Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

A Novel Congestion Control Scheme using Firefly Algorithm Optimized Fuzzy-PID Controller in Wireless Sensor Network

Roland T. Tweh[1], Edwin O. Ataro[2] and George N. Nyakoe[3]

[1]Department of Electrical Engineering, Pan African University Institute for Basic Sciences Technology and Innovation, Nairobi, Kenya, rolandtweh95@gmail.com

[2] School of Electronics & Electrical Engineering, The Technical University of Kenya, Nairobi, Kenya, atato@tukenya.ac.ke

[3] Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya, nyakoe@eng.jkuat.ac.ke

***Correspondence:** Roland T. Tweh; Email: rolandtweh95@gmail.com; Tel.: +254-113-859-155

ABSTRACT- Wireless Sensor Networks (WSNs) consist of several sensor nodes, each of which may collect, receive and transmit data. In recent years, WSNs have emerged as essential technologies due to their ubiquity in applications such as the military, smartphones, disaster management, healthcare monitoring, and other surveillance systems. The inability to send data from the sensor node promptly and the impossibility of new data reaching the node's queue indicate of network congestion. The packet will be either discarded or delayed, which will cause more data loss, longer transmission delays, reduced network throughput, and lower network quality of service. To address this problem, this paper proposes an efficient and novel Firefly Algorithm-optimized Fuzzy-PID (FA-Fuzzy-PID) controller for congestion control in Wireless Sensor Networks (WSNs). The proposed control technique used a fuzzy control algorithm to overcome the standard PID controller's slow optimization parameter, low calculation accuracy, and limited adaptability. The Firefly Algorithm (FA) was used to optimize the PID parameters increment from the Fuzzy-PID controller. The proposed controller was designed in MATLAB and analyzed using Network Simulator 3. Simulation results from the proposed control method achieved 92.00% higher throughput, 44.49% lower packet loss rate, and 75.00% lower for both congestion level and queue size when compared to Cuckoo Search FuzzyPID (CFPID) and Particle Swarm Optimization-Gravitational Search Algorithm (PSOGSA). These improvements will allow regular data flow and improve the overall performance of WSNs. Based on these improvements, the proposed technique is an efficient control method for congestion control in the WSN because end-users will experience faster data transfer speeds and fewer network delays.

Keywords: Firefly Algorithm, Wireless Sensor Network, Congestion Control, PID controller, Fuzzy-PID

ARTICLE INFORMATION

Author(s): Roland T. Tweh, Edwin O. Ataro, and George N. Nyakoe;

Received: 16-11-2022; **Accepted**: 19-01-2023; **Published**: 10-02-2023;
e- ISSN: 2347-470X
Paper Id: IJEER220952
Citation: 10.37391/IJEER.110107
Webpage-link:
www.ijeer.forexjournal.co.in/archive/volume-11/ijeer-110107.html

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

One of the essential technologies in recent years is WSNs [1,2]. The "carry-send" form of data transmission is a capability of WSNs. Since data collection and transmission are not time-limited, network nodes could receive enormous amounts of data in a matter of seconds [3]. In WSNs, high network congestion will significantly affect data delivery, and communication will either be discarded or delayed, thus increasing data loss [4]. Additionally, a node service life can be significantly reduced if left in a high load and full queue state for a lengthy period [5]. Congestion is an unanticipated state that results in packet dropouts, delays, reduced throughput, increased queueing, network buffering, and retransmission [6]. It can also occur due to suspicious or undesired network activity [7]. In WSNs, there are two common causes of congestion: buffer overflow and link collision. Buffer overflow (node-level congestion), as depicted in *Figure 1a*, happens when the quantity of sent packets surpasses the packet handling ability of a single node (node A), resulting in packet losses and waste of node energy. Link collision (link-level congestion), on the other hand, happens when multiple vigorous nodes seek to interact with a single node (node B) simultaneously, resulting in the loss of packet owing to struggle and intrusion, as displayed in *Figure 1b* [8].

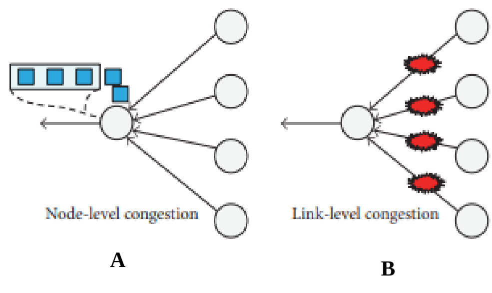


Figure 1: Appearance scenaris od congestion in WSNs [8]

2. RELATED WORKS

Recently, many studies on congestion control in WSN have been conducted using soft computing-based methods including PID and Fuzzy Logic Controllers. With the PID controller, its slow parameter optimization and limited adaptability are limitations that affect its operation. For instance, a redesigned PID controller that improves control while reducing energy usage without sacrificing transmission link quality was proposed by [9]. This result, instead, ignores the issue of low accuracy in PID controllers for high-demand control environments. In [10-13], researchers have introduced various PID controller improvement approaches, yet these still need to be improved when used with WSNs. This study discovered that low adaptability and low calculation accuracy are issues that must be considered when applying PID controllers to control congestion in WSNs. The Fuzzy Logic has also been used with other techniques in addressing the problem of Congestion in WSNs. In a fuzzy algorithm-based adaptive congestion control (FBACC), fuzzy logic was used for congestion estimation [14]. Utilizing the rate of traffic, occupancy of the buffer, and participation, congestion was detected. This protocol did an excellent job by reducing packet loss by 6%, but it also had issues with extremely low throughputs and high network congestion levels. A fuzzy logic based on fairness congestion control for a distrustful WSN (FCCTF) was introduced to enhance packet delivery [15]. This method assessed how well each node could identify and isolate malicious nodes while some packets were at risk of dropping due to overflow. Although the FCCTF was able to increase packet delivery up to 18.5% and decrease packet drop by 20%, it failed to solve the issue of congestion level, which led to a high level of congestion on the network.

