

Research Article

Traffic and Energy Aware Optimization for Congestion Control in Next Generation Wireless Sensor Networks

Saneh Lata Yadav,¹ R. L. Ujjwal,¹ Sushil Kumar ,² Omprakash Kaiwartya ,³ Manoj Kumar,² and Pankaj Kumar Kashyap²

¹School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, New Delhi 110078, India

²School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

³Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK

Correspondence should be addressed to Omprakash Kaiwartya; omprakash.kaiwartya@ntu.ac.uk

Received 20 March 2021; Revised 30 May 2021; Accepted 9 June 2021; Published 30 June 2021

Academic Editor: Xavier Vilanova

Copyright © 2021 Saneh Lata Yadav et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Congestion in wireless sensor networks (WSNs) is an unavoidable issue in today's scenario, where data traffic increased to its aggregated capacity of the channel. The consequence of this turns in to overflowing of the buffer at each receiving sensor nodes which ultimately drops the packets, reduces the packet delivery ratio, and degrades throughput of the network, since retransmission of every unacknowledged packet is not an optimized solution in terms of energy for resource-restricted sensor nodes. Routing is one of the most preferred approaches for minimizing the energy consumption of nodes and enhancing the throughput in WSNs, since the routing problem has been proved to be an NP-hard and it has been realized that a heuristic-based approach provides better performance than their traditional counterparts. To tackle all the mentioned issues, this paper proposes an efficient congestion avoidance approach using Huffman coding algorithm and ant colony optimization (ECA-HA) to improve the network performance. This approach is a combination of traffic-oriented and resource-oriented optimization. Specially, ant colony optimization has been employed to find multiple congestion-free alternate paths. The forward ant constructs multiple congestion-free paths from source to sink node, and backward ant ensures about the successful creation of paths moving from sink to source node, considering energy of the link, packet loss rate, and congestion level. Huffman coding considers the packet loss rate on different alternate paths discovered by ant colony optimization for selection of an optimal path. Finally, the simulation result presents that the proposed approach outperforms the state of the art approaches in terms of average energy consumption, delay, and throughput and packet delivery ratio.

1. Introduction

The advancement in low-cost, small, and tiny sensor nodes makes a significant role where the sensor nodes have very attractive characteristics of sensing the environmental conditions and process the received signals. The sensor nodes can be deployed in all accessible and inaccessible areas for sensing the data across various applications like battlefield, building inspection, target field imaging greenhouse, and monitoring disaster area [1]. The deployment of sensor nodes is application-dependent, so it can be random or deterministic [2]. The sensor nodes of the wireless sensor network (WSN) sense the data or event, gather the data

under defined infrastructure, and process the received signals. After a lot of advancement in wireless sensor networks, it is lacking with few specifications like limited memory, inadequate computation, limited bandwidth, and battery-powered nodes [3, 4]. Due to the short communication range of sensor nodes, the intermediate nodes collaborate in forwarding the data packets. There are various applications of WSN where sensor nodes are deployed in an infrastructure-less network. The sensor nodes sense an event and report to the nearest base station for respective action. To obtain the quality of service such as congestion-free end-to-end data delivery and delay-free data transmission, an effective and efficient network is required to design to

tackle congestion and energy issues of WSN [5, 6]. When a node receives data more than its capacity to process, congestion may occur which leads to the retransmission of unacknowledged packets. But this frequent retransmission of packets may deplete the energy of the nodes. The nodes in WSN are battery-powered, so energy efficient decisions are desired solution to maximize the lifetime of the network [7]. Load balancing, duty cycling, and data aggregation are the various traditional approaches that have been studied and implemented on WSN. But due to the rapid increase in sensor nodes, the traditional approach faces many issues. Nowadays, trajectory-based data forwarding, mobile sink, and energy supply based on node selection approaches are used for residual energy consumption [8, 9].

WSN is bounded with limited transmission and processing capabilities; therefore, the energy of sensor nodes and successful data transmission are the two important and mandatory requirements. There are four popular applications of WSN such as event-oriented, query-oriented, continuous, and hybrid applications. The application performs on few pre-defined situations, resulting in an unforeseeable traffic rate. The applications where one sensor node transmits a query and another sensor node has to respond to it are known as query-oriented applications. On the other side, continuous applications perform periodically or in few time slots. Lastly, the hybrid applications are the fusion of the above three applications [6]. The transport layer deals with the congestion problem, which is accountable for end-to-end network connection. The objective of the transport layer is to provide fair bandwidth allocation, control the data flow rate with reliable connection links, and retransmit lost packets in an energy-efficient fashion. Few existing strategies for controlling congestion are a slow start, congestion avoidance, fast retransmission, and fast resumption, but these are not able to outperform in today's network scenario [10]. Therefore, an effective and efficient CC mechanism under few constraints, such as resource, performance, and scalability, is in high demand. Congestion is noticed on two locations, namely, node or link. When the node's buffer is fully occupied owing to the fast arrival rate of incoming packets, node-level congestion happens. When data packets collapse on network/radio links owing to the fast transmission of data packets, link-level congestion is noticed. The performance is influenced by both types of congestion. Congestion should be detected at its early stage, then be notified to the nearby nodes to the source node, and eventually, control it [11]. Traffic control and resource control are the two most popular approaches for this domain. When the data flow rate is controlled, traffic controlled approach is considered, while the resource control approach looks after the fair allocation of resources to divert the data on a less congested path [12]. Both the approaches contemplate the node's priority, which plays the foremost role in dealing with congestion. To address the abovementioned challenge, we present an energy-efficient, robust, and heuristic-based approach to boost the lifetime of the network, provided congestion-free. Therefore, major research contributions are as follows:

- (1) An efficient congestion avoidance approach using Huffman coding enabled ant colony optimization

(ECA-HA) in wireless sensor networks is proposed considering the important constraints of WSN such as energy, packet loss rate, and congestion level

- (2) Ant colony optimization has been applied to search congestion-free alternate paths whenever there is a hike in congestion on the current path. The forward ant constructs multiple congestion-free paths from source to sink node, and backward ant ensures about the successful creation of paths moving from sink to source node, considering energy of the link, packet loss rate, and congestion level
- (3) Packet loss probability is computed for each path using maxima entropy principle. Packet entropy helps in evaluating the uncertainty of congestion degrees on alternate paths considering packet loss rate
- (4) Huffman coding is used to select an optimal path to direct the load. The optimal path is the best path among multiple paths identified by ACO in terms of energy, congestion level, and packet loss rate. Huffman coding considers the packet loss probability on different alternate paths discovered by ant colony optimization for selection of an optimal path
- (5) Finally, performance of the proposed (ECA-HA) approach is compared with the state of the art approaches

The rest of the paper is organized as follows: in Section 2, literature review related to congestion avoidance and control is discussed. The efficient congestion avoidance approach is elaborated in Section 3 that includes problem statement and congestion indicator model followed with Huffman coding enabled ant colony optimization concepts. The simulation and results of proposed mechanism (ECA-HA) are explained in Section 4. Section 5 elaborates the conclusion of the proposed mechanism.

2. Literature Review

This section explains the literature survey of different research works concerned with congestion avoidance and congestion control algorithm.

2.1. Nonnature-Inspired Techniques. This proposed work is aimed at avoiding congestion-like situations in WSN which is a very challenging issue. It directly affects the throughput of the network. It broadly occurs when the sensor nodes in WSN send data to a link more than its capacity to process. So, when these nodes accept data packets with a higher rate than their capacity, then the data packets may get dropped in between the path or face delay. If the packets acknowledged are not received, the packet is considered as dropped or lost. Such an unacknowledged packet requires retransmission. But this retransmission of lost/drop packet has a limit. Retransmission of the lost packet again and again may degrade the network performance. So, to tackle such situations, various algorithms have already been proposed. These

algorithms are not suitable for all WSN applications. CODA (congestion detection and avoidance) algorithm is proposed by Wan et al. The mechanism works on three broad steps: congestion detection, congestion notification, and congestion control. Congestion detection is done using buffer occupancy and channel status. This is a mechanism to detect a faulty situation before their occurrence. After detecting congestion, a notification can be done by the backpressure method. It applies hop by hop backpressure notification. This notification can be implicit or explicit. Both notifications have its advantages and disadvantage. To notify, it sends a beacon message implicitly or explicitly to the backward node, which increases overhead on an already congested link. For controlling purpose, AIMD (additive increase multiplicative decrease) is used, which additively increases and decreases the data rate [13]. CCF (congestion control and fairness) is proposed by Brahma et al. for many to one routing to control congestion and a transparent end-to-end packet delivery. Fairness is evaluated based on the ratio of the number of packets transmitted and the number of packets received [14]. Kasyap and Kumar proposed Trickle which is a self-regulating algorithm for code propagation and maintenance. It broadcast updated data information to disseminate any sort of data [15]. PCCP (priority-based congestion control) algorithm is proposed by Wang et al., which prioritizes the nodes for critical situation. This method uses a parameter congestion degree by evaluating the interarrival time and service time of packets. In any critical situation, prioritized packet may process first then other packets, which is aimed at maintaining the overall performance of the network [16]. CAF (congestion avoidance and fairness) proposed by Ahmad and Turgut calculates the characteristic ratio of the number of downstream nodes and upstream nodes. It monitors buffer occupancy of downstream nodes to avoid congestion. It fairly balances the load by monitoring buffer occupancy levels [17]. DALPaS (dynamic alternate path selection algorithm) proposed by Sergiou and Vassiliou measures congestion by the increased capacity of the link and opts alternate congestion-free path to flow the data rate. Various optimization techniques were also introduced for congestion avoidance and control [18].

