This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.Doi Number

An Enhanced Distributed Congestion Control Method for Classical 6LowPAN Protocols Using Fuzzy Decision System

Mohammad Hossein Homaei[1,2], Faezeh Soleimani[3], Shahaboddin Shamshirband[4,5], Amir Mosavi[1,2,6], Narjes Nabipour[7], and Annamaria R. Varkonyi-Koczy[1,2]

[1]Institute of Automation, Kalman Kando Faculty of Electrical Engineering, Obuda University, Budapest, Hungary
[2]Department of Mathematics and Informatics, J. Selye University, 94501 Komarno, Slovakia
[3]Department of Mathematics, University of Texas at Arlington, TX, USA
[4]Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[5]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[6]Institute of Structural Mechanics, Bauhaus University Weimar, 99423 Weimar, Germany
[7]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

Corresponding author: Shahaboddin Shamshirband (shahaboddin.shamshirband@tdtu.edu.vn)

ABSTRACT The classical Internet of things routing and wireless sensor networks can provide more precise monitoring of the covered area due to the higher number of utilized nodes. Because of the limitations in shared transfer media, many nodes in the network are prone to the collision in simultaneous transmissions. Medium access control protocols are usually more practical in networks with low traffic, which are not subjected to external noise from adjacent frequencies. There are preventive, detection and control solutions to congestion management in the network which are all the focus of this study. In the congestion prevention phase, the proposed method chooses the next step of the path using the Fuzzy decision-making system to distribute network traffic via optimal paths. In the congestion detection phase, a dynamic approach to queue management was designed to detect congestion in the least amount of time and prevent the collision. In the congestion control phase, the back-pressure method was used based on the quality of the queue to decrease the probability of linking in the pathway from the pre-congested node. The main goals of this study are to balance energy consumption in network nodes, reducing the rate of lost packets and increasing quality of service in routing. Simulation results proved the proposed Congestion Control Fuzzy Decision Making (CCFDM) method was more capable in improving routing parameters as compared to recent algorithms.

INDEX TERMS Internet of things, wireless sensor network, congestion control, fuzzy decision making, and back-pressure.

I. INTRODUCTION

The classical internet of things network and low power and lossy networks (LLNs) are useful domains of wireless sensor networks, which can provide area monitoring and control operations with high potential in unpredictable and dynamic environments [1]. However, these networks face challenges in transferring information to the base station in the network due to wireless media and changing topology, which hinders routing protocols. Hardware and software limitations, which are an intrinsic property of these types of networks, make them susceptible to physical elements and environmental effects and lead to error and damages. The protocols designed for LLN and wireless sensor networks are very specific to their applications.

Nevertheless, one of the main challenges to these types of networks is increasing the quality of service and node longevity in the network, which is due to the limited resources of wireless sensor nodes [2]. Several factors cause energy loss in network nodes, which include collision frames, retransmitting from source, crosstalk, queue delay, hidden terminal, over emitting, idle-listening and control overhead of designed protocols [3]. Most of the aforementioned factors can be managed and controlled at the Medium Access Control (MAC) sub-layer [4]. However, most protocols designed for the MAC sub-layer have single-channel properties in the network. In such protocols, especially in high concentrations of nodes in the network, there is a probability of collision increase, noise increase due to crosstalk, end-to-end delay, and ultimately reduced network longevity. By creating hardware's

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

such as the transceiver CC2538, which can achieve a 16-channel information transmission and reception at 5-megahertz frequency gaps, and by utilizing this transceiver at sensor nodes like OpenMote, the path has been cleared for creating multi-channel MAC protocols [5]–[7]. Nevertheless, to implement multi-channel protocols in wireless sensor networks has its challenges. To achieve higher performance in multi-channel protocols, the node radios have to change between different frequencies. Consequently, for proper information exchange, the transmitter and receiver have to be constantly synchronized into common frequencies. Therefore, it is essential to manage the distributed nodes to the correct channel [8]. There are other challenges to designing the multi-channel protocols for the MAC sub-layer of LLNs including [9]: multi-channel hidden terminal, absent receiver (no listening), broadcast support, delayed channel switch, optimum allocation of channels to nodes, connecting a new node to the network and avoiding network partitioning [10]. Based on the aforementioned, other researchers manage the problem of package collision in the network-by-network layer solutions [11], [12]. In these methods, the aim is for the network to operate within an objective function that has the least probability of collision; this procedure is known as preventative processes [13]. Obviously, by preventing collision and congestion in the network, the resource loss and overhead due to message retransmission can be decreased. Moreover, it increases the nodes required time to access wireless media. On the other hand, congestion and collision in a high traffic network are inevitable. In this regard, the more management of the network leads to higher overhead control of network nodes. Therefore, the process of congestion management in the network must be a complete process including congestion prevention, detection, confrontation and control. To achieve these goals the network faces a challenge since numerous parameters within the network without the use of the decision system complicates the process. However, the MADM and MODM have been utilized as good solutions [14], [15]. In the present study, a congestion control method based on Congestion Control Fuzzy Decision Making (CCFDM) has been proposed, which prevents congestion in the network and manages the data flow by timely detecting congestion and confronting it in a distributed manner, this in turn helps in maintaining network resources.

The paper is organized as follows: In Section II, previous studies on congestion detection (sub-section A) and control (sub-section B) in classical internet of things and wireless sensor networks are reviewed. In Section III, a method of routing informed in quality of service based on fuzzy decision making has been introduced, which benefits from queue status for congestion detection and in congestion control from a proposed method of back-pressure. In section IV, we assessed and compared the proposed CCFDM method with other recent methods considering average lifetime, packet delivery, delay, queue efficiency, Jain Fairness Index, and power efficiency. Finally, section V is dedicated to conclusions and future recommendations.

II. REVIEW OF LITERATURE

A. *Congestion detection and control in classical internet of things network*

Limited resources of network nodes and various traffic patterns present many challenges to routing and data flow transmission in classical internet of thing networks. These challenges are so serious that they can affect the whole process of data transfer between nodes and sink and even cause disruption [16], [17]. For this purpose, various methods have been proposed to control and manage congestion, which some of the more important ones will be discussed hereafter. Three main models in congestion control have been suggested, namely[18]:

1) *End to End methods:*

In this procedure, a congestion control mechanism has been embedded in the transport layer [19]. The receiver can identify the number of sent and delivered packages by reviewing the Acknowledgement message (ACK) and the sequence number of packages. Nevertheless, due to long and multiple routes and because the information has been periodic data and there may be packets ready for sending at any given time, implementing this method in LLNs is not possible The suitable and non-delayed use of ACK in the network is not possible, and studies are trying to solve this problem.