Based on the limitations of PID and Fuzzy techniques that have been used, meta-heuristic algorithms were proposed to overcome those limitations thereby improving network performance [16]. In nature, social groupings work collaboratively to forward a common objective. When social groups are compared to wireless sensor networks, it is evident that nodes in WSNs perform tasks collectively like members of social groups. By simulating the social behavior of swarms, which are small groups of interacting, low-IQ agents, swarm intelligence has been proposed as a way to decrease congestion. In [17], congestion is managed using a hybrid multi-objective optimization (PSOGSA) that combines PSO and GSA. PSOGSA is used to optimize and manage the data arrival rate from the child node to the parent node, which accounts for the node energy in the suitable fitness function. This technique was limited because of PSO's slow convergence rate resulting in low network performance metrics. In, the Cuckoo Search algorithm was used to optimize the output variable of the Fuzzy-PID controller. This technique was limited because of the cuckoo search's inability to subdivide the entire population (nodes) into subgroups thereby resulting in low-performance metrics and inaccuracy in results. An Active Queue Management (AQM) that incorporates Random Early Detection (RED) and Fuzzy-PID methods are provided in [18] to evaluate packet loss rate. Occupancy of buffer, rates of node, and congestion notification implicit were used to identify the congestion level. In [19], the procedure results in efficient energy, but it needs to be equitable. The proposed methodology results are contrasted with those of the current Energy Aware clustering algorithm in [20-22]. Network nodes' performance metrics like packet delivery ratio and network lifetime increase when the firefly algorithm is employed to address the clustering issue.

The review of related literature shows many problems with the various congestion control approaches utilized in different network environments. These problems range in relevance from low to high network performance metrics. As a result, before beginning the implementation phase, researchers can evaluate the mechanism and any associated problems using a correctly constructed virtual testbed as a framework. To achieve a better method to control congestion that satisfies the network needs, changes to the framework can be implemented. The queueing length of the node is one of the crucial issues that need to be addressed in systems for controlling congestion.

According to the analysis done, no researcher has combined Firefly Algorithm (FA) and the Fuzzy-PID algorithm for congestion control in WSNs. Yet, comprehensive studies in [22-24] have proven that FA has significant advantages over popularly used meta-heuristic algorithms like Cuckoo Search (CS), Ant Colony Optimization (ACO), Partial Swarm Optimization (PSO), and Genetic Algorithm (GA). The automated separation of the entire population into subgroups, the capacity to naturally handle multimodal optimization, the high ergodicity and the variety of its solution, and the rapid convergence time are only a few of these advantages FA has over popularly used meta-heuristic algorithms.

Therefore, in this paper, the authors propose Firefly Algorithm-optimized Fuzzy-PID (FA-Fuzzy-PID) controller design under various operating situations. The aim was to allow a regular data flow and improve the overall performance of WSNs allowing end-users to experience faster data transfer speeds and fewer network delays.

The contributions of this paper, the authors propose a novel congestion control scheme based on Firefly optimized Fuzzy-PID controller based on the queue model. The system packet loss was determined using the proposed congestion control method's current queue length to ensure that WSNs will respond in real-time. In addition, the queuing theory was used to create a way of analysis for assessing essential performance indicators such as packet loss rate, queue length, throughput, and congestion level. As a result, these performance indicators improve the Quality of Service on WSNs. Finally, the congestion control process was modeled using a cluster head system architecture.

The rest of this paper is arranged as follows: *Section 2* briefly summarizes the studies related to congestion control in WSNs. *Section 3* discusses the modeling of the queue length. The architecture of the proposed FA-Fuzzy-PID controller is discussed in *Section 4*. *Section 5* covers the simulation process and analysis of the results through conclusive studies, analysis, and comparison with already-in-place algorithms. Finally, the research is concluded in *Section 6*.

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

3. QUEUE MODELLING

This section introduces the cluster head system architecture of the proposed system that provides service for WSNs. Based on this framework, a model for controlling congestion in WSNs was created.

3.1 System Overview

There should be an effective service computing technique for data processing that processes the raw data collected from sensor nodes to bridge the gap between information collected from the sensor nodes and those received by end-users. *Figure 2* shows a graphical depiction of how the entire network scenario is structured, from how the sensor nodes collected data and how it got to the end-users. This process was done in three steps: Sensing, Processing, and Servicing.

- Sensing was done by the sensor nodes, mainly for collecting data from the physical environment/object they monitored. These sensor nodes processed many different data quickly and transmitted relevant information to the cluster head node.
- The processing was done at the cluster head node, which acts as the brain of the entire system. The cluster head node processed and computed the information sent to the end users.
- Servicing was done at the sink/base station where the outputted information from the cluster head node was finally encapsulated and provided to the internet and subsequently to the end users.

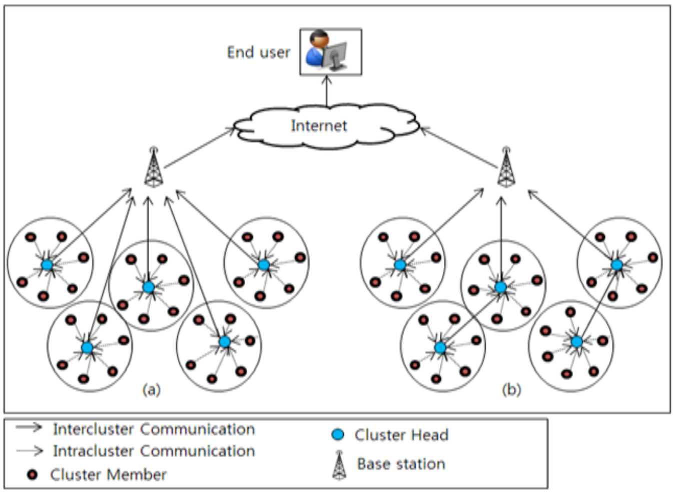


Figure 2: A cluster-head system architecture for WSN [3]

3.2 System model

The proposed system queue length model assumes input and output packets for the system to be $M/M/1/\max_{th}$. Where the arrival rate λ is Poisson distribution, service rate μ is experiential distribution, the queue length is finite and the processing is done through one server.

Figure 3 depicts the distributed queue model based on the cluster head architecture. This congestion control model has queue thresholds: a minimum threshold ($\min_{th}$) and a maximum threshold ($\max_{th}$).

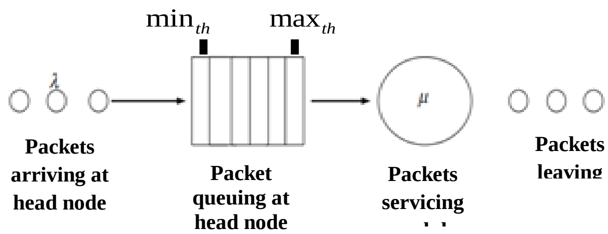


Figure 3: Distributed queue model [18]

Random Early Detection (RED) technique is one of the known Active Queue Management (AQM) techniques used to detect and control congestion [25]. In the RED technique, the current queue length (L_q) is calculated for every arriving packet at the router buffer. When L_q is less than the minimum threshold no congestion occurs and thus no packets can be dropped. However, if L_q is equal to or greater than the maximum threshold, heavy congestion is presented, and every arriving packet will be dropped to manage the congestion. This RED technique is limited because it increases the packet loss rate.