2.2. Nature-Inspired Techniques. All the above approaches lead to excessive communication load and battery power depletion. These traditional approaches are not suitable for complex problems and do not provide the optimal solution. These drawbacks cannot be recovered easily in the communication networks. These are not favorable for a dynamic and unpredictable environment of WSN applications. So, it is a need of time to introduce self-adaptive approach. Self-adaptation is the strength of nature-inspired techniques. Nature-inspired techniques are very attractive to compute issues that rose in WSN. Nature-inspired techniques are performing the complex tasks in a seemly manner with limiting resources and capability. These techniques take advantage of various disciplines for computing complex tasks with significant features of emerging fields. These algorithms are developed by drawing inspiration from nature, social, and local behavior leading to emergent global behavior of reducing

congestion. These take knowledge from various branches of science like chemistry, mathematics, physics, biology, and engineering which helps in developing computation tools for complex problems. Nature-inspired approaches have opted when a problem is nonlinear and complex, with the huge number of potential solutions and objectives.

Based on produced solutions, optimization problems are divided into two categories such as deterministic and non-deterministic (stochastic) algorithms. Deterministic algorithms are conventional and classical algorithms that are based on mathematical programming like linear or nonlinear programming, quadratic programming, gradient-based, or gradient-free methods. Nondeterministic algorithms exhibit some randomness and produce variant results for different problems. Stochastic algorithms explore different search spaces to obtain global optimum escaping local optima. It is more capable of handling NP-hard problems. Optimization techniques like ACS (adaptive cuckoo search), bird flock behavior, artificial honey bee, fireflies and particle swarm intelligence, honey pot, and gravitational search algorithm, and ant colony optimization are some nature-inspired approaches. These all nature-inspired optimization techniques help in providing the optimum solution to control congestion to maintain performance and throughput of WSN. The adaptive cuckoo search uses cuckoo behavior for an optimal rate adjustment to minimize congestion. It adjusts the share rate within the limits of the packet service rate. It uses a fitness function for share rate, provided the congestion of minimum drop at the congested link. This method maximizes the system performance [19–21]. Flock-CC is based on bird behavior proposed by Antoniou et al. This approach adopts a swarm intelligence paradigm where packets are guided to form a flock and flow towards the sink like a global attractor. It dynamically balances the loads within available network resources. The method provides graceful performance in terms of packet delivery ratio, packet loss rate, and delay. This approach is very scalable and can be adopted in the complex networks [22]. REFIACC proposed by Kafi et al. is reliable, efficient, fair, and interference-aware congestion control protocol which schedules the communication to prevent interference in inter and intrapath hot spots. Maximum utilization of available bandwidth is done which outperforms the overall performance [23]. PSO (particle swarm intelligence) is inspired by fish schooling and bird swarm intelligence. Each particle flows in some direction by learning from its own experience or experience of its companion. All individual particle moves cooperatively and competitively in search space synchronously to obtain the best solution [24, 25]. Every particle focuses to move on the best neighbor node on the path. The particle moves from one location or node to another best possible node with a regulating velocity. It discovers p best and g best solution. Finally, fitness function value for each node is obtained. Further, GSA is used in collaboration with PSO, where best performer nodes are attracted towards sink nodes. Heuristic and metaheuristic approaches are mainly two derivative-free stochastic optimization algorithms. Metaheuristic approaches provide better results for complex problems as compared to the heuristic approach. So, the work goes

towards a metaheuristic approach because it provides an optimal solution after considering various concerned parameters. It discovers results using the trial and error phenomenon. It includes a group of search agents that explore the most feasible output depending on randomization and few specific rules. These rules are nature-inspired [26, 27]. Lee and Teng proposed an improved version for LEACH clustering protocol to reduce packet loss rate as well as to prolong the network lifetime using fuzzy inference systems. The approach is named as enhanced hierarchical clustering approach. The nodes with higher residual energy, slower moving rate, and longer pause time would be selected as cluster head [28]. To conserve energy and packet loss in WSN, El Alami and Nagid work on routing techniques considering the mobility of sink as the significant challenge for packet loss rate. The technique named routing-Gi proves itself as more efficient than the existing techniques in terms of lifetime of network, energy efficiency, and packet delivery rate [29]. El Alami and Nagid further adopt an enhanced clustering hierarchy approach to maximize the network lifetime, where the nodes with higher energy will gather data to transmit it to the base station. This approach resolves the issue of redundant data collection by nodes also. It works in a sleeping-waking mechanism that conserves the energy consumption. This approach is utilized in homogeneous and heterogeneous networks [30]. Sangeetha et al. introduced a heuristic approach for searching an energy efficient optimal path in WSN. The author focused on adjusting node degree and topology periodically to save battery power. After this, the data flow is balanced using fuzzy logic to avoid congestion. If congestion has occurred, it finds an alternate best path using learning real-time A star heuristic algorithm [31]. Logambigai et al. continue to work on heuristic approaches, and another energy efficient grid-based clustering approach was introduced with intelligent fuzzy rules. In this method, the routing is performed using grid coordinator with fuzzy rules considering minimum intermediate nodes in routing process. This approach is good in terms of lifetime of network and energy [32]. One more improved form of congestion aware routing mechanism using fuzzy rule sets is proposed by Sangeetha et al. This is a traffic prudent method which identifies more reliable path and handles excessive traffic using fuzzy rule prediction. It works in two segments; one is to identify path through positioning nonlocalized nodes, and another is to identify routes to mitigate congestion free path. The congestion estimate is done through ECFM algorithm [33]. One more clustered gravitational and fuzzy-based energy efficient approach was proposed by Selvi et al. to address the limitations and challenges of existing routing system. It utilizes a heuristic gravitational clustering approach to provide an optimal solution for effective routing and efficient clustering. The most appropriate route with cluster head nodes is chosen after applying fuzzy rule sets. The approach increases the lifetime of the network and reduces the consumption of energy [34]. ACO is a nature-inspired metaheuristic mechanism carries few inherent features that manifest excellent scaling characteristics. Metaheuristic techniques outperform to mitigate congestion and significant for large-scale networks as well. The objective of ACO is to discover congestion-free

alternate paths. ACO prospects alternate paths by generating and forwarding artificial ants on search space. Like real ants' budge on search space in search of food, they budge one after another perceiving the existence of pheromone. The probabilistic movement from one node to another allows the ants to prospect new and safe paths. The density of pheromone attracts other ants to budge on the best possible path. Also, heuristic values of different paths incorporate a vital role in adopting the most appropriate path [35–39]. The heuristic value determines local information about the path which is cooperating in evaluating congestion degree. The data rate on different routes is uncertain or random. So information entropy or packet entropy is measured here along with buffer entropy, to calculate the packet loss rate. The packet loss rate helps measure congestion degree and move towards the best solution. The entropy function is used to evaluate the probability of congested paths via packet loss probability on different paths. This indicates the uncertainty of the flow of data, which can be the cause of congestion. Entropy is a concept used to measure the disorder, uncertainty, or randomness of a system. Information theory introduced entropy. The concept of entropy was first introduced for statistical thermodynamics. But now, its applications are communication networks, biological research, and many more. In communication networks, it is widely used to measure the abnormality degree of an event by monitoring abnormal events which helps improve performance metrics of the network like throughput and energy efficiency. Entropy is mainly studied in three categories such as Shannon entropy, Renyi entropy, and Tsallis entropy. Most of the studies discussed only Shannon and Tsallis entropies. Shannon entropy is discovered by C. E. Shannon, and it is extensive, while Tsallis entropy is nonextensive. Shannon entropy is a quantitative measure of uncertainty in a data set. Tsallis entropy explores problems with multifractal structure for long-range dependence. It focuses on the effectiveness of entropy in controlling congestion. Shannon's work is extended by Jaynes named maximum entropy principle which has the inherent property to optimize entropy measure when incomplete information is provided in moment constraint form [40–44]. Shannon expressions and their relevance in the proposed technique are well described in the congestion indicator model section. Table 1 includes the notations used in this paper.

3. Proposed Efficient Congestion Avoidance Approach:(ECA-HA)

In this section, the problem statement and congestion indicator model are presented to elaborate the congestion issue and behavior of buffer occupancy. Alternate congestion-free path selection using ACO and Huffman coding is proposed.

3.1. Problem Statement. For monitoring purposes, N sensor nodes are deployed in a network size A . The WSN is represented by a connecting graph $G = \{V, E\}$ where $V = \{v_1, v_2, v_3, v_4, v_5, v_6 \dots \dots v_n\}$ a set of vertices and $E = \{e_1, e_2, e_3, e_4, e_5, e_6 \dots \dots e_n\}$ a set of connecting edges between sensor nodes. V_n node is considered as sink node and has an adequate energy resource as shown in Figure 1. All the

TABLE 1: Notations and their description.

