Therefore, Q (0.2) is still equal -6 packets due to initial overshoot. This is indicating that the packets may be dropped at the queue since SR exceeds the ITR; thus, the PID tries to correct this negative value by increasing the SR through tuning it based on the e (0.2) and a smaller Kp (0.05) to avoid further overshoot. So, the new ΔSR (0.2) = Kp × e (0.2) + Ki × Σ (e (t)) = 0.05 × 306 + 0.01 × 606 packets = 15.3 packets/sec + 6.06 packets/sec = 21.36 packets/sec (rounded to 21) packets/sec. Therefore, adjusted Sending Rate at 0.2: SR (0.2) = SR (0.1) + ΔSR (0.2) = 280 + 21 = 301 packets/sec.

Figure 6 shows the whole diagram (black curve) of reaching the steady state of the TCP with PID. From the results of the above three scenarios, it can conclude the following:

Scenario 1: A steady state may not certainly be reached.

Scenario 2: FLC reaches the steady state faster than scenario 1 and scenario 2.

Scenario 3: With tuning, a PID controller can potentially reach the steady state. However, because of the kd and ki gradually adjustments, this scenario would be slower than scenario 2.

3.5 The quantitative metrics for the performance

Beside the traditional metrics overshoot, undershoot, settling time, and rising time, a wider set of quantitative metrics was applied to guarantee a comprehensive evaluation, which are listed below:

1- **Delay Time**: the analysis, by applying Eq. (13), of the FLC and PID shows that FLC reduces the Delay Time in the high movements' environments.

$$D = T_{arrival} - T_{sent} \tag{13}$$

where *D*: Delay Time and *T:* time of sending or receiving packet.

2- **Packet Loss**: an analysis of packets' ratio that haven't reached the receiver because of congestion or the poor signal proves that the FLC overpasses the PID, especially with heavy nodes' density and movements. Eq. (14) was used for analysis.

$$P_{loss} = \frac{N_{sent} - N_{received}}{N_{sent}} \tag{14}$$

where P is Packet and N is the number of sent or received packet.

3- **Throughput**: using Eq. (15), the FLC shows an increase in the throughput with an improvement in the TCP performance in the Ad-hoc network.

$$T = \frac{N_{received} - S_{packet}}{T_{total}} \quad (15)$$

where T is the time and S is the packet size in bit.

4- **Jitter**: the results show that the FLC lowers the Jitter, leading to improve Ad-hoc stability, which is important for time-sensitive applications such as chats and videos. Jitter is calculated by applying Eq. (16).

$$J = |D_n - D_{n-1}| \quad (16)$$

where D is the time delay.

5- **Spectral Efficiency**: by applying Eq. (17), the FLC proves its effectiveness to improve the Spectral Efficiency compared with the traditional systems.

$$S_{eff} = \frac{T}{B} \quad (17)$$

where T is the throughput and B is the bandwidth.

The results of applying the simulation using MATLAB and the equations are shown in Table 2, which is based on a limited network using FLC and PID controllers. Analyzing these results proves the FLC surpassing in improving the performance, reducing the delay, and increasing the throughput.

Table 2. Quantitative metrics for the performance of the FLC and PID

Quantitative Metric	FLC	PID	FLC Improving % VS PID
Delay Time msec	2.5	3.5	28.6
Packet Loss %	2.5	5	50
Throughput Mbit/s	18	15	20
Jitter Msec	5	7	28.6
Spectral Efficiency	3.5	3	16.7

3.6 Performance evaluation under different network

Conditions

To evaluate the robustness of this work, the below applications were performed:

1- Applying different ranges of node density, between 10-50 nodes, in a 1000 × 1000 m area to analysis the delay and stability in order to ensure the system's work ability under heavy density circumstances. The system with FLC shows stability in high density, reaching 50 nodes in spite of the slight increase in the delay.

2- Applying different mobility levels by using different nodes' speeds between 1-10 m/s to reflect slow and fast mobility. The aim is to evaluate the system's ability to deal with the rapid changes in the topology as a result of node movements. There was a remarkable increase in the delay because of the large number of the topology changes, but the FLC improved the response and reduced delay compared with the PID, which indicates that FLC has an adaptation ability with the rapid changes.

3- The traffic load was adjusted to include the range from low traffic (low TCP's connections), 5 Mbit/s, to high traffic, 50 Mbit/s. Then, it was analyzed to see the impact of the adjustment on the delay, stability, and rise time. The system proves its efficiency to deal with heavy traffic reaching 50 Mbit/s with relatively small delay.