Unlike the RED technique, the proposed FA-Fuzzy-PID controller brings a different approach to managing the queue length. In the proposed control technique, the PID controller calculated L_q for every arriving packet at the cluster head node. When L_q is less than the minimum threshold no congestion occurs and thus packets keep arriving at the initial arrival rate (λ). However, if L_q is equal to or greater than the maximum threshold, congestion is detected. Instead of arriving packets being dropped, every arriving packet takes a reduced arrival rate (λ_{eff}) thus reducing the packet loss rate. Please see *Figure 4* and *Equation 2*. The Fuzzy Logic was introduced to address the limitation of the PID controller's low calculation accuracy. The Fuzzy-PID controller calculated the packet loss rate (P_d). If the packet loss is equal to or greater than 1, the system will measure the congestion level based on L_q. Firefly Algorithm (FA) was used because its subgroups the entire population (node) like a cluster-based wireless sensor network. This advantage was suitable for the cluster head system architecture used in the system model of the proposed control technique. The Firefly Algorithm was used to reduce the congestion level. The packet resumes entering the head node at the initial arrival rate when L_q is less than the minimum threshold following the congestion avoidance stage. The entire system model is shown in *Figure 4*. *Equation 1* provides the loss function for the proposed congestion control method.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

$$P_d = \begin{cases} 0, & L_q \le \min_{th} \\ \dfrac{\left(L_q - \min_{th}\right)^3}{\left(\max_{th} - \min_{th}\right)^3}, & \min_{th} \le L_q < \max_{th} \\ \ge 1, & L_q \ge \max_{th} \end{cases} \quad (1)$$

To compute the steady-state probability distribution, the arrival rate and service rate were assumed to be Poisson and exponential distributions respectively. The arrival rate is set to λ, and the service rate is set to μ. The state transition diagram is shown in *Figure 4*.

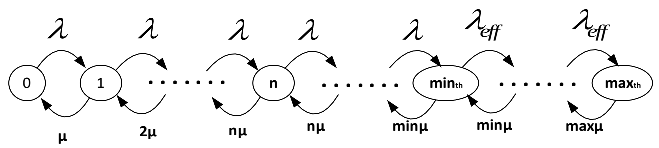


Figure 4: State transition diagram of the queuing model

The following probability expressions were obtained from the state transition diagram in *Figure 4*.

State 1: $P_1 = \rho P_0$

Stare 2: $P_2 = \dfrac{\rho P_0}{2} = \dfrac{\rho^2 P_0}{2!}$

State $\min_{th}$: $P_{\min_{th}} = \dfrac{\rho P_{\min_{th}-1}}{\min_{th}} = \dfrac{\rho^{\min_{th}}}{\min_{th}!} P_0$

State $\max_{th}$: $P_{\max_{th}} = \dfrac{\rho P_{\max_{th}-1}}{\max_{th}} = \dfrac{\rho^{\max_{th}}}{\left(\min_{th}!\min_{th}^{\max_{th}-\min_{th}}\right)} P_0$

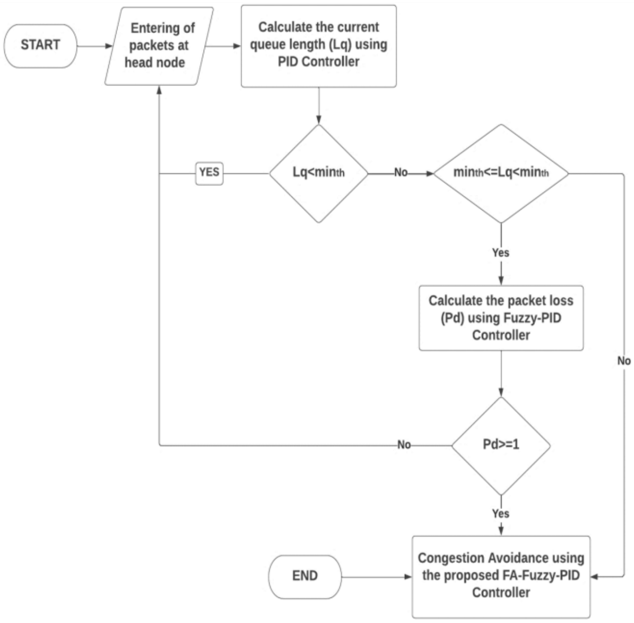


Figure 5: Flow chart of the proposed system model

From the probability expressions obtained from the state transition diagram, the probability of n^{th} packet in the queue P_n and the initial probability P_0 are given in *Equations (2) and (3)*, respectively.

$$P_n = \begin{cases} \rho P_0, & n = 1 \\ \dfrac{\rho^n P_0}{n!}, & 1 < n \le \min_{th} \\ \dfrac{\left(\dfrac{\lambda_{eff}}{\mu}\right)^{\min_{th}}}{\min_{th}!}\left(\dfrac{\dfrac{\lambda_{eff}}{\mu}}{\min_{th}}\right)^{n-\min_{th}}, & \min_{th} < n \le \max_{th} \end{cases} \quad (2)$$

$$P_0 = \left(\sum_{n=0}^{\min}\frac{\rho^n}{n!} + \frac{\rho^{\min_{th}}}{\min_{th}!}\sum_{n=\min_{th}}^{\max_{th}}\left(\frac{\dfrac{\lambda_{eff}}{\mu}}{\min_{th}}\right)^{n-\min_{th}}\right)^{-1} \quad (3)$$

Using Little's theorem, the average queue length (L_s) at the head node of each cluster is given in *Equation 4*.

$$L_s = \sum_{n=0}^{\max_{th}} nP_n \quad (4)$$

The average number of packets in the buffer (L_w) of the cluster head node is given in *Equation 5*.

$$L_w = L_s - \frac{\left(\lambda - \lambda_{eff}\right)}{\mu} \quad (5)$$

Because the queue model has a finite length, there is a reduction in the arrival rate. The reduced arrival rate is given in *Equation 6*.

$$\lambda_{eff} = \lambda(1 - P_{\max_{th}}) \quad (6)$$

The average time a packet spends in the queue (W_w) is given by *Equation 7*.

$$W_w = \frac{L_w}{\left(\lambda - \lambda_{eff}\right)} \quad (7)$$

The average time it takes a packet to leave the queue (W_s) is given by *Equation 8*.