Notation	Description
E_A	Average residual energy
E_i	Initial energy
E_T	Energy consumed in transmitting
E_R	Energy consumed in receiving
D_i	Delay
E_{amp}	Amplifying energy
d	Distance from source to base station
$E(\text{link})$	Energy consumed on link
$h(\text{link})$	Intermediate node count on link or hop count
N_L	Lifetime of link
N_{lifetime}	Lifetime of complete network
D_{pkt}	Data packet
λ_A	Data arrival rate
λ_D	Data service rate/departure rate
BO_V	Virtual buffer occupancy
BO_{Max}	Maximum buffer size
L_C	Congestion level
thr_{max}	Maximum threshold of buffer
TTL	Time to live
$P_{ij}^{(t)}$	Probability of selecting next hop
A_{ij}	Pheromone density
B_{ij}	Heuristic value
R_{ij}	Updated pheromone value
M_k	Memory to store visited nodes
W_{ij}	Load to distance ratio
E_V	Evaporation factor
Li	Load on i node
d_{ij}	Distance from i node to j node
M_p	Message generated per unit time
$P_L(N)$	Packet loss probability

nodes are static once deployed in the network. The initial energy and communication range of all the nodes are equal. The nodes within each other's communication range are assumed to be adjacent nodes. E_i is the initial energy of all sensor nodes where $i = 1, 2, 3 \dots n$. E_A is the average residual energy. $(E_A - E_i) = 0$, D_i is the delay, and $h(\text{link})$ is the hop count. When V_i a source node transmits data packets to V_n (sink node), the energy level is decreased in each transmission and receiving of the data packet. E_T and E_R are the energy consumed in transmitting and receiving k bits of information from V_i to V_n .

$$E_T(V_i, V_n) = k(E_d + E_{\text{amp}} * d^2), \quad (1)$$

$$E_R(V_i, V_n) = k(E_d). \quad (2)$$

E_T and E_R are the energy consumed in transmitting and receiving one bit. E_{amp} is the amplifying energy, and E_d is the energy dissipation per bit in transmitting and receiving. Distance between V_i and V_n is d . Total energy consumed on the link is the sum of energy consumed in transmitting and receiving.

$$E(\text{link}) = \sum_1^{n-1} (E_T + E_R). \quad (3)$$

This can be simplified as energy consumed in transmission and receiving of k bit data, multiplied by several intermediate nodes including the source node. Suppose v_1 as source node and $v_2, v_3, v_4 \dots v_{n-1}$ are intermediate nodes before reaching V_n . The node V_1 to V_{n-1} be intermediate nodes between V_i and V_n .

$$(v_1, v_2, v_3, v_4 \dots v_{n-1}) \in h(\text{link}), \text{ where } h(\text{link}) \text{ is the hop count of the link.} \quad (4)$$

We define link lifetime (N_L) using the minimum residual energy as

$$N_L = \frac{E_{\min}(\text{link})}{(E_T + E_R)}. \quad (5)$$

When one node is depleted its all energy, it will not participate in future network communication. Using equation (5), lifetime of complete network can be expressed as

$$N_{\text{lifetime}} = \min_{\text{link} \in n} (N_L). \quad (6)$$

Equation (6) can be expressed as

$$N_{\text{lifetime}} = \min_{\text{link} \in n} \left(\frac{E_{\min}(\text{link})}{(E_T + E_R)} \right). \quad (7)$$

We assume that the nodes in WSN are deployed in Gaussian distribution fashion. Gaussian distribution for a random variable x can be defined as

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1(x-\mu)/2\sigma}. \quad (8)$$

Mean or expected value of the data packet is denoted by μ , and standard deviation is σ . It is experienced that if sensor nodes are distributed in Gaussian fashion, the probability of congestion detection is higher as compared to any other distribution strategy. D_{pkt} is the data packets, travelled between v_i and v_n . When arrival rate of source node exceed the service rate of sink node, the data packets are required to wait in the buffer of the sink node. In WSN, each sensor node has a limited buffer space in which waiting data packets can be hold till their service time. When the data

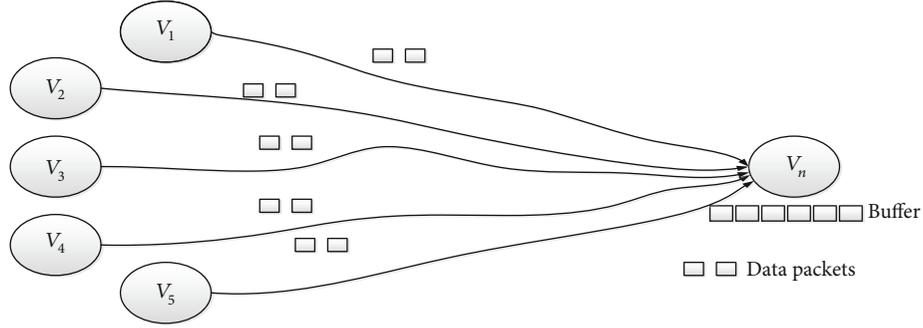


FIGURE 1: Wireless sensor network communication.

packets exceed the limit of buffer, congestion occurs and the data packets may get dropped. This indicates the congestion on the path. Data arrival rate is the average time elapsed from arrival to successfully process from queue. Let λ_A be the arrival rate and λ_D be the service rate or departure rate of D_{pkt} . The following conditions should be met to declare the network congestion-free or congested.

$$\lambda_A > \lambda_D \text{ OR } \lambda_A = \frac{1}{\lambda_D}, \text{ congestion occurs,} \quad (9)$$

$$\lambda_A < \lambda_D, \text{ congestion-free transmission.} \quad (10)$$

Interarrival time of data is $1/\lambda_A$, and data departure rate is $1/\lambda_D$. Also, congestion can be predicted by calculating message generated per unit time (M_p) from a particular node plus message arrival rate. If this is greater than message departure rate of that particular node then it is also a clear indication of congestion in near future. It can be simplified as

$$\frac{1}{\lambda_A + M_p} > \lambda_D. \quad (11)$$

This needs retransmission of dropped data packets. Frequent retransmission of D_{pkt} may lead to queue delay, more energy consumption, and increase congestion. Congestion deteriorates network lifetime and quality of service. The queue occupancy level can be the indicator of congestion. In worst case, if congestion on path occurs, it must be controlled. So to control congestion, the flow of data is redirected to an alternate congestion-free path. In such case, metaheuristic approach is expected to be adopted to redirect the flow on an alternate congestion-free path. Metaheuristic technique promises fast, effective, and efficient relief from congestion. The proposed approach ECA-HA is aimed at controlling congestion by identifying optimal alternate route for data flow and by considering different parameters.

3.2. Congestion Indicator Model. This section elaborates the behavior of buffer occupancy that eventually affects congestion level followed with notification regarding congestion on path. Virtual buffer occupancy (BO_v) by incoming packets and BO_{Max} is the actual maximum buffer size of a node. If

BO_v level exceed to BO_{Max} or buffer occupancy of incoming packet is 95% or more, then it is an indication of congestion. 5% buffer space is left for the packets on fly to prevent them from drop. Here, thr_{max} is the upper threshold of the buffer. Once the congestion indication is favorable, it is mandatory to evaluate congestion level

(a) Buffer occupancy status

BO_{Max} and BO_v are buffer occupancy maximum and virtual buffer occupancy.

If

$$\sum_v^n (\text{BO}_v > \text{BO}_{\text{Max}} \cdot \text{thr}_{\text{max}}) \text{ OR } \text{BO}_v > 0.95 * \text{BO}_{\text{Max}}, \quad (12)$$

where $v = 1, 2, 3, \dots, n$, congestion occurs.

(b) Level of congestion (L_C)

If

$$\frac{\text{BO}_v}{\text{BO}_{\text{Max}}} > = 1, L_C = \frac{\text{BO}_v}{\text{BO}_{\text{Max}}} - 1. \quad (13)$$

Else

$$L_C = 0. \quad (14)$$

Congestion on intermediate node can be computed as

$$L_{C_n} = \sum_{i=1}^n (V_i * L_c). \quad (15)$$

L_{C_n} is the level of congestion at node n . $V_i = V_1, V_2, V_3, \dots, V_n$ are nodes of network. The bandwidth of network also decreases with increasing level of congestion. As the incoming packets increases to its thr_{max} level, congestion level also increases.

- (c) *Notification for Congestion.* When a congestion indicator provides evidence for congestion occurrence, it must implicitly or explicitly notify backward nodes. So to reduce extra overhead and an implicit notification, beacon packet is sent to the source node.
- (d) *Congestion Control.* To control drop rate, the data flow must be redirected to some alternate route. This alternate route must be congestion-free. So to choose alternate congestion-free route, ant colony optimization technique is used. ACO helps in identifying multiple alternate congestion-free paths. Huffman coding helps in selecting an optimal path from multiple alternate paths. The proposed approach is preemptive which monitors the occurrence of an event before its occurrence and avoids its initiation.

3.3. Alternate Congestion-Free Path Selection. This section explains the steps involved to obtain multiple congestion-free alternate paths. Each sensor node maintains a routing table that contains the following information as source ID, sink ID, F-ANT ID (forward ant), distance between source and sink in terms of number of intermediate hops, residual energy of node, and packet loss rate and congestion level of node and pheromone value. Ant colony optimization method constructs multiple alternate paths between sources to sink. F-ANT (forward ant) and B-ANT (backward ant) are two types of ants used in constructing paths. F-ANT moves from source to sink and gathers information in the forward direction. It will initialize a pheromone value in this direction. Then, the pheromone value is updated in the reverse direction from the sink to the source. B-ANT ensures source node regarding energy consumption, packet loss rate, congestion level, and hop counts. In this way, multiple alternate congestion-free paths are constructed by F-ANT and confirm about a successful path creation by B-ANT to the source node.

3.3.1. Forward Ant (F-ANT). The number of F-ANT is generated to construct congestion-free multiple alternate paths between the sources to the sink shown in Algorithm 1. When ant moves from one location to another, it drops pheromone on that path. The density of pheromone is higher on a most favorable path. F-ANT carries information like source ID, sink ID, F-ANT ID, hop count (h_i), energy consumption $E(\text{link})$, congestion level L_C , minimum energy E_{\min} , and time to live TTL. Every node in network maintains pheromone table and routing table.