4. CONCLUSIONS

This study states the importance of FLC to improve wired TCP performance in wireless Ad-hoc networks, which is approved by the outstanding results compared with the PID. These results confirm the system's suitability for special apps that need sensitive response times and high stability. Moreover, based on these results, this work can offer an effective solution to the problems of the Ad-hoc traffic and support the QoS. On top of that, the study initiates a strong basis for future research on control development techniques in the field of wireless networks. The inferences of the experimental part are listed below:

1- Mastering Signal Power and Mobility to balance the relationship with Into Steady state guides to optimize the metrics' values.

2- Undershoot, Overshoot, and Rise Time are temporary system situations, and they are results of Integral Transfer Function and fuzzy logic modification as response to changes.

3- The linguistic variables and metrics relevancy is formed according to the engineering concepts.

4- Metrics reflect the system statues between the start-Into Steady phases influenced during this journey by the initial conditions, system changes, the fuzzy controller's I/O variables, and integral and feedback loop steps.

5- The results show that the TCP fuzzy controller model to manage the TCP/Ad-hoc network inconsistency outperforms conventional and PID TCP models.

6- The fuzzy controller with the transfer function of this model are important to expand the Ad-hoc covering area since applying the model on mobiles that connect to Ad-hoc networks makes them reliable routers.

7- In Eq. (9), while dividing by t is mathematically the right step, the original equation shows the time dependency of queue building and to keep tracking the queue length at specific time.

8- A suggestion for future work is taking in count the concurrency between the I/O parameters with the metrics to measure system stability.

9- By reducing the time delay and packet loss and increasing the throughput, the system proves its ability to deal with the dynamics of wireless networks in the heavy nodes' density and mobility environments.

4.1 Future work

In spite of the promising results, we plan for future work to consolidate the system performance in a real environment by conducting field experiments using more developed simulation tools such as NS-3 or OMNET in addition to wireless access points and mobile sensors. These experiments will enable us to measure the performance in real circumstances, including noise effects, signal interference, and the real challenges of the nodes' movements.

REFERENCES

[1] Putra, M.A.R., Ahmad, T., Hostiadi, D.P. (2022). Analysis of botnet attack communication pattern behavior on computer networks. International Journal of Intelligent Engineering & Systems, 15(4): 533-544. https://doi.org/10.22266/ijies2022.0831.48

[2] Athab, S.D., Karim, A.A. (2023). A tagging model using segmentation proposal network. Fusion: Practice and Applications, 13(2): 136-144. https://doi.org/10.54216/FPA.130212

[3] Enaya, Y.A., Deb, K. (2014). Network path optimization under dynamic conditions. In 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, pp. 2977-2984. https://doi.org/10.1109/CEC.2014.6900603

[4] Jassim, S.A., Farhan, A.K. (2022). Combined Chebyshev and logistic maps to generate pseudorandom number generator for internet of things. International Journal of Electrical and Computer Engineering (IJECE), 12(3): 3287-3297.

[5] Ma, H., Luo, X., Xu, D. (2023). Intelligent queue management of open vSwitch in multi-tenant data center. Future Generation Computer Systems, 144: 50-62. https://doi.org/10.1016/j.future.2023.02.018

[6] Poonyanirun, S., Kanchanaharuthai, A. (2023). Congestion tracking control for wireless TCP/AQM network via immersion and invariance. Asian Journal of Control, 25(3): 1834-1850. https://doi.org/10.1002/asjc.2824

[7] Ghani, N.M.A., Nasir, N.M., Hashim, A.A.A. (2024). Comparative analysis of spiral dynamic algorithm and artificial bee colony optimization for position control of flexible link manipulators. Journal Européen des Systèmes Automatisés, 57(3): 773-779. https://doi.org/10.18280/jesa.570315

[8] Hassan, S.O., Ajaegbu, C., Odule, T.J., Abdullah, K.K.A., Akande, O., Ayinde, S., Solanke, O.O. (2023). A Triplex Region-Random Early Detection (TR-RED) algorithm for active queue management in internet routers. International Journal of Computing and Digital Systems, 14(1): 193-200. http://doi.org/10.12785/ijcds/140117