$$W_s = \frac{L_s}{\left(\lambda - \lambda_{eff}\right)} \quad (8)$$

4. THE PROPOSED FA-FUZZY-PID CONTROLLER

The PID controller was used to calculate L_q using *Equation 9*. Due to the limitation of the PID controller, the fuzzy rules were added to provide a more precise result but there was still room for advancement. Although not ideal, the Fuzzy-PID calculated

Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

the packet loss using the loss function in *Equation 1*. This research used the Firefly Algorithm's effective optimization capability to reduce the queue length at the head node based on the result of the Fuzzy-PID controller, leading to a more precise final value and queue length control.

$$L_q = Y_{in} + P_k e + I_k T_s \sum_{n=0}^{min_{th}} \frac{\rho^{min_{th}}}{n+1} e + D_k \left(\frac{\rho^n}{T_s} \right) e^{-nP_0} \tag{9}$$

Where Y_{in} represents arriving packet, P_k is the proportional gain, I_k is the integral gain, D_k is the derivation gain, T_s is the sampling time, and e is the error. Y_{in} and e can be calculated as given in *Equations 10 and 11*, respectively.

$$Y_{in} = \sum_{n=1}^{max_{th}} \left(\frac{\rho^n - \lambda_{eff}}{n!} \right) \tag{10}$$

$$e = Y_{in} - Y_{out} \tag{11}$$

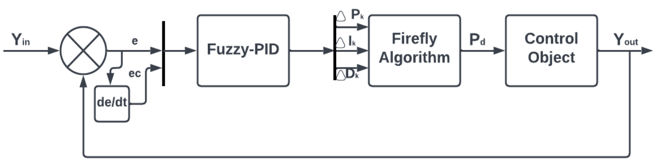


Figure 6: Firefly Algorithm-based Fuzzy-PID Model

As shown in *Figure 6*, the Firefly Algorithm and Firefly FuzzyPID controller were developed with the packet loss rate in the Fuzzy-PID model as the goal object.

4.1 Firefly Algorithm

An optimization technique called the Firefly Algorithm (FA) is based on the flashing behavior of fireflies and is inspired by nature. To tackle a continuous optimization problem, *Yang* [23] was the first to employ the Firefly Algorithm. The flashing light produced by bioluminescence can distinguish fireflies from other insects. FA has adjusted by discretizing to deal with the permutation problem. The FA follows the following three idealized rules [22]:

- Fireflies come in both genders.
- As the distance between them increases, their appeal, which is proportional to their brightness, decreases. The less dazzling of two flashing fireflies will move in the direction of the more brilliant one. If there is not another brighter firefly nearby, one will move randomly.
- The objective function determines a firefly's brightness (light intensity).

Attraction, randomization, and absorption are the parameters used in this technique to create the Firefly Algorithm.

- The light intensity disparity between two fireflies is a proxy for the attraction metric.
- If this parameter is set to zero, the random walk that the Gaussian distribution principal dictates corresponds to the randomization parameter and acts as though it were generating a value from an interval.
- In contrast, as the value of the absorption parameters shifts from zero to infinity, it impacts the attractiveness parameters value.

Although these advantages of FA were helpful in this research, there were limitations that were considered when using FA. These limitations are:

- *Local Optima*: Like many optimization algorithms, FA was prone to get stuck in local optima, which limited its ability to find the global optimum solution.
- Crossover and mutation: FA do not include the crossover and mutation operations commonly used in evolutionary algorithms, limiting its ability to explore the solution space effectively and improve the diversity of solutions.

The fluctuation in attraction with distance r is given in *Equation 12.*

$$\beta = \beta_0 e^{-\gamma r^2} \tag{12}$$

Where β_0 is the attractiveness at distance $r = 0$.

The Firefly Algorithm was updated and iterated using P_d as the loss function and the firefly motions as the search path to determine the greatest value of P_d. When $P_{d(i)} > P_{d(j)}$ was true, the updated position for the firefly i is attracted to a brighter firefly j. This movement is expressed in *Equation 13.*

$$P_{d(i)}(t+1) = P_{d(i)}(t) + \beta e^{-\gamma r^2} \left(P_{d(j)} - P_{d(i)} \right) + \alpha \varepsilon_{(i)} \tag{13}$$

In this case, $P_{d(i)}(t)$ represents the positional packet loss for the K^{th} update at the T^{th} iteration. Additionally, $\alpha \varepsilon_{(i)}$ represents the Steps Size which was a random value.

When $P_{d(i)} > P_{d(j)}$ was false, the firefly was moved randomly. The updated position is given in *Equation 14.*

$$P_{d(i)}(t+1) = P_{d(i)}(t) + \alpha \varepsilon_{(i)} \tag{14}$$

Firefly Algorithm (FA) has several advantages over other popular optimization algorithms such as Genetic Algorithm (GA), Cuckoo Search (CS), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). These advantages are:

- Subgroups are created for the entire population automatically
- The capacity to naturally handle multi-modal optimization
- Prominent convergence period

4.2 Fuzzy-PID Controller Design

The parameters are set in standard PID controller designs, but adaptive modification is absent [26]. This study uses fuzzy rules and control technology to modify the PID controller parameters in response to this problem. The fuzzy control can be

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

appropriately set for nonlinear and real-time control scenarios and does not require a precise mathematical description of the controlled item [20]. Fuzzy control is primarily concerned with fuzzy control objects and fuzzy reasoning, both based on fuzzy rules. This research employed rules and logic to incorporate and control the PID controller to raise the PID parameters and optimize the PID controller settings. The parameter self-tuning optimization function of PID was then implemented by weighting the basic parameters to produce new PID parameters. *Figure 7* shows the design of the Fuzzy-PID controller. Fuzzy-based PID controller was designed in MATLAB for the congestion control process. The researcher combined the triangle membership function with the trapezoid membership function to increase the control effect. The triangle and trapezoid membership function analytic function is represented in *Equations 15 and 16,* respectively.

$$\mu(x) = \left\{ \begin{array}{ll} \frac{x-a}{b-x}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \end{array} \right\} \tag{15}$$

$$f(x,a,b,c,d) = \max_{th}\left(\min_{th}\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-a}\right), 0\right) \tag{16}$$