In F-ANT, $P_{ij}^{(t)}$ is the probability of selecting next hop while moving from the source to the sink.

$$P_{ij}^{(t)} = \frac{(A_{ij})^x * (B_{ij})^y}{\sum_{i,j=1}^n (A_{ij})^x * (B_{ij})^y}. \quad (16)$$

Therefore, $x, y > 0$, and all j belongs to nodes of the network.

A_{ij} is the pheromone density of link (i, j) , and B_{ij} is the heuristic value of link or the attractive coefficient at

```

1. If (TTL! = 0)
2. If (F-ANT is at sink node)
3. Initialize B-ANT
4. Else
5. If Random( $y$ ) < 0.001 then
6. Initialize F-ANT
7. Go to next hop randomly
8. Else
9. Nodes as current and next node are updated here
10. If node already visited
11. Go to other next neighbor node
12. Update heuristic value by step 11
13. END

```

ALGORITHM 1: Pseudo-code for F-ANT.

time “ t .” M^k is the memory to store information about path (visited nodes) which is used to compute path length in terms of hop count. Here, x and y are controlling parameters for pheromone and heuristic values. The pheromone value is updated with updating visiting nodes and expressed in equation (15).

$$A_{ij}(t, t+1) = (1 - E_v)A_{ij}(t) + E_v R_{ij}(t, t+1), \quad (17)$$

$$R_{ij}(t, t+1) = \frac{a * L_i * L_j}{(d_{ij})^2}, \quad (18)$$

$$W_{ij} = \frac{(L_i)^p}{(d_{ij})^q}. \quad (19)$$

E_v is the evaporation factor, and it is decremented with time. L_i and L_j are loads on nodes i and j , and “ a ” is the coefficient. R_{ij} stores load information from source to destination. Here, W_{ij} denotes ratio of load to distance where d_{ij} is the distance between i to j , and p and q are weights or controlling parameters. B_{ij} is the local heuristic value for different paths which incorporates in measuring congestion degree and works as an important parameter to calculate packet loss probability on different paths.

3.3.2. Estimation of Packet Loss Probability. This section explains the applicability of the packet loss probability in the congestion avoidance that is calculated from packet entropy.

As the data rate on different routes is uncertain which can be the cause of congestion in WSN, the packet loss rate is also uncertain here. The uncertainty of information is measured through entropy which was introduced under information theory concept. The information theory was first introduced for statistical thermodynamics. In communication networks, entropy is widely used to measure the abnormality degree of an event which ultimately helps to improve performance metrics of the network such as throughput and energy efficiency. Here, Shannon entropy [40] is used which is a quantitative measure of uncertainty in a data set. Shannon’s work is extended by Jaynes named maximum entropy principle

which has the inherent property to optimize entropy measure when incomplete information is provided in moment constraint form.

Every node in a WSN is designed with a finite buffer space to store the incoming data packets. To evaluate the packet loss rate, the buffer entropy is calculated. The maximum entropy principle is used when there is absence of information regarding mean arrival and service rates. This yields Lagrange's loss formula for providing the express for the state probability distribution of loss system. This maximum entropy principle framework is used to study queue behavior for packet loss rate which considers arrival rate and service rate under normalization constraints, moment constraint, and utilization constraint.

While data travelled from one node to another, the value of entropy changes and rate of data arrival in buffer is also different which changes buffer entropy too. The heuristic data observes data packet arrival rate in buffer. Assume a single server queue with finite buffer of N size. Probability distribution of arrival rate of packets is P_{λ_i} or it is a distribution of the queue with size λ_i . Entropy is defined here as the average of information that is received. Therefore, information quantity (IQ) of the data packets is shown in equation (20).

$$\text{IQ} = -\log_2 \frac{1}{P_{\lambda}}. \quad (20)$$

Shannon entropy [40] is used here, that is, a quantitative measure of uncertainty. The uncertainty arises in probabilistic as well as in deterministic phenomenon where the outcome is about possibility of some specific outcome. Maximum entropy finds the maximum packet loss probability which helps in selecting an optimum path. The path having minimum packet loss probability will be chosen as an optimal path. Packet entropy is average of information quantity. $H(P)$ is a function defined by Shannon to measure uncertainty where n is finite number of data packets transmitted between sources to sink [41].

$$H(P) = -\sum_{i=1}^N P_{\lambda_i} (\log_2 P_{\lambda_i}). \quad (21)$$

The cases are as follows:

- (1) $H(P) = 0$, if the probability of arrival packets is either 0 or 1. This is also known as minimum entropy
- (2) $H(P) = 1$, if the probability of arrival packets is $\frac{1}{2}$. This is also known as maximum entropy

Moment constraints to deduce maximum uncertainty for finite buffer queuing system are defined in equation (22).

$$\sum_{i=1}^N (P_{\lambda_i}) \lambda_i^k = A_k, \text{ where } k = 1, 2, 3, 4 \dots \quad (22)$$

Utilization U defines queue utilization function for non-empty queue is defined in equation (23)

$$\sum_{i=1}^N (h(\lambda_i) (P_{\lambda_i})) = U = 1 - (P_{\lambda_0}). \quad (23)$$

The queuing system also includes the empty system as problem of state space, where (P_{λ_0}) is a probability of zero job in system and $h(\cdot)$ is a function defined as 0 when n is equal to 0 and 1. When n is not equal to 0 where natural probability constraint or normalization constraint is shown in equation (24),

$$\sum_{i=0}^n (P_{\lambda_i}) = 1. \quad (24)$$

Maximizing Shannon entropy is subject to moment constraint in equations (22), (23), and (24). Lagrange multiplier is used to determine maxima or minima of a function. It is a weighted sum of objective and constraint function. α , β , and γ are Lagrange multiplier associated with moment constraint, utilization, and standard probability constraint.

$$\begin{aligned} L(P_{\lambda_i}, \alpha, \beta, \gamma) &= H(P) - \alpha \left(1 - \sum_{i=0}^n (P_{\lambda_i}) \right) \\ &+ \beta \left(U - \sum_{i=1}^n (h(\lambda_i) (P_{\lambda_i})) + \gamma (A_k - \sum_{i=1}^n (P_{\lambda_i}) \lambda_i^k) \right), \end{aligned} \quad (25)$$

$$\begin{aligned} L(P_{\lambda_i}, \alpha, \beta, \gamma) &= -\sum_{i=1}^N P_{\lambda_i} (\log_2 P_{\lambda_i}) - \alpha \left(1 - \sum_{i=0}^n (P_{\lambda_i}) \right) \\ &+ \beta \left(U - \sum_{i=1}^n (h(\lambda_i) (P_{\lambda_i})) + \gamma (A_k - \sum_{i=1}^n (P_{\lambda_i}) \lambda_i^k) \right). \end{aligned} \quad (26)$$

Differentiate the Lagrange function with respect to P_{λ_i} , and maximize Lagrange function, that is, $\partial L / \partial P_{\lambda_i} = 0$, as given in [43, 44], and the value of P_{λ_i} is

$$P_{\lambda_i} = e^{\alpha - \beta \cdot h(\lambda_i) - \gamma \cdot \lambda_i^k}. \quad (27)$$

Now, by substituting the value of P_{λ_i} in differential constraint functions (22) and (23),

$$\lambda_i^k \sum_{k=1}^m e^{\alpha - \beta \cdot h(\lambda_i) - \gamma \cdot \lambda_i^k} = A_k, \quad (28)$$

$$h(\lambda_i) \sum_{k=1}^m e^{\alpha - \beta \cdot h(\lambda_i) - \gamma \cdot \lambda_i^k} = U, \quad (29)$$

1. While (B-ANT not at source node)
2. Move to reverse direction to select next link between i and j
3. Pheromone value is updated
4. Update $P_L(N)$, $E(\text{link})$, $h(\text{link})$, L_{C_i}
5. End while

ALGORITHM 2: Pseudo-code for F-ANT.

$$\sum_{k=1}^m e^{\alpha - \beta \cdot h(\lambda_i) - \gamma \cdot \lambda_i^k} = 1. \quad (30)$$

Now, for $k = 1 \dots m$, solve equation (30) to obtain value of e^α as shown in equation (31).

$$e^\alpha = e^{\beta \cdot h(\lambda_i) + \gamma \cdot \lambda_i^k}. \quad (31)$$

The packet loss probability $P_L(N)$ [43] for fully occupied finite buffer using maximum entropy probability distribution of system size is computed as

$$P_L(N) = P_{\lambda_i} = \frac{(1 - P_{\lambda_0}) \lambda_i^{-\gamma}}{\zeta(\gamma)}. \quad (32)$$

Here, $\zeta(\gamma)$ is Riemann zeta function widely used in number theory when distribution is related to prime numbers. As $i \geq 1$ is an infinite series shown as in equation (33),

$$\zeta(\gamma) = \sum_{i=1}^N \lambda_i^{-\gamma}, \gamma > 1. \quad (33)$$

3.3.3. Backward Ant (B-ANT). The B-ANT follows backward path from sink to source node identified by F-ANT. The pheromone value is updated with every move in communication link shown in Algorithm 2. It also contains source ID, sink ID, B-ANT ID, energy consumption, hop counts, and congestion level. $A_{ij}(t, t+1)$ is updated pheromone value of link. The computation function of path for packet loss rate, total link energy consumption, hop counts of current path, and congestion level of link are $f_1(t)$, $f_2(t)$, $f_3(t)$, and $f_4(t)$, respectively, along with e , l , m , and n weight parameters. Here, $a1$ and $a2$ are used as controlling parameters to monitor pheromone updating. The ant moves to the next node in backward direction and updates the pheromone value. It also recalculates and updates the mentioned parameters, in order to make it capable for optimal decision.

$$A_{ij}(t, t+1) = a1 * F(t+1), \quad (34)$$

$$F(t+1) = a2 \frac{F_1(t)^e}{F_2(t)^l * F_3(t)^m * F_4(t)^n}, \quad (35)$$

$$f_1(t) = \min(P_L(N)), \quad (36)$$

$$f_2(t) = \sum E(\text{link}), \quad (37)$$

$$f_3 = \sum_{i,j \in h(\text{link})} (i, j), \quad (38)$$

$$f_4 = \sum_{i=1}^n L_{C_i}. \quad (39)$$

3.4. Select Optimal Path Using Huffman Coding. After ACO identified multiple alternate congestion-free path, Huffman coding selects optimal path. For this, a tree of N nodes is generated with nodes having low congestion level near to root node and high congestion level near to leaf node. The root node act like sink node. The objective is to choose a path where congestion is less or no congestion near sink node. Such path is considered as an optimal one. Well, all the parameters play important role in selecting optimal path. But congestion is our major concern, so Huffman coding opts an optimal path having minimum congestion level.