2) *Route based methods:*

This method tries to remove some of the problems of the End to End method. The main difference of this method is the faster congestion detection in the network [20], [21]. When congestion is detected in the network, the source node is notified by a backward signal that is sent from the point of congestion and received step by step. This solution is only practical when the congestion is near to the source. If the congestion is close to the destination and hence far away from the source of the package, the problems of the end-to-end method persist [22].

3) *Step by step methods:*

This approach uses the step by step mechanism in the detection and prevention of congestion. Compared to previous methods, this approach does not require a backward mechanism in long routes [23], [24]. The congestion problem is locally solved through the connection between neighboring nodes. This mechanism is not focused on a certain type of transfer; rather, it focuses on local congestion detection and information transfer to all neighbors.

However, before the decisions of the MAC sub-layer, on the previous layer (network), there have been numerous proposed solutions to congestion and energy-aware routing in LLNs, which can prevent congestion and energy loss in the nodes by dividing the routing load through optimal routes gained by exploratory and ultra-exploratory methods [4].

B. Popular congestion control methods

1) Congestion control MAC protocols

The IEEE 802.15.4 protocols that are originally designed for Wireless Personal Area Networks (WPAN) with low transfer rates can also be used for the internet of things and wireless sensor networks [25]. This protocol uses multichannel communication to reduce the interference effect. This interference occurs due to the presence of neighboring network communications using the same frequency spectrum. The protocol has two working modes: Beacon-Enabled and Beaconless. In Beacon-Enabled mode, the FFD nodes are responsible for channel matching; in the channel of communication the RFD nodes must seek the status of the channel from the FFD nodes (that are the coordinator) so in the absence or possibility of congestion, they can exchange data. In this case, scheduled communication is occurring in a star network such as single Hop network. Even if a node wants to communicate with its counterpart node within its radio range, all information must be provided through the coordinating node. However, when the protocol operates in Beaconless mode, the CSMA / CA method is used, and the nodes operate in a fixed channel [26]. Due to network hierarchies, coordinators are also responsible for scheduling, routing, and connecting new nodes to the network. In addition, since all nodes in this standard, exchange information in the same channel, the problem of node competition in the network is not resolved.

MC_LMAC has presented a single radio-multi channel MAC protocol based on scheduling with the aim of increasing network throughput. In the initial phase of this protocol, the nodes are synced with the parent node in a tree structure, and during network operation, whenever there is a discrepancy in time scheduling of the nodes, the scheduling operation is redone. Therefore, with the increasing number of nodes, the control messages overhead increases, which reduces network performance [27].

Rainbow is also a tree-based protocol suggested for data collection with high reliability [28]. In this protocol, local TDMA and Frequency Hopping Spread Spectrum (FHSS) techniques are used in tandem to lower collision, increase throughput, and avoid foreign radio wave interference. The main downside to this protocol is the high overhead of different control messages for channel allocation and tree making.

The Control multi-channel MAC is an asynchronous multi-channel protocol that uses two radios: one always awakes and is used for awakening the nodes called the LR, and the other is used for receiving and sending data consecutively called MR. Although the CMAC does not require synchronization, it uses an additional radio, which increases node costs and leads to higher costs in network installment. Besides, the additional radio, which is always on, leads to the higher energy consumption of the sensor node [29].

In [30], the single-channel MAC protocol is proposed using a grid limit to save on energy. Although this protocol tries to boost network longevity by increasing nodes' sleeping time, in high traffic, it leads to higher collision and frame resubmitting and hence higher energy consumption in the network because it only utilizes one channel. Besides, allocating different grid sizes to network nodes is a challenge in this protocol.

In [31] the Tree-based Multi-Channel Protocol (TMCP) is introduced, which is not applicable in sensor networks with high energy costs or sensitive to events where its nodes must listen to the wireless media for long periods or transfer data [32]. Due to the partitioning in the network, this protocol is unable to send broadcasting packets. Since nodes connect through one channel, competition and interference within tree branches is still an unsolved problem. Furthermore, there is no aggregation in TMCP because the connection between nodes in different branches is congested. Other MAC algorithms that benefit from being multichannel like Multi-frequency Media access control for wireless Sensor Networks (MMSN), Hybrid TDMA/FDMA Medium Access Control (HyMAC) and Energy-efficient multi-channel MAC protocol (YMAC) have been discussed in previous researches [17], [33]. Either the mentioned protocols need node synchronization, or in case of avoiding sync overhead, node energy will not be efficiently used [34].

One of the best methods for network routing in wireless sensor networks is the Minimum Cost Forwarding Algorithm (MCFA) [35]. In order to select the appropriate next step, the MCFA algorithm first explores the available space and selects the value of each path by each neighboring node. In this case, each node knows which path to choose and how much each path will be worth. One of the main problems with this method is the initial processing and computational overhead of the value of its single-hop neighbor nodes, which creates the overhead routing tables. In fact, in the MCFA method, cost calculation and generating optimal paths must be completed before starting the nodes sending-receiving operations. Initial preprocessing to find the optimal paths increases the computational overhead; it is, in fact, observing the environment, which will also result in increased energy consumption by the sensor nodes as well as wasting time.

Another method is the Congestion Avoidance, Detection, and Alleviation (CADA) where the congestion level is determined by buffer aggregation and the tree-based channel distributed CC algorithm, which leads to higher operational power, energy consumption and end to end delay, therefore it's only practical in application-based topologies [36], [37]. The Flock-CC protocol guides the packets to the sink for grouping and routing. Robust against failing nodes, this method is self-organizing and energy-efficient but fails to offer an appropriate load-balancing capability.

In Event-to-Sink Reliable Transport (ESRT) protocol, the nodes adjust their transmission rate based on the sink feedback and the routing reliability or via congestion detection. When reaching a specific buffer threshold, each node sets the congestion notification (CN) bit in the packets. The sink periodically calculates a new report rate based on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

reliability estimates, the CN, and the previous reporting rate and transmits it at its maximum radio output power [38]. The ESRT execution can operate in five different modes: non-congestion with low reliability (NCLR), non-congestion with high reliability (NCHR), congestion with high reliability (CHR), congestion with low reliability (CLR), and optimum operational region (OOR). In NCLR, the reporting rate increases to provide acceptable reliability, whereas a reverse trend is considered in NCHR and CHR. In CLR, the reporting rate decreases at a higher speed. In OOR, the load reporting rate does not change in the next decision area. Attempting to solve the congestion problem of different regions (event priority, node density) through similar methods reduces the throughput. Further, the ESRT protocol does not factor in the interferences [39].