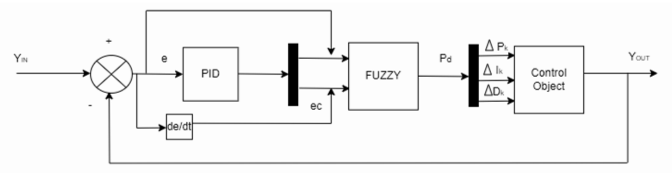
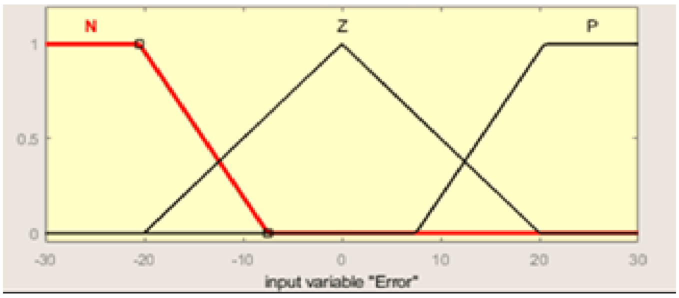


Figure 7: Fuzzy-PID Controller Model

4.2.1 Formulation of Fuzzy Rules

The researcher used the Fuzzy Logic rules to adjust the three PID parameters in the Fuzzy-PID controller. When error e and error deviation ec were fed into the FIS, the PID parameters K_p, K_i, and K_d were generated, with their beginning values being the same as those obtained using the congestion control technique. For each input and output, the following three membership functions were used: negative (N), zero (Z), and positive (P). A range of values for these functions was selected using the expertise of manual PID controller tuning. *Figures 8* and *9* demonstrate the input and output membership functions for the FIS, respectively. The Fuzzy rule is shown in *Table 1*.



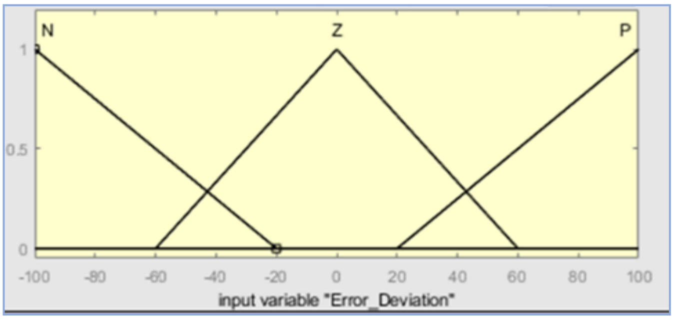


Figure 8: Input Member Function (Error and Error Deviation)

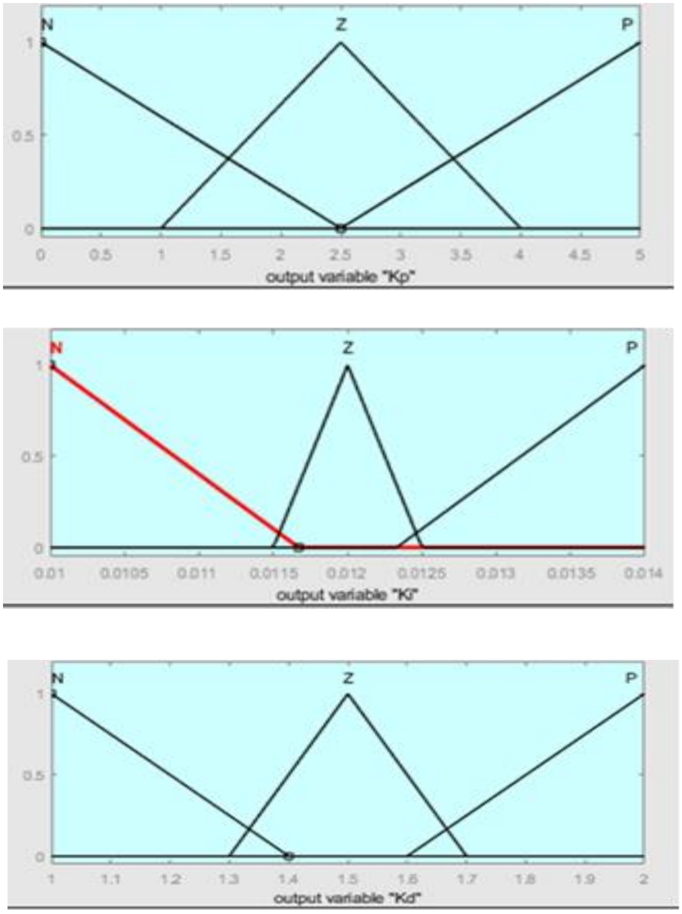


Figure 9: Output Member Function (K_p, K_i, and K_d)

- When e is large and ec is negative, the system response is still distant from the set point even though it is heading in the right direction. K_p needs to be big, while K_i and K_d need to be small for the set point to close in quickly.
- When e is negative and ec is large, the system's reaction has gone above the set point, and the error is growing. K_d has been set excessively high, whereas K_i and K_p have been set inadequately, to avoid overshoot.
- In a steady state, the system response is closed when e is small and positive. To quickly reach steady-state, K_p should be high; yet, to lower overshoot and prevent

Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

oscillation, K_d should be raised while K_i should be lowered.

- When e is large and ec positive, the system response overshoots to the negative side. K_d should be high, but K_p and K_i should also increase to reduce the error.

Table 1. Fuzzy Logic Rules

K_p/ K_i / K_d		e (-30, 30) N	e (-30, 30) Z	e (-30, 30) P
ec (-100, 100)	**N**	Z/N/N	Z/Z/Z	P/N/N
	Z	Z/Z/Z	Z/Z/Z	Z/Z/Z
	P	P/N/N	P/N/Z	Z/Z/P

4.3 Simulation and Analysis of FA-Fuzzy-PID Controller using NS-3

Network Simulator 3 (NS-3) Simulator is a discrete-event network simulator primarily intended for academic and research purposes [27][28]. NS-3 is built using C++ and Python to construct simulation models. It allows the use of several external animators, data analysis, and visualization tools. The researcher analyzed the proposed controller queue length, packet loss rate, congestion level, and throughput. The simulation parameters for the proposed control technique are given in *Table 2*.

Table 2. Simulation Parameter

Characteristics	PID Controller	Fuzzy-PID Controller	FA-FuzzyPID
Slew Rate	198.60	156.906	98.734
Overshoot	0.17059	0.00633	0
Undershoot	0.02329	0.01891	0
Rise Time	3.118 sec	0.0025 sec	0.001 sec

5. RESULTS AND DISCUSSION

This section summarizes the results and assesses the performance metrics obtained by performing the simulations.