Multiple paths can be represented by Q_i , and $h(\text{link})$ is the path length with W_i a weight. So,

$$Q_i = \sum_{i=1}^N W_i h_i(\text{link}). \quad (40)$$

Congestion level-oriented optimal path evaluation function g_i can be represented as

$$g_i = \left\{ \frac{L_{C_{\text{Avg}}}}{L_{C_i}} * P_L(N) \right\}^\tau + (L_{C_{\text{min}}})^\omega, \quad (41)$$

$$\text{Opt}(\text{path}) = \min_{i \in N} (g_i). \quad (42)$$

Here, $\text{opt}(\text{path})$ is optimal path chooses after Huffman coding. $P_L(N)$ is the packet loss rate and $L_{C_{\text{min}}}$ is the minimum congestion on link. τ and ω are the controlling parameters for hop count and congestion level.

3.5. Huffman Coding for Congestion Control. This section defines the working of Huffman coding for selection of optimal path among multiple alternate paths identified by ACO. It focuses on packet loss rate on different paths while selecting optimal path. The working of Huffman coding (HC) is elaborated in Figure 2, where a, b, c, d, e, and f are the different nodes of a network with different packet loss rate 6, 10, 13, 14, 17, and 46 as shown in Figure 2(a). Huffman tree is also constructed to make better understanding of how to opt an optimal path in WSN during congestion. This Huffman coding approach is aimed at arranging the nodes in an optimal way of their usage. Create a min heap of six nodes, and each node represents root of a tree with single node.

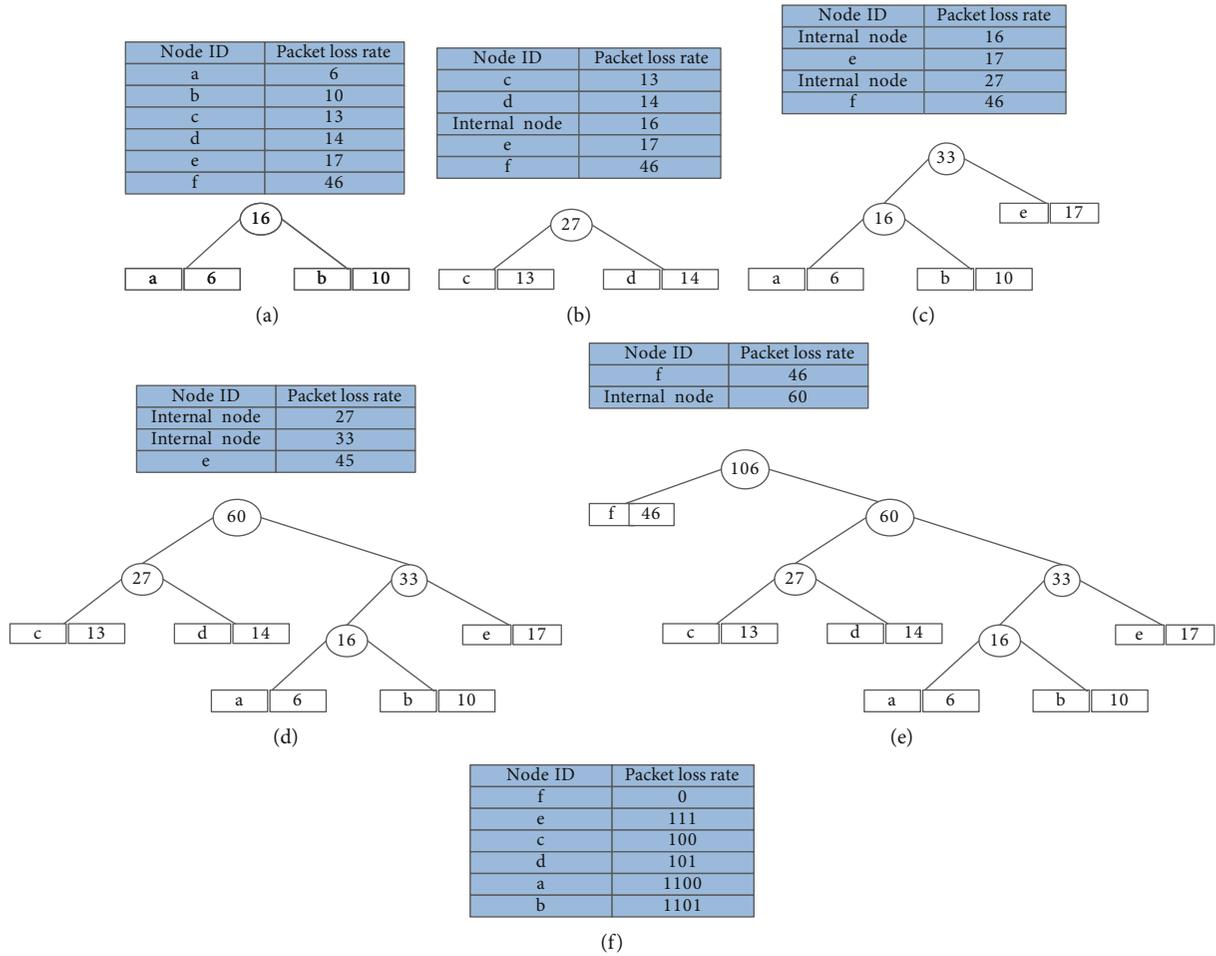


FIGURE 2: Huffman coding for optimal path selection.

Extract two nodes with minimum packet loss rate from min heap and add an internal node with packet loss rate $6 + 10 = 16$, shown in Figure 2(a). Now min heap has five nodes, where four nodes are roots of tree with three elements shown in Figure 2(b). Extract two nodes with minimum packet loss rate from heap. Add a new internal node with packet loss rate $13 + 14 = 27$. Now, min heap has four nodes, where two nodes are roots of tree with single element each and two heap nodes are roots of tree with more than one node shown in Figure 2(c). Extract two minimum nodes with packet loss rate again. Add internal node with packet loss rate $16 + 17 = 33$. Now, min heap has three nodes. Extract two nodes with minimum packet loss rate. Add a new internal node with packet loss rate $27 + 33 = 60$ shown in Figure 2(d). Now, min heap has two nodes. Extract two nodes with minimum packet loss rate. Add a new internal node with packet loss rate $46 + 60 = 106$ shown in Figure 2(e). Now, min heap has only one node. Since the heap has only one node, the algorithm ends here. Start traversing tree from root node. For all left child, write 0, and for all right child, write 1.

3.6. *ECA-HA Flow Chart.* The complete operations of proposed technique are presented by a flow chart in Figure 3. The three major portions of this flow chart clearly define

the operations of proposed technique. During communication, if λ_A is greater than λ_D then check the congestion level by computing buffer occupancy level, and if virtual buffer occupancy is greater than maximum buffer limit then it is a link layer congestion and it is a time to reroute the data on an alternate route; otherwise, the network is safe and communication goes smooth. Then, ACO operation performs here, and ACO identifies multiple alternate congestion-free paths. It identifies packet loss probability by calculating packet entropy which helps in identifying an optimal path. Huffman coding selects an optimal path among multiple congestion-free alternate paths. The pseudo-code for ACO clearly defined all the steps involved.

3.7. *Time Complexity.* The time complexity of the proposed ECA-HA algorithm depends upon the complexity of Huffman coding and ant colony optimization. For the “ n ” number of sensor nodes, the Huffman coding takes $O(n \log n)$ to control the congestion in the path of the network [10], whereas ant colony optimization selects the best route under duration $O(1/E_v n^2 \log n)$, with E_v defines the evaporation rate of pheromone [35]. Thus, overall time complexity of the proposed ECA-HA is the summation of $O(n \log n) + O(1/\rho n^2 \log n) = O(1/\rho n^2 \log n)$.