2) *Congestion control routing and clustering protocols*

Another approach to reducing congestion in the classic Internet of Things (IoT) networks and wireless sensor networks (WSNs) is the use of node clustering, where clusters of existing nodes are generated in an attempt to minimize dispersion of data and facilitate quicker data exchange among the sensor nodes and the base station [18]. Evidently, network management can be facilitated by grouping network nodes as clusters, raising the question of what clustering methods and cluster sizes can be used to prevent congestion at cluster heads. In other words, optimal clustering of network nodes should be looked at as a challenging problem, although it might seem unimportant and non-critical at the beginning. However, the network will definitely start to struggle as some cluster centers are removed over time due to low energy. In clustering, each cluster head is constantly working, and, in fact, when two nodes attempt to transmit data to the cluster head, one of the packets will definitely drop due to the cluster center being able to receive only one data packet at a time. As such, a novel combined clustering approach has recently been introduced for WSNs which is significantly different compared to the classic clustering methods. The selection of cluster centers is the main difference between hybrid and simple clustering methods [40]. In the hybrid method, any node can be assigned as the cluster center as long as it has a central position compared to the other nodes in the cluster. The increased input and output traffic of the cluster centers close to the sink is a serious concern in both classic and hybrid clustering techniques. In other words, balancing the load and traffic during communication between the network cluster heads and the sink is a challenging task that has been addressed by some recent studies. There are still unresolved issues for data exchange in combined clustering.

In [23], a congestion-aware HOPbyHOP routing protocol is proposed to provide an appropriate efficiency in multi-sink networks. In this method, a traffic-aware routing plan with the ability to regulate the transfer rate of network nodes is designed, which effectively manages and controls the communication between nodes and the sink. This method uses the normalized depth and traffic of network nodes to balance the flow between sensor nodes. Another part of the study proposes a model for improved control and allocation of weights to traffic-balancing routing cost and congestion window, ultimately enhancing the performance of this method compared to its counterparts such as the ESRT [38], CODA, GRAdient-based Traffic-Aware routing for wireless sensor networks (GRATA) [41], and Shortest Path First (SPF).

3) *The back-pressure congestion control method*

The back-pressure method works based on local data and node decisions for routing [42]. In fact, this method is not meant to select a specific route at the beginning or even the start of transmission, but rather helps each node decide to which node a packet should be transmitted for the subsequent delivery to the sink. The back-pressure method follows a local approach in which each node selects the next data transmission node based on the table of neighbors and knowledge of their queue state. In fact, each node requires a list of its neighbors and information, such as the distance, queue length, and the cost function output. Despite the significant growth in its application in WSNs due to its features, this method suffers from a serious problem known as the loop trap. That is, the information between two valuable nodes may be directly or indirectly interchanged in a loop, leading to packet loss and expiry of data freshness, in which case the packets may not reach the sink or introduce a high delay in the reception route. To tackle this problem, a specific metric is used in the proposed method in this study, known as the CCFDM method, to eliminate the loop trap in the network.

The congestion detection and avoidance (CODA) protocol is the first partial development of congestion detection and prevention in WSNs [43], [44]. This protocol features three main components, namely receiver-based congestion detection, open-loop hop-by-hop back-pressure, and closed-loop multi-source regulation. In the open-loop hop-by-hop back-pressure, the sources take into account the current congestion state. Similarly, in the closed-loop multi-source regulation, the source receives acknowledgments (ACKs) from the sink, which is stopped once congestion occurs. This protocol controls the packet flow rate based on the Additive Increase Multiplicative Decrease (AIMD) algorithm. Although energy-efficient, this technique does not guarantee the successful delivery of the packets to the destination. In the received back-pressure signals (i.e., signals received by the source node from the middle node), the nodes control their packets based on the congestion parameter.

III. Proposed CCFDM Method

A. *Assumptions of the proposed network*

Due to the limited resources in classic IoT networks and WSNs, shorter communications can increase network lifetime and reduce the hidden terminal rate and crosstalk in the wireless medium [15]. To identify the challenges involved in hop-by-hop routing methods connecting nodes and the sink, a sample network is assumed on which our proposed method is implemented. In the CCFDM method, the surrounding environment of the sink node in the network is considered for the implementation of the layering decisions. This is because, in the decision process for the delivery of data packets to the sink, a structural distinction should be considered in the fuzzy decision calculations for prioritization of nodes closer to the sink. Utilizing concentric circles can divide the network into multiple node groups to facilitate the decision-making process.

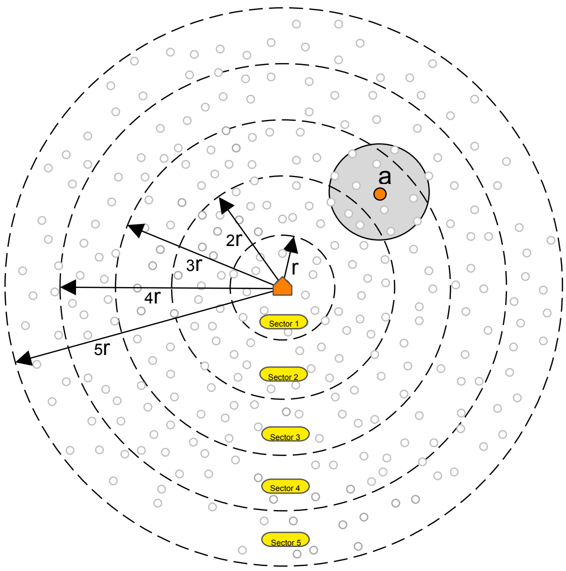

FIGURE 1. **Dividing the sink via concentric circles.**

According to the literature and studies on classic IoT networks, WSNs, and low-power and lossy networks (LLNs), the funneling effect is the main cause of early energy drain and increased congestion and collision. For instance, Fig. 1 shows a sensor network with a sink at the position of $(\frac{x}{2},\frac{y}{2})$. The spherical transmission of signals from the sink node is represented as concentric circles in the 2D network.

In order to make this issue clear, it should be noted that we have used concentric circles for the following reasons:

- To classify nodes and avoid wondering of network packets (because if all nodes in the network have the same priority, the packet will not be justified to reach the sink node. Over time, the status of nodes farther away from the sink will have better resources and the network packets will be led outside of the network due to the funnel effect).
- Weights can be entered at different stages using different coefficients in the decision system for each parameter involved. As we mentioned in the article because the (potential) congestion and resource consumption of nodes in different areas of the network are different (vary), we must somehow be able to generate accurate values, thus we have used concentric circles for this purpose.
- Another work was to correlate the weights of each parameter in each layer of concentric circles with the density of the nodes referred to in Formula 1 and Fig. 5.