5.1 Queue Length Analysis using FA-Fuzzy-PID

The simulation result in *Figure 10* exemplifies the actions and findings of the FA-Fuzzy-PID controller, the Fuzzy-PID Controller, and the PID controller. The results in *Table 3* show that the FA-Fuzzy-PID Controller performed better with less overshoot, slew rate, rise time and a positive undershoot value when controller-tuned parameters were compared based on rising time, overshoots, and slew rate. This result shows greater consistency than the standard PID and Fuzzy-PID controllers.

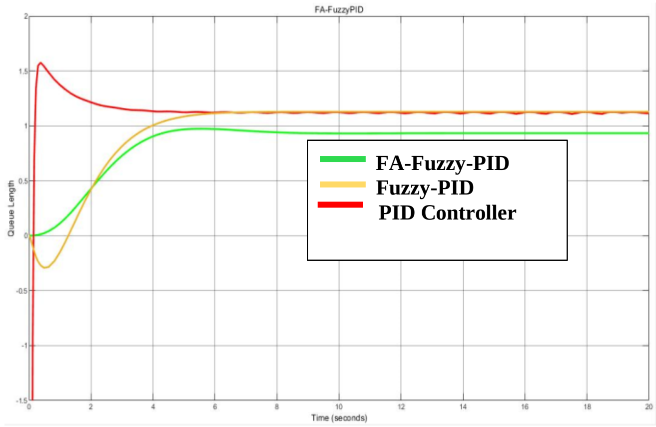


Figure 10: Simulation result of the proposed FA-Fuzzy-PID Controller

Table 3. Comparative analysis of the designed FA-Fuzzy-PID Controller

Simulation Parameter	Value
Simulating Area	1000*1000 m sq.
Number of nodes	100 nodes
Simulating Time	10 sec
Traffic Rate	100 Mbps
Traffic Source	Constant Bit Rate (CBR)
Transmission Speed	1.2 m/s

5.2 Performance Analysis of the Proposed Controller with Existing Techniques

The researcher used MATLAB R2020a simulator and NS-3 to simulate and analyze the FA-Fuzzy-PID controller, respectively. This section explains the network performance measures used to assess the performance of CFPID, PSOGSA, and the proposed FA-Fuzzy-PID:

- *Throughput*: This is the total number of packets receiving rates at the sink (in bits/s) divided by the sink's overall bandwidth (in bits/s).
- *Packet loss*: This is when the number of packets sent differs from the received packet in a network.
- *Queue size*: This is the total number of packets available in the queue at any time.
- *Congestion level*: This is the level at which congestion has occurred at the head node.

5.2.1 Performance Analysis of Throughput

Network throughput is the total number of packets receiving rates at the sink (in bits/s) divided by the sink's overall bandwidth (in bits/s). For improved network performance, the network throughput needs to be increased. The throughput-based analysis curve for the proposed rate optimization technique. The analysis curve for a sensor network with 100 nodes and a 10-second time instance is displayed in *Figure 11*, and the throughput values for the proposed and existing approaches at each second (time instance). The performance analysis finding reveals that the proposed technique improves with each subsequent time interval compared to CFPID and PSOGSA. In the analysis, CFPID and PSOGSA obtained throughputs of 0.3877 and 0.4697, respectively, while the proposed technique throughput at the sixth occurrence (sec) is 0.7079.

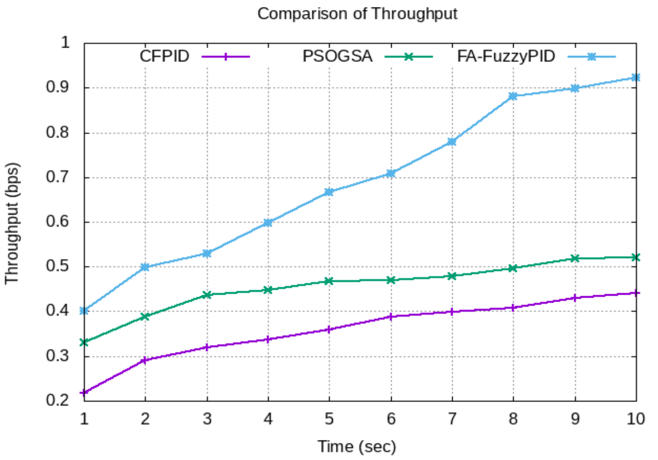


Figure 11: Comparison of Throughput

5.2.2 *Performance Analysis of Packet Loss Rate*

The packet loss rate is when the number of packets sent differs from the received packet in a network. *Figure 12* depicts the assessment curve based on the packet loss rate. For the network to run more efficiently, there must be less packet loss during transmission. *Figure 12* depicts the analytical curve that links time occurrences of packet loss for the network with 100 nodes. The number of incoming data packets increases with each subsequent time event, increasing packet loss. The proposed technique offers the lowest packet loss compared to the existing system even for more significant instances with higher packet arrival. In comparison to the existing system's 9th occurrence (sec) packet loss for the CFPID and PSOGSA systems, the proposed system's value is 0.3972, while the existing system's values are 0.5697 and 0.4872. Based on the analysis curve, the performance of the proposed FA-Fuzzy-PID system exhibits fewer packet losses.

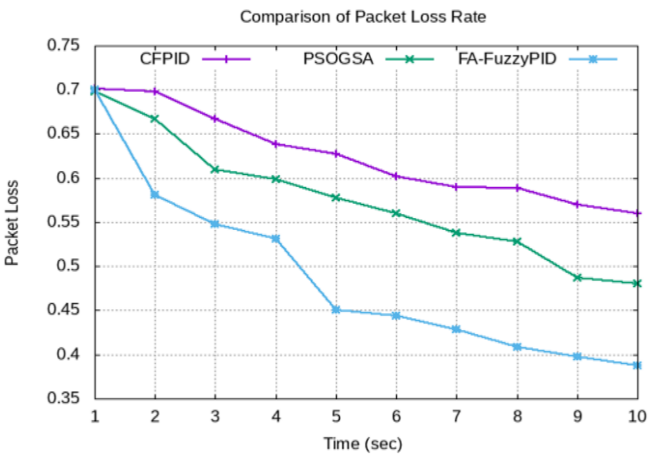


Figure 12: Comparison of Packet Loss Rate

5.2.3 *Performance Analysis of Queue Length*

Queue Length is the total number of packets available in the queue at any time. For a WSN to work at its best, the queue size should be as minimal as possible. *Figure 13* shows the evaluation curve based on queue size for the systems employed in the experiment. For a 100 nodes network, this figure provides the analysis curve between the queue size and the time instance. In contrast to the system CFPID used for the experimental evaluation, which achieved a queue size of 0.0721 for time instance (sec) 8, the FA-FuzzyPID achieved a queue size of 0.01897. While the CFPID queue size is 0.0800 and the PSOGSA queue size is 0.0416 at time instance 10 (10th sec), the FA-Fuzzy-PID queue size is less than 0.03. The proposed system is effective due to the reduction in its queue size.