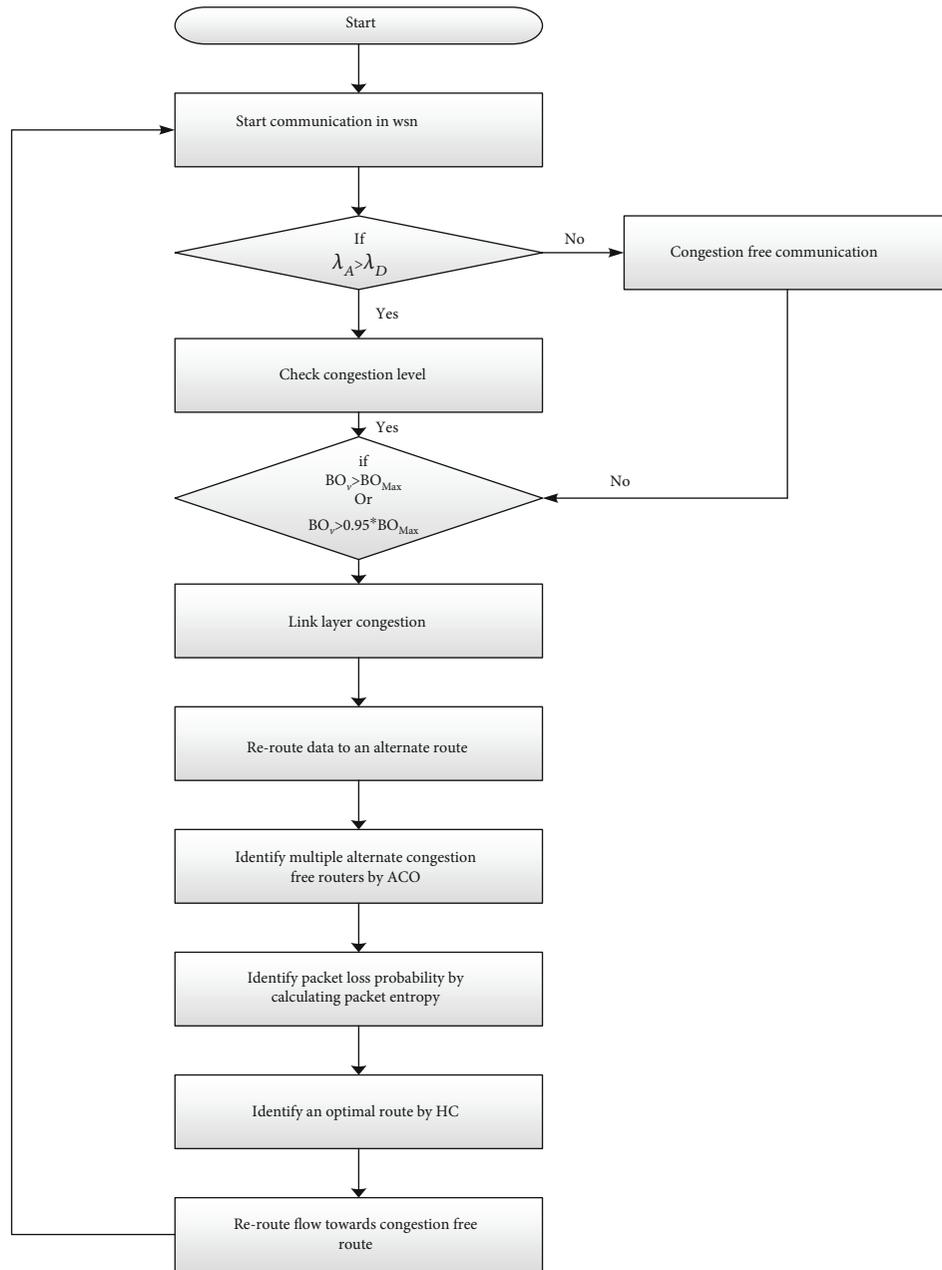


FIGURE 3: Flow chart of ECA-HA

4. Simulation and Result Discussion

In this section, we performed the extensive simulation using simulator MATLAB2017b of the proposed algorithm. The sensor nodes are in range from 20 to 100 randomly distributed in the 2D ($100 \times 100 m^2$) space of the wireless network. Base station is placed inside (40, 72) in the network which has capability of data gathering and query processing. The initial energy of node is $E_i = 0.5$ joules. The pheromone released by ant is evaporation once in every 4 seconds, whereas the amount of search ant is set to be 6 and their maximum lifetime is 20. The maximum number of iterations or round is 1000. The remaining list of parameters used for simulations is shown in Table 2. The proposed algorithm is compared

with the nature-inspired algorithm ACSRO [19] and traditional congestion avoidance algorithm CODA [13]. The performance of the proposed algorithm is evaluated against the average throughput, average hop-by-hop delay, and packet delivery ratio and node death percentage. These parameters define the rate of successful delivery of data over the total bandwidth, time delay in forwarding the packet from current hop to next forward hop, and ratio of the number of packets successfully received by receiver to the number of packets send from source. Further, comparative analysis of the ECA-HA with state-of-art algorithm is also evaluated.

4.1. Average Energy Consumption over Rounds. A comparison of energy consumption of the proposed ECA-HA algorithm

TABLE 2: Simulation parameters.

Parameter	Value
Network area	$100 \times 100\text{m}^2$
Number of nodes (N)	20-100
E_{amp}	$0.0013 \text{ pJ/bit/m}^4$
E_{elec}	50 nJ/bit
E_i	0.5 J
D_{pkt}	500 bits
d_{ij}	[5, 20]
Control packet size	100 bits
Data transmission rate (λ_D)	512 Kb/s
Data arrival rate (λ_A)	700 Kb/s
Communication range	20 m
Max packet in buffer (BO_{Max})	40
Congestion threshold of buffer (thr_{max})	33
Pheromone density of link (A_{ij})	20
Evaporation factor (E_v)	0.65
Antenna type	Omni antenna

with state-of-art algorithms over number of rounds is shown in Figure 4. It can be observed from the results, for the initial phase of rounds at 400, the energy consumption is 0.12 joules, 0.2 joules, and 0.51 joules for ECA-HA, ACSRO [19] (nature-inspired algorithm), and CODA, respectively. And further increase in the rounds up to 1000 rounds the ECA-HA only consumes 1.4 joules, and ACSRO consumes 1.6 joules and worst performance shown by CODA that consumes around 1.9 joules. This is due to the fact that the proposed algorithm chose the next hop routing path based upon higher residual energy and less congested routing density based on Huffman coding and ant colony optimization, whereas ACSRO uses adaptive cuckoo search rate adjustment for the selection of next hop that performs better than CODA algorithm. It is also noted down that the worst performance is shown by without nature-inspired algorithm, because it does not have any selection policy for the next hop routing.

4.2. Average Residual Energy of Nodes over Rounds. A comparison of average residual energy between ECA-HA and the state-of-art algorithm is presented in Figure 5. It is clearly observed from the result as the number of rounds increases, residual energy decreases for all the three algorithms, whereas ECA-HA and ACSRO having residual energy 0.4 and 0.2 joules, respectively, for 1000 rounds. Further, it is noticeable that CODA algorithm exhausts all the available energy and in turn zero residual energy. This is because of proposed algorithm ECA-HA selects the next hop for the routing purpose based on ant colony optimization, and priority is given to nodes having higher residual energy using Huffman coding, whereas ACSRO selects the path for packet delivery based upon residual energy and minimum hop distance but does not include the congestion density on the next available node rather the proposed ECA-HA considered all the mentioned parameters. It is also worthy to note down

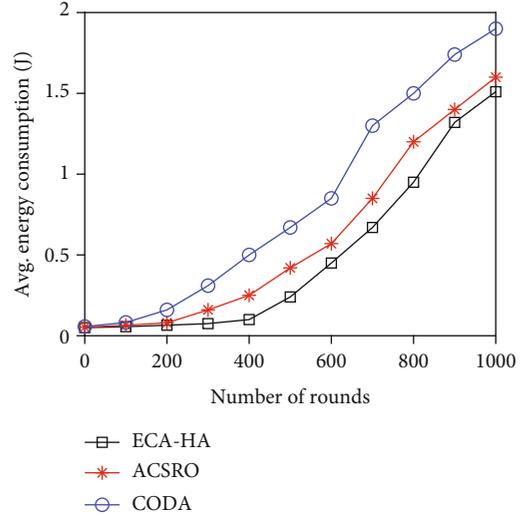


FIGURE 4: Average energy consumption.

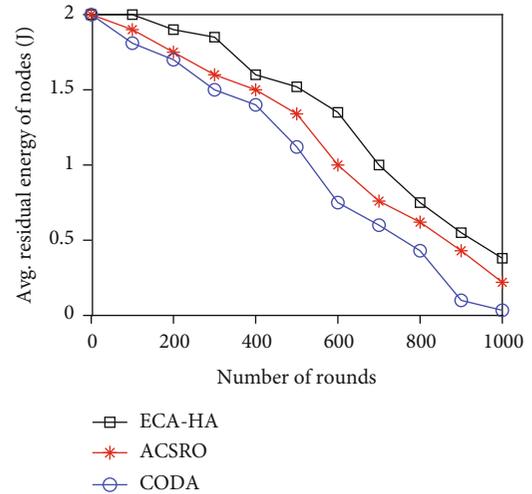


FIGURE 5: Average residual energy.

that CODA algorithm shows poorest performance due to selection of the next route depends on minimum distance and does not consider residual energy of the next hop that in turns increases the overall energy consumption, and life-time of the network is decreased.