In fact, the distribution of network nodes in different situations, whether it is preset (or pre-defined such as grid network) distribution, random or Poisson, has a direct effect on the congestion rate and network efficiency. For this purpose, in equation (1) it is assumed that in the best situation the network (like grid distribution) will also have congestion and the funnel effect. Since we have congestion in the normal or ideal case, we will also have congestion in any random situation, and this has been the basis of our research. We have noted that in any case, congestion is an inseparable effect in the sensor network. So, we have proposed our idea of assuming the existence of congestion in the network with different distributions.

Another issue is that the location of the sink node in the network can also be random. In-network assumptions, the radio range is the distance from the sink node to the end of the network environment. That is, in the worst case, the sink node is located at the point (x, y) = (0, 0) (Corner of the network area). Suppose the network is a square (each side is 1000 meter) the radio range in this network will be 1414m (calculated by$\sqrt{x^2+x^2}$). So, in this case, only the number of concentric circles can be increased, and all weights can be calculated by equations mentioned in figure further on in this article (Table 2 and Figure 5).

Given its unlimited signal power, the sink node can directly deliver its message to any sensor nodes across the network. Conversely, various constraints are involved for packet transmission from network nodes to the sink. The transmission range of sample node ***a*** is 1.5 times greater than radius ***r***. Thus, considering the limited transmission range and energy of the nodes, it is impossible to directly transmit packets from sensor nodes to distances beyond the transmission range of each node. Even if possible, packets will be lost due to the problem of the hidden terminal, meaning that direct data transmission from a node to the sink is unfeasible. In our proposed model, the network nodes are divided into different sectors based on the sink node to facilitate the routing decisions in sensor nodes during transmission of packets to the sink. Fig 1 provides an example of the importance and necessity of congestion issues in 6LowPan networks. According to previous research and documentation, it is rarely possible for researchers to substitute sensor nodes or the Internet of Things in a proportionate and balanced density environment, and almost all applications of these types of networks, nodes are randomly distributed in the environment with different distributions, including Poisson, etc. We will prove that the 6Lowpan network will still face the problem of node congestion for those near the base station, even if the nodes are in the best condition in terms of distribution and alignment in the environment. In the main issue of this article, the network nodes are randomly distributed in the network environment, and certainly, in equation 1, the number of packets passing through the node ***a*** (in figure 1) will be more or less than this threshold because of the density of the environment is different. However, to gain a better understanding of the topic and the significance of traffic management and congestion control in the nodes close to the sink, the traffic load (Funnel Effect) is calculated by the following equation: Assuming a

network with the sink node positioned as the center $(\frac{x}{2}, \frac{y}{2})$, the following relations hold:
Hence, assuming each sensor node uses a mean of $\boldsymbol{x}$ data units to transmit a report, it can be stated that in a network consisting of $\boldsymbol{n}$ sectors, a node in sector $\boldsymbol{i}$ is tasked with transmitting and directing a total of F_i (Funnel Effect) data units:

$$F_i = x + \left(\frac{2i+3}{2i+1}\right) F_{i+1} \quad , \quad i = 0, \ldots, n, \quad F_{n+1} = 0 \qquad (1)$$

Area of sector 1 is $\Psi_1 = \pi r^2$, using which the area for sector 2 is obtained as $\Psi_2 = \pi(2r)^2 - \Psi_1$. By applying the same procedure to other areas, the general relation for the area of the Ψ_i the group is expressed by $\Psi_i = (2i+1)\pi r^2$. The ratio of area for sector 1 to sector 2 is calculated as $\frac{\Psi_2}{\Psi_1} = 3$. Therefore, the general relation for this ratio for the *(i+1)th* group to the i^{th} group obtained as $\frac{\Psi_{i+1}}{\Psi_i} = \frac{2i+3}{2i+1}$, indicating that a sensor node in the i^{th} group on average transmits the traffic of $\frac{2i+3}{2i+1}$ nodes from sector *i+1*. For example, if each node requires transmitting x data units for its report, then a node in sector 3 (Fig. 1) is responsible for sending $\frac{9x}{7}$ data units from nodes in sector 4 in addition to the x data units of its own, ultimately amounting to a total of $\frac{16x}{7}$. Similarly, a node in sector 2 must transmit x data units of its own as well as $\frac{16x}{7}$ data units from sector 3. Accordingly, the closer a sensor node to the sink node, the higher its traffic, which is also known as the funneling effect, as shown in Fig .2. The blue, yellow, orange and red nodes are engaged in the transmission of light or transient traffic, moderate traffic, heavy traffic, and extremely heavy traffic, respectively.

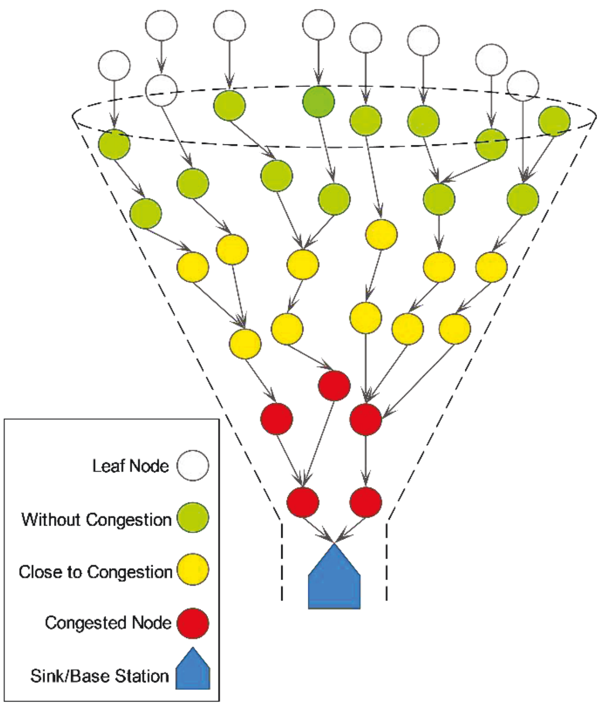

FIGURE 2. The funneling effect in a network.

1) Variable of this paper:
A list of variables used in the paper is shown in Table 1.

Table 1. List of variables used in the paper.