[9] Mohammed, B.M., Alsaadi, M., Khalaf, M., Awad, A.S. (2024). Game theory-based multi-hop routing protocol with metaheuristic optimization-based clustering process in WSN for precision agriculture. Journal Européen des Systèmes Automatisés, 57(3): 653-662. https://doi.org/10.18280/jesa.570302

[10] de Morais, W.G., Santos, C.E.M., Pedroso, C.M. (2022). Application of active queue management for real-time adaptive video streaming. Telecommunication Systems, 79(2): 261-270. https://doi.org/10.1007/s11235-021-00848-0

[11] Oudah, M.K., Sulttan, M.Q., Shneen, S.W. (2021). Fuzzy type 1 PID controllers design for TCP/AQM wireless networks. Indonesian Journal of Electrical Engineering and Computer Science, 21(1): 118-127.

[12] Mahmood, S.H., Farhan, A.K., Abbas, T. (2023). Fingerprint classification via deep convolutional neural networks: A survey. In AIP Conference Proceedings, Cincinnati, USA. https://doi.org/10.1063/5.0150738

[13] Jabbar, H., Sultan, K.F., Anead, H.S. (2022). Design finned electronic curtain for control of solar collector using nanofluids and fuzzy logic control. In AIP Conference Proceedings, Najaf, Iraq. https://doi.org/10.1063/5.0066787

[14] Abd Alkreem, M.H., Karim, A.A. (2021). Feature selection using artificial bee colony for fruit classification. Journal of Physics: Conference Series, 1818(1): 012062. https://doi.org/10.1088/1742-6596/1818/1/012062

[15] Vargheese, M., Vanithamani, S., David, D.S., Rao, G.R.K. (2023). Design of fuzzy logic control framework for qos routing in manet. Intelligent Automation & Soft Computing, 35(3): 3479-3499. http://doi.org/10.32604/iasc.2023.030948

[16] Afifi, W., Hefny, H.A., Darwish, N.R. (2022). Adaptive Kalman filter for free GPS localization with fuzzy intersection method. International Journalof Intelligent Engineering & Systems, 15(4): 394-405. https://doi.org/10.22266/ijies2022.0831.35

[17] Bejarbaneh, E.Y., Bagheri, A., Bejarbaneh, B.Y., Buyamin, S., Chegini, S.N. (2019). A new adjusting technique for PID type fuzzy logic controller using PSOSCALF optimization algorithm. Applied Soft Computing, 85: 105822. https://doi.org/10.1016/j.asoc.2019.105822

[18] Yushuang, W., Yongli, X. (2023). Genetic algorithm-based congestion control optimisation for mobile data network. Applied Mathematics and Nonlinear Sciences, 8(1): 2405-2412.

[19] Belgaum, M.R., Musa, S., Ali, F., Alam, M.M., Alansari, Z., Soomro, S., Su'ud, M.M. (2023). Self-socio adaptive reliable particle swarm optimization load balancing in software-defined networking. IEEE Acess, 11: 101666-101667. https://doi.org/10.1109/ACCESS.2023.3314791

[20] Zong, L., Bai, Y., Zhao, C., Luo, G., Zhang, Z., Ma, H. (2020). On enhancing TCP to deal with high latency and transmission errors in geostationary satellite network for 5G-IoT. Security and Communication Networks, 2020(1): 6693094. https://doi.org/10.1155/2020/6693094

[21] Rai, A., Das, D.K. (2022). Ennoble class topper optimization algorithm based fuzzy PI-PD controller for micro-grid. Applied Intelligence, 52(6): 6623-6645. https://doi.org/10.1007/s10489-021-02704-9

[22] Abdolrasol, M.G., Hannan, M.A., Hussain, S.S., Ustun, T.S. (2022). Optimal PI controller based PSO optimization for PV inverter using SPWM techniques. Energy Reports, 8: 1003-1011. https://doi.org/10.1016/j.egyr.2021.11.180

[23] Khudhair, M., Ragab, M., AboRas, K.M., Abbasy, N.H. (2022). Robust control of frequency variations for a multi-area power system in smart grid using a newly wild horse optimized combination of PIDD2 and PD controllers. Sustainability, 14(13): 8223. https://doi.org/10.3390/su14138223

[24] Awad, B.A., Oudah, M.K., Enaya, Y.A., Shneen, S.W. (2023). Simulation model of ACO, FLC and PID controller for TCP/AQM wireless networks by using MATLAB/Simulink. International Journal of Electrical and Computer Engineering, 13: 2677-2685. https://doi.org/10.11591/ijece.v13i3.pp2677-2685

[25] Jaber, M.H., Oudah, M.K., Sulttan, M.Q., Shneen, S.W.