4.3. Average Throughput over Source Data Rate. A comparison of average throughput of the proposed algorithm using 20 and 100 numbers of nodes in the network is shown in Figures 6(a) and 6(b), respectively. It is clearly observed from the result of Figure 6(a) that proposed algorithm ECA-HA improves throughput about 18% comparing to ACSRO and 21% comparing to CODA algorithm, respectively, whereas average throughput in Figure 6(b) shows that all three state-of-art algorithms in the beginning (source data rate = 10) provide nearly the same 50 packets per second delivery rate. With further increase in the source data rate, our proposed algorithm ECA-HA increases the average throughput rapidly and reaches about 100 packets per second comparing

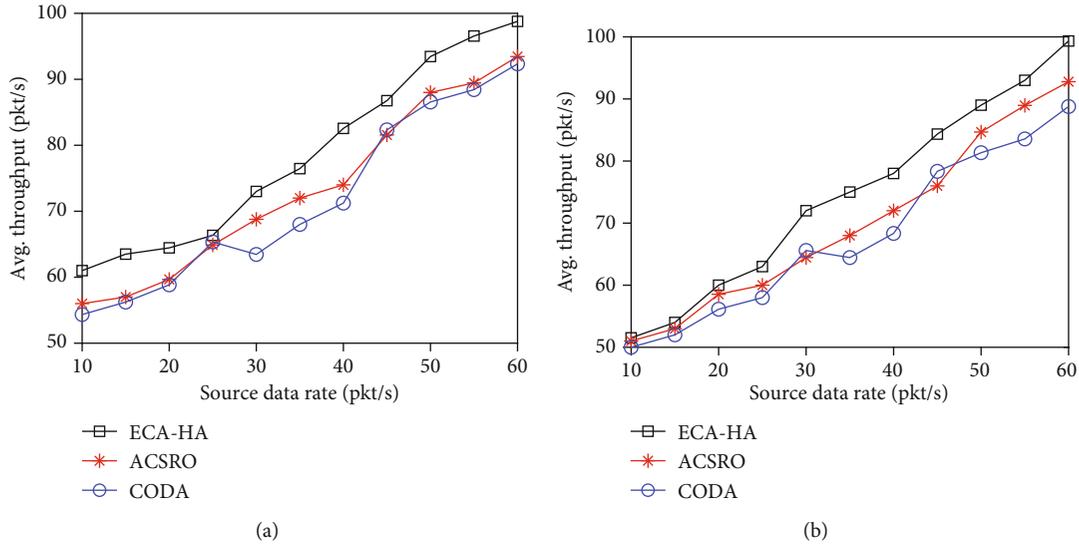


FIGURE 6: (a) Avg. throughput per 20 nodes. (b) Avg. throughput per 100 nodes.

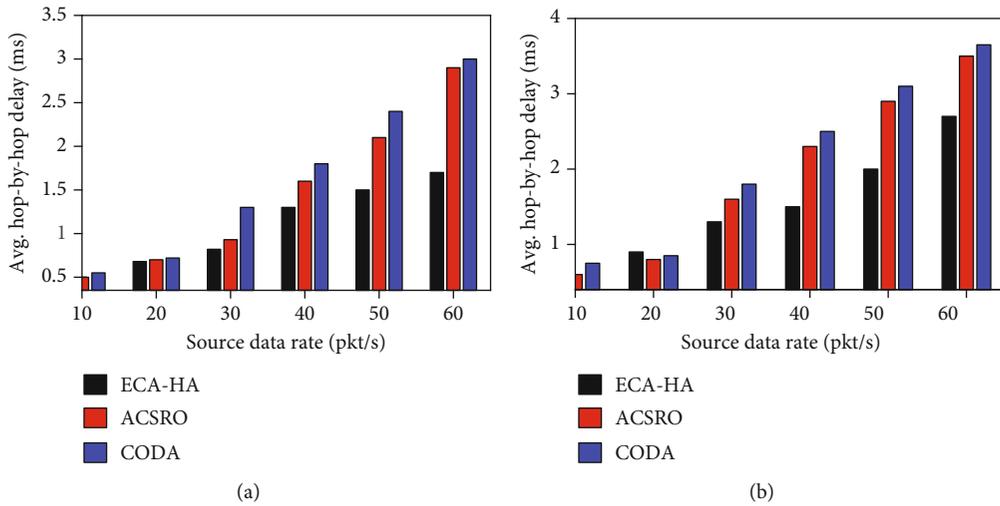


FIGURE 7: (a) Avg. hop delay per 20 nodes. (b) Avg. hop delay per 100 nodes.

to 91 packets per second and 89 packets per second for ACSRO and CODA, respectively. This is due to the fact that data transmission level congestion is controlled through the effective selection of next hop based on ant colony approach that provides alternate path having lower number of packets to forward. It can be also observed that average throughput in the network using CODA algorithm shows much variation than other state-of-art algorithms. This can be attributed to the reason CODA algorithm consumes more energy to forward the packets and rarely selects optimal next hop for the forwarding. Thus, overall it is clearly proved that ECA-HA is the best suitable algorithm for data forwarding in the wireless sensor network composed to 20 or more than 100 nodes environment. This can address the problem of congestion, missing packets in wireless sensor network, and upgradation of network throughput.

4.4. Average Hop-by-Hop Delay over Source Data Rate. A comparison result of average hop-by-hop delay between proposed algorithm ECA-HA and state-of-art algorithm is shown in Figures 7(a) and 7(b) in the scenario nodes 20 and 100, respectively. This metric plays very important to show the interpath interference control mechanism, overhead produce by the exchange of control packets, and link-layer transmission delay. It can be clearly observed from the result of Figure 7(a) that proposed algorithm ECA-HA has less delay about 33% and 38% compared to ACSRO and CODA, respectively, for the source data rate 40 packets per seconds and onwards, whereas in Figure 7(b), the ECA-HA delivers the packets to next hop faster about 39% comparing to ACSRO and 43% comparing to CODA. The reason behind is that proposed algorithm ECA-HA exchanges minimum number of control packets for the selection of routing path.

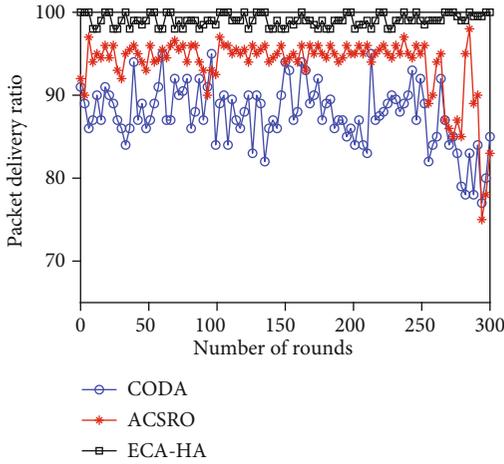


FIGURE 8: Packet delivery ratio.

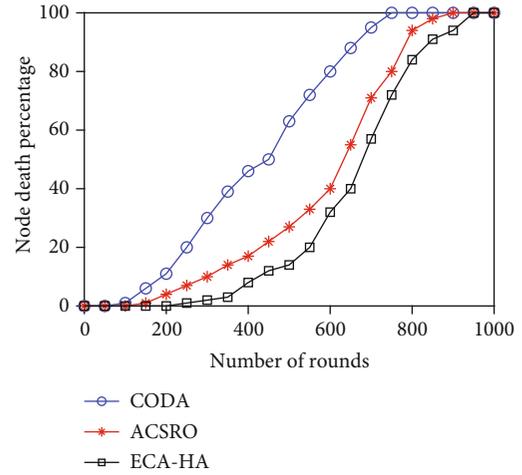


FIGURE 9: Node death percentage.

This can be attributed to the reasons of Huffman coding uses priority-based heap for the selection for next hop based on distance between node and base station. Further, data routing path is balanced using ant colony approach that helps in minimizing the hop-by-hop delay, whereas ACSRO and CODA are not able to optimize the number of control packet exchanges that inherently increases the interference too. This can be attributed to increase the congestion in the network in turn enhance the overall hop-by-hop delay.

4.5. Packet Delivery Ratio over Number of Rounds. A comparison of packet delivery ratio over number of rounds between proposed algorithm ECA-HA and the state-of-art algorithm is presented in Figure 8. Packet delivery ratio turns out to be a throughput measurement parameter; the higher the packet delivery ratio, the higher the throughput in the network. It is clearly observed from the result on the increasing number of rounds the packet delivery ratio is higher about 95%-98% for the proposed algorithm. This is due to the fact that proposed algorithm uses Huffman coding for the selection of routing path to deliver the packet with less congested path, whereas nature-inspired ACSRO algorithm performs better than traditional congestion avoidance algorithm CODA with 35% higher packet delivery ratio. This is because of the ACSRO selects the next hop for the routing purpose based on adaptive cuckoo search rate adjustment for optimized congestion avoidance. Finally, it is observed that proposed algorithm ECA-HA performs better than by 8% and 48% higher packet delivery with respect to ACSRO and CODA, respectively.

4.6. Node Death Percentage over Number of Rounds. A comparison of node death percentage over number of rounds between proposed algorithm ECA-HA and the state-of-art algorithm is presented in Figure 9. It is clearly observed from the result as the number of rounds increases, the death percentage of nodes also increases for all the three algorithms and all the nodes are dead at maximum 950 rounds. Further, it is noticeable that proposed ECA-HA-based congestion control routing algorithm has 20% and 40% lower node

death percentage ratio rather than ACSRO and CODA, respectively, at 600 rounds. It is also worthy to note that proposed algorithm ECA-HA runs about 960 rounds, whereas ACSRO and CODA run only for 820 and 640 rounds. This is due to the fact ECA-HA chose the routing path with lower congestion rate and having more residual energy compare to ACSRO and CODA. And the without nature-inspired-based algorithm CODA selects any node for data forwarding with consideration of hop distance and residual energy, which leads to worst performance, and lifetime of the network is decreased.

5. Conclusion and Future Scope

This paper proposed a congestion control algorithm ECA-HA for wireless sensor networks that can be applied on most of the application like health care system, agriculture, and monitoring. The proposed algorithm uses ant colony approach using two algorithms for backward ant and forward ant algorithms to find more than one path for data routing. Further, Huffman coding was applied on the top to select one of the less congested paths as optimal data routing path from the sample space given by ant colony approach. The extensive performance results of the proposed algorithm outperform in terms of various parametric than state-of-art algorithms. Thus, the proposed algorithm forwards the packets using optimal data routing path in the current round that decreases the hop-by-hop delay, improves the packet delivery ratio, or decreases the node death percentage which ultimately improves the throughput of the network and lifetime, respectively. The only shortcoming of the proposed algorithm is that the network is increasing at a very fast rate but this approach is suitable for small search space. Therefore, in future, we will look forward for large search space. In the future, considering Internet of Things environment, we will include the more parameter such as dynamic traffic load or breakage in the link to find more optimal data routing path based on reinforcement learning approach.