Variables	Definition
Root	The root node of the graph (Sink)
G	Graph
V	V is a set of vertices
E	E is a set of edges
S	Set of nodes
Ψ	Area of the Region
r	Radius
F_i	Traffic rate in funneling effect
T_x	Transmission power
R_x	Reception power
T_s	The time period of the sensing operation
V_s	Voltage of sensors
$\bar{D}_j^\tau$	Local load of the neighbors
τ	Time interval
$\bar{N}_j$	The traffic load or packet congestion node N
α	An exponential value
D_{jk_q}	Euclidian distance between node j and node k
ETX	Expected Transmission Count
Df	Delivery forward
dr	Delivery reverse
ACK	Acknowledge message
w_i	Weight of rule *i*
U_i	Total cost of parameters
V_i	Final Fuzzy Value
P_A	Current probability of node A
P'_A	Next time probability of node A
$\varphi\Delta t$	Queue input rate
$\mu\Delta t$	Queue output rate
K	Constant parameter
λ	Exponential distribution
ALTN	Average Life-Time Network
PDR	Packet Delivery Ratio
M	Number of alive nodes
N	Total number of network nodes
T	Predefined network lifetime
JFI	Jain's fairness index
PE	Power efficiency

B. Quality of Service-aware routing

The primary concern in routing is to select the next proper hop in the network, given that each link consists of one or more connections between sensor nodes. The state of each node depends on various parameters and, therefore, making an optimal choice appropriate for all situations is a complex problem. The proposed in this study seeks to select an optimal hop to create an appropriate link between nodes to the sink. The quality of a link has resulted from the combined quality of its connecting nodes and, additionally, numerous quality-of-service (QoS) parameters are present in a network.
Based on the studies conducted in the context of this paper, in discussing the quality of 6LowPan network services, particularly in the field of communication and routing, we have encountered various metrics which are mentioned below, and the reasons for using or not using them are also discussed: The metrics of effective quality of service in communications between low-power and wasteful networks and wireless sensor networks fall into two general categories [45].

- Link metrics:
    - Throughput: This indicator refers to the average successful link rate of a link and is one of the key factors in identifying, controlling, and managing congestion in the network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

  - SNR and LQI Indicators: These indices are included in the physical layer and evaluate the quality of the link in terms of signal rate, frequency and voltage. Both of these metrics are used to compute the hardware communication layer of the node, from the proposed method scope that starts from the 802.15.4 MAC sub-layer to the Application layer and does not include 802.15.4 Phy computations.
  - ETX Index: This metric indicates the degree of link reliability and expected transfer. This parameter is calculated and evaluated in the MAC 802.15.4 sub-layer based on the MAC acknowledgment message in the MAC sub-layer [2].
- Node metrics:
  - Energy Index: The energy consumption rate of the node in the network varies due to the random distribution of network nodes in the environment and the dependence of the network energy consumption rate on the distance and amount of node activity in the exchanges. So if this indicator is ignored in routing process, there will be an early death of nodes in the network, which is a bottleneck. For this purpose, the design of objective functions in the 6LowPan network energy factor should be considered [45].
  - Number of steps: The most common indicator in calculating the path length of a wireless network is the number of steps to the destination. The main disadvantage of this indicator is that it finds the shortest route and offers no guarantees in terms of route quality.
  - End-to-end Delay: One of the key criteria in route construction and objective functions in the 6LowPan network is the node-to-destination delay criterion.

In this study, we consider the above factors except for RSSI and LQI as indicators of quality of service regarding the network nodes connections because they are evaluated in the physical layer and are outside the scope of this paper. On the other hand, calculating and considering more indicators and metrics in the proposed algorithm can increase the algorithmic complexity and become a negative factor in achieving quality of service. Finally, the energy, traffic, ETX and delay parameters have been considered directly, and the step parameter has been considered indirectly.

Therefore, this study used a fuzzy decision system to generate stable, proper network links. As indicated by single-hop routing protocols, the source node mainly attempts to select the best node as the next hop from its accessible neighbors. However, greedy selection based on parameters such as energy, traffic load, ETX rate, and delay can cause problems in other QoS parameters. Generally, these factors are combined as a proper solution to this problem. That is, combining multiple node parameters in a weighting system can produce better results compared to the greedy method. For this purpose, a multi-criteria decision system was employed in the proposed CCFDM method in this study to combine and allocate weight to factors. According to Fig. 3, the parameters of remaining node energy, traffic rate, ETX rate, and link delay rate are used as the inputs of the multi-factor fuzzy decision system.

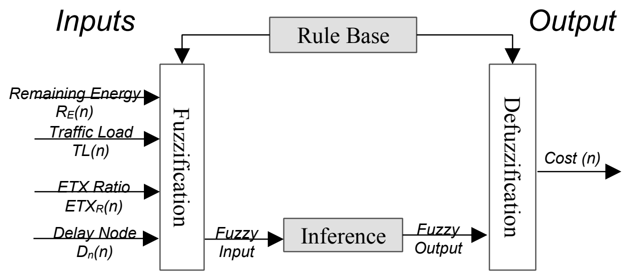

FIGURE 3. The proposed multi-criteria fuzzy decision system.

In the routing phase of CCFDM, selecting the next hop for packet transmit via a node depends on the following parameters:

1) Remaining energy (first input of the fuzzy decision)

The energy consumption model of the study is based on the work of [46] in which an energy consumption model is proposed for sensor nodes under different modes, including the processor, radiofrequency, and the sensors. The processor parameter is in charge of controlling the node, communication protocol, and data processing. Microprocessors usually support three operating modes (sleep, idle, and execution). As shown in Eq 2, the energy consumed by the processor (E_{cpu}) comprises the steady-state energy consumption $E_{cpu-state}$ and the operation mode change $E_{cpu-change}$ energy consumption.

$$E_{cpu} = E_{cpu-state} + E_{cpu-change} \tag{2}$$

In a node, the communication parameter consists of the baseband and the radio frequency and is responsible for receiving and transmitting node data. The transmitter/receiver normally has six modes: T_x or transmission, R_x or reception, OFF, idle, sleep, and CCA/ED or booting up. The energy consumption of the transmitter/receiver (E_{trans}) is equal to the sum of steady-state energy consumption of the processor ($E_{trans-state}$) and the energy required for its state change ($E_{trans-change}$). The parameter $E_{trans-state}$ is given by:

$$E_{trans-state} = E_{TX} + E_{RX} + E_{Idle} + E_{sleep} + E_{CCA} \tag{3}$$

The parameter $E_{trans-change}$ is calculated from:

$$E_{trans-transition} = \sum_{j=1}^{n} N_{trans-change}(j) e_{trans-change}(j) \tag{4}$$

Where $\boldsymbol{j}$ (j=1,2,…,n) is the type of state change, $\boldsymbol{n}$ is the number of state changes, $N_{trans-change}(j)$ is the frequency of type $\boldsymbol{j}$ state change, and $E_{trans-change}(j)$ is the energy consumption during a single state change of type $\boldsymbol{j}$. The sensor component consists of the sensors and the digital-to-analog