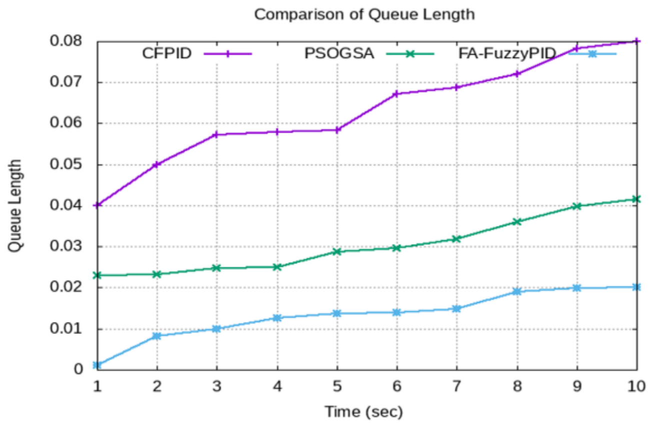


Figure 13: Comparison of Queue Length

5.2.4 *Performance Analysis of Congestion Level*

Congestion Level is the level at which congestion has occurred at the cluster head node. *Figure 14* displays the analysis curve for evaluating the proposed method according to the degree of congestion. The figure below shows the network congestion level analysis curve at 100-node. The existing system CFPID has a congestion level at instance (sec) 10 of 0.2, which is 0.15 higher than that of the proposed FA-Fuzzy-PID system, and instance (sec) 10 of PSOGSA has a congestion level at instance (sec) 10 of 0.1, which is 0.05 higher than the proposed FA-Fuzzy-PID system. The proposed FA-Fuzzy-PID system in this figure has a congestion level at instance (sec) 10 of 0.05.

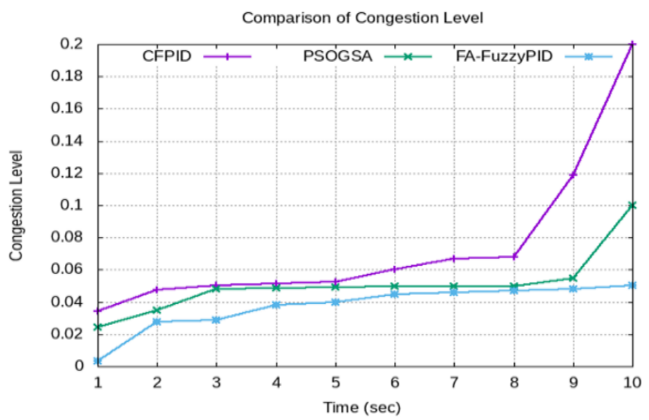


Figure 14: Comparison of Congestion Level

5.2.5 *Summary of Performance Analysis*

The performance analysis evaluation summary is shown in *Figure 15*. Values in this figure correspond to the parameters gathered after 10 seconds of simulation time. According to the figure, the proposed FA-Fuzzy-PID improved performance across the board. For instance, the current CFPID only yielded a result of 0.4400 throughputs whereas the throughput of the FA-Fuzzy-PID technique is 0.8998. The congestion level is lowest when the proposed FA-Fuzzy-PID technique was compared to the existing systems. The proposed technique has improved by reducing the packet loss rate with each iteration.

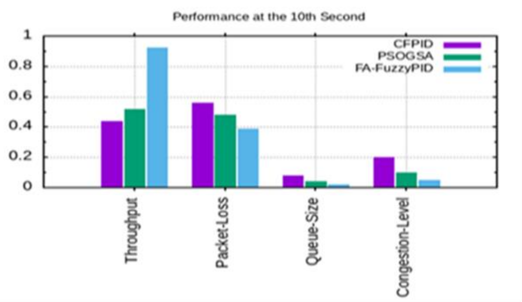


Figure 15: Summary of Performance Analysis

The actual value gleaned from the evaluation analysis is presented in *Table 4*. The values acquired after 10 sec of simulation time are those reported in *Table 4*, representing a fraction of the simulation time.

Table 4. Performance Analysis summary value with existing techniques

Performance Metrics	CFPID [10]	PSOGSA [17]	Proposed FA-Fuzzy-PID
Throughput	0.4400	0.5200	**0.9200**
Packet Loss Rate	0.5599	0.4799	**0.3875**
Queue Size	0.0800	0.0416	**0.0200**
Congestion Level	0.2000	0.1000	**0.0500**

6. CONCLUSION

This paper provides a novel and effective method to control congestion in wireless sensor networks based on Firefly optimized Fuzzy-PID controller. The Firefly Algorithm was applied to optimize the PID parameter increment from the Fuzzy-PID controller to provide a more exact and accurate result. The aim was to allow a regular data flow and improve the overall performance of WSNs allowing end-users to experience faster data transfer speeds and fewer network delays. The simulation results have proven that the proposed control method is efficient in controlling congestion in wireless sensor networks.