(2023). BBO tuned PI controller for the stability of TCP networks. Telecommunication Computing Electronics and Control, 21(2): 264-271. http://doi.org/10.12928/telkomnika.v21i2.21795
[26] Hamidian, H., Beheshti, M.T. (2018). A robust fractional-order PID controller design based on active queue management for TCP network. International Journal of Systems Science, 49(1): 211-216. https://doi.org/10.1080/00207721.2017.1397801
[27] Sulttan, M.Q., Jaber, M.H., Shneen, S.W. (2020). Proportional-integral genetic algorithm controller for stability of TCP network. International Journal of Electrical and Computer Engineering, 10(6): 6225-6232. https://doi.org/10.11591/ijece.v10i6.pp6225-6232
[28] Domański, A., Domańska, J., Czachórski, T., Klamka, J., Szyguła, J., Marek, D. (2021). The IoT gateway with active queue management. International Journal of Applied Mathematics and Computer Science, 31(1): 165-178. https://doi.org/10.34768/amcs-2021-0012
[29] Shneen, S.W., Sulttan, M.Q., Oudah, M.K. (2021). Design and implementation of a stability control system for TCP/AQM network. Indonesian Journal of Electrical Engineering and Computer Science, 22(1): 129-136. https://doi.org/10.11591/ijeecs.v22.i1.pp129-136
[30] Mchergui, A., Moulahi, T., Zeadally, S. (2022). Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs). Vehicular Communications, 34: 100403. https://doi.org/10.1016/j.vehcom.2021.100403
[31] Enaya, Y., Mohammed, M.J., Bilal, G. (2020). Password-free authentication for smartphone touchscreen based on finger size pattern. International Association of Online Engineering, 14(19): 163-179. https://doi.org/10.3991/ijim.v14i19.17239%0d
[32] Rajeswari, A.R., Lai, W.C., Kavitha, C., Balasubramanian, P.K., Srividhya, S.R. (2023). A trust-based secure neuro fuzzy clustering technique for mobile ad hoc networks. Electronics, 12(2): 274. https://doi.org/10.3390/electronics12020274
[33] Khan, S.A., Lim, H. (2022). Novel fuzzy logic scheme for push-based critical data broadcast mitigation in VNDN. Sensors, 22(20): 8078. https://doi.org/10.3390/s22208078
[34] Choudhary, S., Narayan, V., Faiz, M., Pramanik, S. (2022). Fuzzy approach-based stable energy-efficient AODV routing protocol in mobile ad hoc networks. In Software Defined Networking for Ad Hoc Networks, Springer, Cham, pp. 125-139. https://doi.org/10.1007/978-3-030-91149-2_6
[35] Eissa, S.A., Shneen, S.W., Ali, E.H. (2023). Flower pollination algorithm to tune PID controller of TCP/AQM wireless networks. Journal of Robotics and Control, 4(2): 149-156.
[36] Ahmed, W.S. (2020). The impact of filter size and number of filters on classification accuracy in CNN. In 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, pp. 88-93. https://doi.org/10.1109/CSASE48920.2020.9142089
[37] Hassoon, A.J., Anead, H.S., Sultan, K.F. (2023). Design an electronic curtain with an acrylic mirror to enhance the solar collector's efficiency using artificial intelligence techniques. In AIP Conference Proceedings, Baghdad, Iraq, 2830(1). https://doi.org/10.1063/5.0156853
[38] Åström, K.J., Murray, R. (2021). Feedback systems: An introduction for scientists and engineers. Princeton University Press.
[39] Dayananda, P., Srikantaswamy, M., Nagaraju, S., Nanjundaswamy, M.H. (2024). A machine learning-based smart grid protection and control framework using Kalman filters for enhanced power management. Journal Européen des Systèmes Automatisés, 57(3): 639-651. https://doi.org/10.18280/jesa.570301
[40] Kaur, A., Kacprzyk, J., Kumar, A., Kaur, A., Kacprzyk, J., Kumar, A. (2020). A brief introduction to fuzzy sets. Fuzzy Transportation and Transshipment Problems, 385: 11-29. https://doi.org/10.1007/978-3-030-26676-9_2