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

convertor tasked with collecting data and digital conversions. The energy consumption of the sensor component is the result of multiple operations, including signal sampling, analog-to-digital signal conversion, and signal modulation. In [46], it was assumed that the sensor component works periodically and the sensors open and close periodically, corresponding to the "on" and "off" modes, respectively. Assuming a constant energy consumption for open and close operations, the energy consumption of sensor E_{sensor} is given by:

$$\begin{aligned} E_{sensor} &= E_{on-off} + E_{off-on} + E_{sensor-run} \\ &= N(e_{on-off} + e_{off-on} + V_s I_s T_s) \end{aligned} \quad (5)$$

Where e_{on-off} and e_{off-on} denote the energy used in a single sensor switch-off and switch-on, respectively. Moreover, $E_{sensor-run}$ is the energy consumption of the sensing operation, $\boldsymbol{V_s}$ and $\boldsymbol{I_s}$ are the operating voltage and current of the sensors, respectively, $\boldsymbol{T_s}$ is the time period of the sensing operation, and $\boldsymbol{N}$ is the number of switch-on and switch-off operations.

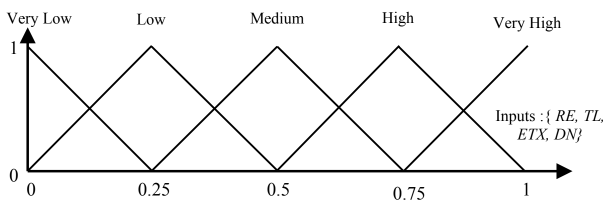

FIGURE 4. Fuzzy diagram of the inputs at different levels.

Based on the conventional fuzzy system, a triangular model wherein each crisp input parameter corresponds to two relative fuzzy outputs. In Fig. 4, the input parameters of each node at the network, which may lie in two of the five existing levels, namely very low, low, medium, high, and very high.

2) Traffic load (second input of the fuzzy decision)

In the point-to-point (P2P) communication traffic model, packet loss in the network occurs on the node level and communication link level. The constraints of the communication channel of a wireless medium and those of the node influence each other. This study assumes the environmental and communication signal noises to be negligible. Hence, since the limitation of network nodes are queue size, buffer, reception rate, storage, and processing [47], the second input of the fuzzy decision model is considered to be the traffic load of the candidate node. The higher the traffic load, the stronger the chances of collision in the node. As such, this parameter attempts to prioritize the candidate node with less traffic load in order to reduce collision. This paper defines the traffic load of the candidate node as the combination of neighbors' local load and packet traffic load:

$$nd_j^\tau = \frac{1}{(\bar{D}_j^\tau \times \bar{N}_j)^\alpha} \quad (6)$$

Where $\boldsymbol{\bar{D}_j^\tau}$ is the inverse of the local load of the neighbors, indicating the mean Euclidian distance of the neighbors from Node $\boldsymbol{j}$ within the time interval $\boldsymbol{\tau}$, and $\boldsymbol{\bar{N}_j}$ is the inverse of the traffic load or packet congestion, indicating the mean total data packets received from the neighbors within time interval $\boldsymbol{\tau}$. The node traffic load is considerably smaller than the delivery ratio and packet progress and is dominated by them in Eq. (7), for the prevention of which the exponent $\boldsymbol{\alpha}$ is used. The appropriate value for $\boldsymbol{\alpha}$ was obtained as 0.005 after numerous simulations. In what follows, the measurement procedure for $\boldsymbol{\bar{D}_j^\tau}$ and $\boldsymbol{\bar{N}_j}$ is described:

$$\bar{D}_j^\tau = \sum_{k=1}^{n} \bar{D}_{jk}^\tau \quad (7)$$

$$\bar{D}_{jk}^\tau = \frac{\sum_{q=1}^{\tau} D_{jk_q}}{\tau} \quad (8)$$

Where $\boldsymbol{n}$ is the number of neighbors around node $\boldsymbol{j}$, and $\boldsymbol{\bar{D}_{jk}}$ is the mean Euclidian distance between node $\boldsymbol{j}$ and its neighboring node $\boldsymbol{k}$ within the time interval $\boldsymbol{\tau}$. Because the beacon messages are transmitted periodically $\boldsymbol{D_{jk_q}}$ is the Euclidian distance between node $\boldsymbol{j}$ and its neighboring node $\boldsymbol{k}$ for each beacon message. In fact, this parameter is the inverse of the local load of the neighboring nodes. The higher the total distance of a node from its neighbors, the lower the traffic load of the neighbors and the lower the congestion around the node in question. The node with a lower traffic load also has a smaller channel access time. Moreover, increasing the distance between a node and its neighbors reduces the packet collision region for the node, consequently reducing its collision probability.

$$\bar{N}_j^\tau = \frac{\sum_{k=1}^{n} count_{j,k}^\tau}{n} \quad (9)$$

In this equation, $count_{j,k}^\tau$ is the data packet count that node $\boldsymbol{j}$ has received from its $\boldsymbol{k^{th}}$ neighbor within the time interval $\boldsymbol{\tau}$. Reduction in $\boldsymbol{\bar{N}_j^\tau}$ indicates increased collision and congestion in node $\boldsymbol{j}$.

3) Link ETX rate (third input of the fuzzy decision)

Another metric influencing the communication QoS of LLNs is linked to ETX [48], [49]. This parameter selects the link with the least expected transmission count for reaching the destination. It aims to find the link with a high packet throughput. Link ETX consists of the number of data transmissions required for transmitting a packet via the link, which also includes re-transmissions. The total ETX of a route is the sum of the ETX of its links. For instance, the ETX of a route with three ideal hops is 3 while this value is 2 for a single-hop route with 50% throughput. The ETX of a link is calculated from its transmission and reception rates. The delivery forward ($\boldsymbol{df}$) ratio of a transmitted packet is the probability of a data packet successfully reaching its destination. The delivery reverse ($\boldsymbol{dr}$) ratio is the probability of the ACK packet successfully received by the node sending the packet. The probability of an acknowledged successful transmission is then calculated by $df \times dr$. The transmitter re-transmits a packet if the packet sent in the previous time period is not successfully acknowledged (no ACK message for successful delivery is received). Since each attempt in sending

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

a packet can be assumed as a Bernoulli distribution, the number of expected transmissions is:

$$ETX = \frac{1}{d_f \times d_r} \tag{10}$$

The ETX factor is designed for protocols that send the ACK at the link layer. Therefore, to prevent re-transmissions, both directions of a link must function correctly. Note that ETX is, in fact, the mathematical expectation for the required transmissions (including re-transmissions) for delivering a packet. Accordingly, using ETX can give an estimate of the link loss ratio:

$$ETX_l = \frac{1}{(1 - d_f) \times (1 - d_r)} \tag{11}$$

Where ETX estimates the link loss ratio in each direction. If the link is asymmetric or unidirectional, then $\boldsymbol{d_r} = 0$.