REFERENCES

[1] S. Srinivasa Rao, K. C. Keshava Reddy, and S. Ravi Chand, "A Novel Optimization based Energy Efficient and Secured Routing Scheme using SRFIS-CWOSRR for Wireless Sensor Networks," Int. J. Electr. Electron. Res., vol. 10, no. 3, pp. 644–650, 2022, doi: 10.37391/IJEER.100338.
[2] E. H. Kim, "Fan-Shaped Flooding in Wireless Sensor Networks," Int. J. Electr. Electron. Res., vol. 10, no. 2, pp. 225–229, 2022, doi: 10.37391/IJEER.100230.
[3] R. Bhaskaran, K. Ramamoorthy, C. Fancy, and T. Jayasankar, "Replica Node Detection using Metaheuristic Algorithms in Wireless Sensor Networks," Int. J. Eng. Trends Technol., vol. 70, no. 5, pp. 339–345, 2022, doi: 10.14445/22315381/IJETT-V70I5P237.
[4] B. Guruprakash, C. Balasubramanian, and R. Sukumar, "An approach by adopting multi-objective clustering and data collection along with node sleep scheduling for energy efficient and delay aware WSN," Peer-to-Peer Netw. Appl., vol. 13, no. 1, pp. 304–319, 2020, doi: 10.1007/s12083-019-00779-3.
[5] V. Srivastava, S. Tripathi, K. Singh, and L. H. Son, "Energy efficient optimized rate based congestion control routing in wireless sensor network," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 3, pp. 1325–1338, 2020, doi: 10.1007/s12652-019-01449-1.
[6] L. Jenila and R. A. Canessane, "Cross Layer Based Dynamic Traffic Scheduling Algorithm for Wireless Multimedia Sensor Network," Int. J. Electr. Electron. Res., vol. 10, no. 2, pp. 399–404, 2022, doi: 10.37391/IJEER.100256.
[7] S. Panimalar and T. P. Jacob, "A Comparative Study of Hybrid Optimized Algorithms for Congestion Control in Wireless Sensor Network," Proc. 2020 IEEE Int. Conf. Commun. Signal Process. ICCSP 2020, pp. 889–893, 2020, doi: 10.1109/ICCSP48568.2020.9182194.
[8] S. Qu, L. Zhao, and Z. Xiong, "Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control," Neural Comput. Appl., vol. 32, no. 17, pp. 13505–13520, 2020, doi: 10.1007/s00521-020-04758-1.
[9] M. Zareei, C. Vargas-Rosales, R. Villalpando-Hernandez, L. Azpilicueta, M. H. Anisi, and M. H. Rehmani, "The effects of an Adaptive and Distributed Transmission Power Control on the performance of energy harvesting sensor networks," Comput. Networks, vol. 137, pp. 69–82, 2018, doi: 10.1016/j.comnet.2018.03.016.
[10] L. Lin, Y. Shi, J. Chen, and S. Ali, "A novel fuzzy PID congestion control model based on cuckoo search in WSNs," Sensors (Switzerland), vol. 20, no. 7, 2020, doi: 10.3390/s20071862.
[11] L. Tan, Q. Liu, and S. H. Yang, "Congestion Control Of High Speed Computer Networks : A Pid Method," vol. 86, no. 0.
[12] H. O. Bansal, R. Sharma, and P. R. Shreeraman, "PID Controller Tuning Techniques : A Review," vol. 2, no. October, pp. 168–176, 2012.
[13] Y. Shatnawi, "Congestion Control in ATM networks using PID Controller with Immune Algorithm," 2019 10th Int. Conf. Inf. Commun. Syst., pp. 19–24, 2019.
[14] S. Jaiswal and A. Yadav, "Fuzzy based adaptive congestion control in wireless sensor networks," 2013 6th Int. Conf. Contemp. Comput. IC3 2013, no. August 2013, pp. 433–438, 2013, doi: 10.1109/IC3.2013.6612234.
[15] M. Zarei, A. M. Rahmani, R. Farazkish, and S. Zahirnia, "FCCTF: Fairness Congestion Control for a distrustful wireless sensor network using Fuzzy logic," 2010 10th Int. Conf. Hybrid Intell. Syst. HIS 2010, no. September, pp. 1–6, 2010, doi: 10.1109/HIS.2010.5601071.
[16] J. V. Chen, F. C. Chen, J. M. Tarn, and D. C. Yen, "Improving network congestion: A RED-based FuzzyPID approach," Comput. Stand. Interfaces, vol. 34, no. 5, pp. 426–438, 2012, doi: 10.1016/j.csi.2012.02.002.
[17] T. S. Yuslinda and W. Mohamad, "Congestion control in wireless sensor network using fairness bandwidth allocation / Husna Zainol Abidin , Yuslind ...," doi: 10.1016/j.comnet.2018.03.023.
[18] A. A. Rezaee and F. Pasandideh, "A Fuzzy Congestion Control Protocol Based on Active Queue Management in Wireless Sensor Networks with Medical Applications," Wirel. Pers. Commun., vol. 98, no. 1, pp. 815–842, 2018, doi: 10.1007/s11277-017-4896-6.
[19] X. Yang, X. Chen, R. Xia, and Z. Qian, "Wireless sensor network congestion control based on standard particle swarm optimization and single neuron PID," Sensors (Switzerland), vol. 18, no. 4, 2018, doi: 10.3390/s18041265.
[20] R. Rajesh, "Energy-Resourceful Routing by Fuzzy Based Secured CH Clustering for Smart Dust," Int. J. Electr. Electron. Res., vol. 10, no. 3, pp. 659–663, 2022, doi: 10.37391/IJEER.100340.
[21] A. Sujith, V. N. Kamalesh, H. P. Srinivasa, and S. Suresh, "Energy-Efficient Adaptive Routing Algorithm Based on Fuzzy Inference System using Zone-Based Clustering of Wireless Sensor Network," Int. J. Eng. Trends Technol., vol. 70, no. 6, pp. 221–236, 2022, doi: 10.14445/22315381/IJETT-V70I6P224.
[22] M. S. Manshahia, M. Dave, and S. B. Singh, "Firefly algorithm based clustering technique for Wireless Sensor Networks," Proc. 2016 IEEE Int.

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 1 | Pages 44-53 | e-ISSN: 2347-470X

Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016, pp. 1273–1276, 2016, doi: 10.1109/WiSPNET.2016.7566341.

[23] X. S. Yang, Cuckoo search and firefly algorithm: Overview and analysis, vol. 516. 2014.

[24] X. S. Yang, "Firefly algorithms for multimodal optimization," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5792 LNCS, pp. 169–178, 2009, doi: 10.1007/978-3-642-04944-6_14.

[25] G. Linta Salvin and J. Arul Linsely (2022), Full-duplex QoS Optimization using Enhanced firefly Algorithm. IJEER 10(3), 585-589. DOI: 10.37391/IJEER.100329.

[26] U. R. S. Yalavarthy and V. S. K. R. Gadi, "PEM Fuel Cell Powered Electric Vehicle Propelled by PMSM Using Fuzzy PID Controller-A Research," Int. J. Eng. Trends Technol., vol. 70, no. 1, pp. 63–74, 2022, doi: 10.14445/22315381/IJETT-V70I1P208.

[27] M. M. Babu, R. P. Sam, and P. C. Reddy, "A3C Based Dynamic BitRate for Video Streaming in 5G Edge Assisted D2D Communication Using H.266 With Conv-DBN," Int. J. Eng. Trends Technol., vol. 70, no. 1, pp. 93–107, 2022, doi: 10.14445/22315381/IJETT-V70I1P211.

[28] S. Silmi, Z. Doukha, R. Kemcha, and S. Moussaoui, "Wireless Sensor Networks Simulators and Testbeds," pp. 141–159, 2020, doi: 10.5121/csit.2020.100912.