Data Availability

The experimental data and associated settings will be made available to researchers and practitioners on individual request to the corresponding author, with the restrictions that it will solely be used for further research in literature progress, as the associated research data is being further utilized for development research by the team.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the SC&SS, Jawaharlal Nehru University, New Delhi.

References

- [1] S. Yadav, "A study on congestion control mechanisms in wireless sensor networks," *Journal of Advanced Research in Dynamical & Control Systems*, vol. 10, pp. 842–850, 2015.
- [2] F. Ullah, A. H. Abdullah, O. Kaiwartya, S. Kumar, and M. M. Arshad, "Medium access control (MAC) for wireless body area network (WBAN): superframe structure, multiple access technique, taxonomy, and challenges," *Human-centric Computing and Information Sciences*, vol. 7, no. 34, pp. 1–39, 2017.
- [3] S. L. Yadav and R. L. Ujjwal, "Mitigating congestion in wireless sensor networks through clustering and queue assistance: a survey," *Journal of Intelligent Manufacturing*, pp. 1–16, 2020.
- [4] S. L. Yadav and R. L. Ujjwal, "Sensor data fusion and clustering: a congestion detection and avoidance approach in wireless sensor networks," *Journal of Information and Optimization Sciences*, vol. 41, no. 7, pp. 1673–1688, 2020.
- [5] A. Khasawneh, M. S. B. A. Latiff, O. Kaiwartya, and H. Chizari, "Next forwarding node selection in underwater wireless sensor networks (UWSNs): techniques and challenges," *Information*, vol. 8, no. 1, p. 3, 2017.
- [6] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud et al., "Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4049–4062, 2018.
- [7] O. Kaiwartya, A. H. Abdullah, Y. Cao et al., "T-MQM: testbed-based multi-metric quality measurement of sensor deployment for precision agriculture—a case study," *IEEE Sensors Journal*, vol. 16, no. 23, pp. 8649–8664, 2016.
- [8] P. Kumar Kashyap, S. Kumar, U. Dohare, V. Kumar, and R. Kharel, "Green computing in sensors-enabled internet of things: neuro fuzzy logic-based load balancing," *Electronics*, vol. 8, no. 4, p. 384, 2019.
- [9] A. Khatri, S. Kumar, O. Kaiwartya, N. Aslam, N. Meena, and A. H. Abdullah, "Towards green computing in wireless sensor networks: controlled mobility-aided balanced tree approach," *International Journal of Communication Systems*, vol. 31, no. 7, article e3463, 2018.
- [10] S. Kumar, O. Kaiwartya, and A. H. Abdullah, "Green computing for wireless sensor networks: optimization and Huffman coding approach," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 592–609, 2017.
- [11] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and A. H. Gandomi, "Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network," *IEEE Sensors Journal*, 2021.
- [12] L. Farhan, O. Kaiwartya, L. Alzubaidi, W. Gheth, E. Dimla, and R. Kharel, "Toward interference aware IoT framework: energy and geo-location-based-modeling," *IEEE Access*, vol. 7, pp. 56617–56630, 2019.
- [13] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 266–279, New York, NY, USA, Nov 2003.
- [14] S. Brahma, M. Chatterjee, and K. Kwiat, "Congestion control and fairness in wireless sensor networks," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 413–418, Mannheim, Germany, 2010, March.
- [15] K. Kashyap and S. Kumar, "Load-balanced distributed intra-clustering algorithm," in *2015 Annual IEEE India Conference (INDICON)*, pp. 1–6, New Delhi, India, 2015.
- [16] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu, "Priority-based congestion control in wireless sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, pp. 1–8, Taichung, Taiwan, June 2006.
- [17] M. Z. Ahmad and D. Turgut, "Congestion avoidance and fairness in wireless sensor networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pp. 1–6, New Orleans, LA, USA, 2008, November.
- [18] C. Sergiou and V. Vassiliou, "DALPaS: a performance aware congestion control algorithm in wireless sensor networks," in *18th International Conference on Telecommunications*, pp. 167–173, Ayia Napa, Cyprus, May 2011.
- [19] V. Narawade and U. D. Kolekar, "ACSRO: adaptive cuckoo search based rate adjustment for optimized congestion avoidance and control in wireless sensor networks," *Alexandria Engineering Journal*, vol. 57, no. 1, pp. 131–145, 2018.
- [20] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 205–210, San Antonio, TX, USA, 2009.
- [21] N. Siddique and H. Adeli, "Nature inspired computing: an overview and some future directions," *Cognitive Computation*, vol. 7, no. 6, pp. 706–714, 2015.
- [22] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "Congestion control in wireless sensor networks based on bird flocking behavior," *Computer Networks*, vol. 57, no. 5, pp. 1167–1191, 2013.
- [23] M. A. Kafi, J. Ben-Othman, A. Ouadjaout, M. Bagaa, and N. Badache, "REFIACC: reliable, efficient, fair and interference-aware congestion control protocol for wireless sensor networks," *Computer Communications*, vol. 101, no. C, pp. 1–11, 2017.
- [24] P. K. Kashyap, "Genetic-fuzzy based load balanced protocol for WSNs," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 2, pp. 2088–8708, 2019.
- [25] P. K. Kashyap, S. Kumar, and A. Jaiswal, "Deep learning based offloading scheme for IoT networks towards green computing," in *IEEE International Conference on Industrial Internet (ICII)*, pp. 22–27, Orlando, FL, USA, Nov 2019.

- [26] K. Singh, K. Singh, L. H. Son, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Computer Networks*, vol. 138, pp. 90–107, 2018.
- [27] A. Jaiswal, S. Kumar, O. Kaiwartya et al., "Quantum learning enabled green communication for next generation wireless systems," *IEEE Transactions on Green Communications and Networking*, 2021.
- [28] J. S. Lee and C. L. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1095–1103, 2017.
- [29] H. El Alami and A. Najid, "MS-routing-G i: routing technique to minimise energy consumption and packet loss in WSNs with mobile sink," *IET Networks*, vol. 7, no. 6, pp. 422–428, 2018.
- [30] H. El Alami and A. Najid, "ECH: an enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks," *IEEE Access*, vol. 7, pp. 107142–107153, 2019.
- [31] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "A heuristic path search for congestion control in WSN," in *Industry Interactive Innovations in Science, Engineering and Technology*, Springer, Singapore, 2018.
- [32] R. Logambigai, S. Ganapathy, and A. Kannan, "2018. Energy-efficient grid-based routing algorithm using intelligent fuzzy rules for wireless sensor networks," *Computers & Electrical Engineering*, vol. 68, pp. 62–75, 2018.
- [33] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "An improved congestion-aware routing mechanism in sensor networks using fuzzy rule sets," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 890–904, 2020.
- [34] M. Selvi, S. V. N. Santhosh Kumar, S. Ganapathy, A. Ayyanar, H. Khanna Nehemiah, and A. Kannan, "An energy efficient clustered gravitational and fuzzy based routing algorithm in WSNs," *Wireless Personal Communications*, vol. 116, no. 1, pp. 61–90, 2021.
- [35] M. Mandloi and V. Bhatia, "Congestion control based ant colony optimization algorithm for large MIMO detection," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3662–3669, 2015.
- [36] A. A. Gurav and M. J. Nene, "Multiple optimal path identification using ant colony optimisation in wireless sensor network," *International Journal of Wireless & Mobile Networks*, vol. 5, no. 5, pp. 119–128, 2013.
- [37] W. Ding, L. Tang, and S. Ji, "Optimizing routing based on congestion control for wireless sensor networks," *Wireless Networks*, vol. 22, no. 3, pp. 915–925, 2016.
- [38] C. J. Raman and V. James, "FCC: fast congestion control scheme for wireless sensor networks using hybrid optimal routing algorithm," *Cluster Computing*, vol. 22, no. S5, pp. 12701–12711, 2019.
- [39] A. A. Rezaee and F. Pasandideh, "A fuzzy congestion control protocol based on active queue management in wireless sensor networks with medical applications," *Wireless Personal Communications*, vol. 98, no. 1, pp. 815–842, 2018.
- [40] J. Zhang, Z. Lin, P. W. Tsai, and L. Xu, "Entropy-driven data aggregation method for energy-efficient wireless sensor networks," *Information Fusion*, vol. 56, pp. 103–113, 2020.
- [41] W. Osamy, A. Salim, and A. M. Khedr, "An information entropy based-clustering algorithm for heterogeneous wireless sensor networks," *Wireless Networks*, vol. 26, no. 3, pp. 1869–1886, 2020.
- [42] J. N. Kapur and H. K. Kesavan, "Entropy Optimization Principles and Their Applications," in *Entropy and Energy Dissipation in Water Resources*, Springer, Dordrecht, 1992.
- [43] A. K. Singh and Karmeshu, "Power law behavior of queue size: maximum entropy principle with shifted geometric mean constraint," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1335–1338, 2014.
- [44] S. Sharma and S. Kumar, "Generation of power law: maximum entropy framework and super statistics," in *Man-Machine Interactions, Advances in Intelligent Systems and Computing*, pp. 45–59, Springer, Champions, 2016.