4) Node delay rate (fourth input of the fuzzy decision)

The delay estimation model in classic IoT networks and WSNs consists of [50], [51]:
Link delay: Link delay in network nodes consists of the queuing delay in the access control layer besides the transmit delay. For ***n*** transmitted packets from node ***i*** to the parent node ***p***, this delay is given by:

$$LinkDelay_{ip}^{n} = QueueDelay_{i}^{n} + TransmissionDelay_{ip}^{n} \tag{12}$$

- *Queue delay*: The time between a packet's entry to the MAC layer queue and its removal is the queue delay. It also includes the transmit delay caused by the time required for successful packet delivery with an ACK notification from the receiver.
- *Process delay*: The processing task varies in network nodes depending on the hardware and software type. For instance, in a node, the packet is generated by the function layer and sent to the network layer for delivery to the MAC sub-layer. In a packet delivery forwarding node, after the reception, the packet is sent to the MAC layer and then the network layer. At the sink node, after receiving the packet, the data is given to the network from the MAC layer and then delivered to the function layer. As an example, for the node $\boldsymbol{i}$ generating packet ***n***, the following delays occur:

$$GenProcDelay_{i}^{n} = Delay_{L5L3}{}_{i}^{n} + Delay_{L3L2}{}_{i}^{n} \tag{13}$$

- *The forwarding delay:* see (13).

$$FwdProcDelay_{i}^{n} = Delay_{L2L3\_FWD}{}_{i}^{n} + Delay_{L3L2}{}_{i}^{n} \tag{14}$$

The key feature of the HOPbyHOP routing method is its successful performance under high traffic. For instance, regarding the traffic entering the network in a crowded environment, even if the data generation rate at the leaf nodes of the network graph is one packet per 10 minutes, the traffic rate generated in the nodes close to the sink increases to one packet per 0.5 seconds. Therefore, the delay parameter is considered as an appropriate candidate in the fuzzy decision system.

C. Metric weights

The principal rule in classic WSNs and LLNs is that each node according to its position in the network graph has a different weight. Thus, the energy of the nodes close to the base station or the sink should be optimally maintained not to jeopardize the connection between other network nodes and the sink. Further, considering the lack of node access to geographic data or other network map information, it is rather impossible to allocate weights to every single node. Otherwise, a high computation overhead is imposed along with the need for very large routing tables (which runs counter to the nature of LLNs). Hence, this study proposes a division in the form of the annulus to classify each node in a specific group. The weight of each fuzzy input metric varies according to the node position and in which sector the node is located. As expected and as noted in Figure 2 (funnel effect), the position of each node in the network due to the distance and number of steps to the Sink node can face challenges in achieving quality of service. These challenges including early energy discharge increased traffic throughput, increased fill and drop rate of node buffers and increased delay, and high rate of packet loss in the network, all of which occur when the node is close to the Sink and the traffic is very high. Therefore, we had to consider a balance between the parameters of the decision system in order for justice to be implemented among the nodes of the network. For example, the sensitivity of the network nodes to the remaining energy parameter near the sink and away from the sink would have to be different. Otherwise the network would face premature death of the traffic node. Therefore, the following experiments were performed and the simulations were evaluated:

- The weight of each parameter in the decision system is considered the same as any distance from the sink.
- The weight of each parameter is varying according to the calculations made (for example in the node located near the sink, the energy criterion will be of higher weight and importance than the node is located further away from the sink). For other parameters, depending on the distance from the sink, some of which are of higher or lower importance.

After several rounds of simulation, evaluation, and calculation of variance of the results, we concluded that the optimal rate of desired metric weights resulted in the following tables. This weight optimization results the improvement in all the tests performed in Section IV of this article. Finally, after inputting the numbers in the Excel sheet, we have obtained approximate functions and regressions with a high accuracy of 97% as shown in Figure 5. Parameter *R* is the estimated rate of term *Y* in each equation with actual data. That is, for energy, traffic load, ETX rate, and delay, each sector has a different optimum weight depending on the Y term for node ***a***. For each node, the coefficients for energy, traffic load, ETX rate, and delay are acquired in order for the fuzzy weight of the node to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

correspond to its network position. In table 2 we have shown the optimum weight of parameters in each sector.

Table 2. The optimum weight of parameters in each sector

Sector	Remaining Energy	Traffic Load	ETX	Delay	Sigma
1	0.35	0.25	0.25	0.15	1
2	0.33	0.23	0.27	0.17	1
3	0.3	0.22	0.28	0.2	1
4	0.28	0.21	0.29	0.22	1
5	0.25	0.2	0.3	0.25	1
6	0.22	0.19	0.31	0.28	1
7	0.2	0.18	0.31	0.31	1
8	0.18	0.17	0.31	0.34	1
9	0.17	0.16	0.3	0.37	1
10	0.15	0.15	0.27	0.43	1

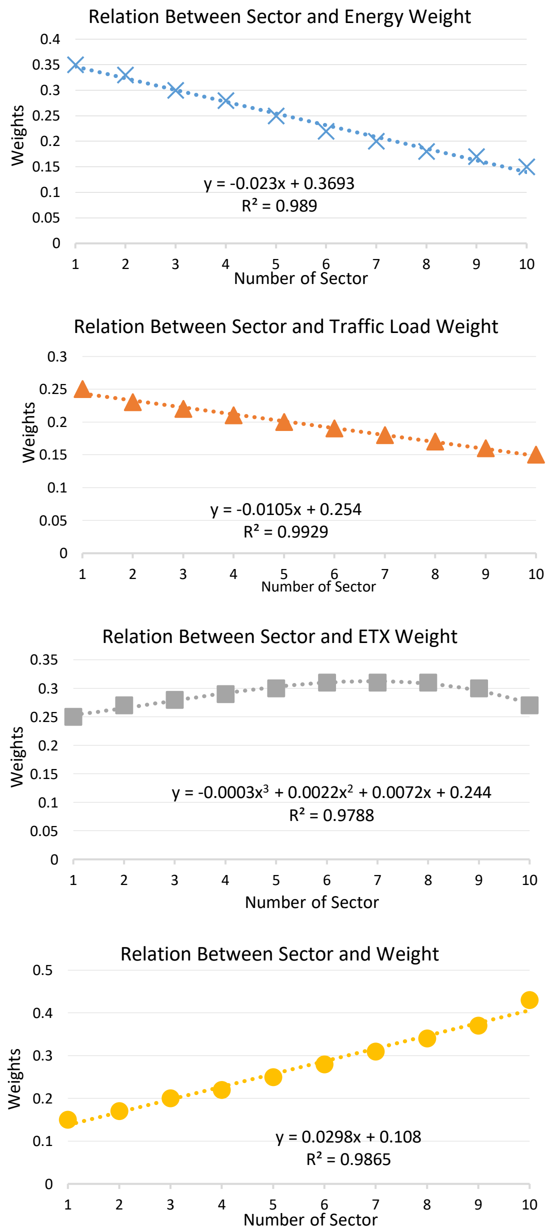

FIGURE 5. Fuzzy diagram of Allocation of weight to parameters of the decision system in each sector

D. Overall Weight of a Node

In the proposed decision system, one of the criteria is the minimum number of hops between the source and the network sink. This number is assumed constant due to the stationary attribute of the CCFDM network. In the proposed method, the sink node divides the network area to specific concentric sectors based on the CCFDM. The distance between each sector and the next is referred to as a hop. For instance, in Fig. 1, the sink is at the center of the network, and sectors are shaped like circles. Clearly, depending on the sink position in a network, each sector has a different number of nodes. The nodes in sectors farther from the sink normally have a higher fuzzy weight due to lower usage. Hence, they can attract network traffic considering in proportion to their value, which is considered as one of the challenges in weight-based routing. Since the packet is exchanged among the nodes with a high allocated weight in the fuzzy decision system, the packets are expected to flow at the network edges. The relative weights of indicators are multiplied by the matrix of relative weights for the candidates (based on each indicator) and the sum of the four parameters in each sector and for each node is calculated as follows:

$$U(i = 1 \dots . n) = \sum_{i=1}^{n} (Param_i \times w_i) \tag{15}$$

Distinguishing the rules as shown in Table 3, a total of 16 rules is obtained for the four input parameters.

Table 3. Each fuzzy rule and the unique respective c_i coefficient

U_i	c_i
$RE_0 \times TL_0 \times ETX_0 \times DN_0$	C(0000)
$RE_0 \times TL_0 \times ETX_0 \times DN_1$	C(0001)
$RE_0 \times TL_0 \times ETX_1 \times DN_0$	C(0010)
$RE_0 \times TL_0 \times ETX_1 \times DN_1$	C(0011)
$RE_0 \times TL_1 \times ETX_0 \times DN_0$	C(0100)
$RE_0 \times TL_1 \times ETX_0 \times DN_1$	C(0101)
$RE_0 \times TL_1 \times ETX_1 \times DN_0$	C(0110)
$RE_0 \times TL_1 \times ETX_1 \times DN_1$	C(0111)
$RE_1 \times TL_0 \times ETX_0 \times DN_0$	C(1000)
$RE_1 \times TL_0 \times ETX_0 \times DN_1$	C(1001)
$RE_1 \times TL_0 \times ETX_1 \times DN_0$	C(1010)
$RE_1 \times TL_0 \times ETX_1 \times DN_1$	C(1011)
$RE_1 \times TL_1 \times ETX_0 \times DN_0$	C(1100)
$RE_1 \times TL_1 \times ETX_0 \times DN_1$	C(1101)
$RE_1 \times TL_1 \times ETX_1 \times DN_0$	C(1110)
$RE_1 \times TL_1 \times ETX_1 \times DN_1$	C(1111)

The above rules are in the form of IF-Then rules wherein the relationship between fuzzy input and the output variables is described by the linguistic variables of each along with the fuzzy sets and the fuzzy operators.According to the standard fuzzy membership function, the value of each node in the fuzzy decision system can be obtained as:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

$$C(n) = \frac{\sum_{i=1}^{n} U_i \times c_i}{\sum_{i=1}^{n} U_i} \tag{16}$$

Ultimately, to calculate the overall node weight in each sector, after performing the calculations for the fuzzy decision-making system for the four metrics, the fuzzy value of node $\boldsymbol{n}$ is combined with the inverse of sector value $1/_{Sector_n}$. It increases the tendency or attraction of the packet towards the sink. The $\boldsymbol{V(n)}$ function determines the final node value at the next hop. The output of the fuzzy model (node value or $\boldsymbol{C_{(n)}}$) is given by:

$$V(n) = C(n) + (\frac{1}{Sector_n}) \tag{17}$$

The $\boldsymbol{V(n)}$ value is periodically broadcasted to the neighbors of network nodes to update their neighborhood tables.

```
MADM_Calculate (x_1, ..., x_i) |
 i is number of node's parametrs
  Set priority for each pair of node's parametrs
  Create priority_matrix[1...i,1...i] and inialize it
According to first line
  Create col_sum[1 ... i] and initialize it with 0
   for x ← 1 to i
     for y ← 1 to i
        col_sum[x] ← col_sum[x] + periorit_matrix[y, x]
   for x ← 1 to i
     for y ← 1 to i
       periority_matrix[x, y] ← periority_matrix[x, y]/
col_sum[x]
  Create AHP[1 ... i] and initialize it with 0
     for x ← 1 to i
       for y ← 1 to i
         AHP[x] ← AHP[x] + periority_matrix[x, y]/i
  Return AHP[1 ... i]

MADM_value (v_1, ..., v_i, AHP[1 ... i])
  for x ← 1 to i
   MADM_value ← MADM_value + (v_x × AHP[x])]
   Return MADM_value

MADM_Selection (u)
   compare[1 ... i] ← MADM_Calculate (x_1, ..., x_i)
     max_MADM ← 0
     for each v in adj[u]
   temp ← MADM_value(v_1, ..., v_i, compare[1 ... i])
          if temp > max_MADM
             then max_MADM ← temp
        return max_MADM
```

FIGURE 6. Pseudo-code of the multi-criteria fuzzy decision algorithm

The computations for the fuzzy decision system are according to Fig. 6 and lead to selecting the best available node. Clearly, the most valuable node in the single-hop neighborhood of the node has the most appropriate state considering the total parameters of remaining energy, traffic load, ETX rate, and delay rate.

E. Congestion detection and control via the back-pressure method

Literature studies suggest that congestion occurs when the buffer or queue of the intermediary nodes reaches its full capacity. In classic IoT networks and LLNs, congestion control is generally conducted in two ways: the use of preventive algorithms (congestion prevention), or the use of congestion regulation algorithms after congestion detection. Congestion detection may be applied to the source or intermediary nodes. However, congestion control is performed by reducing the transmission rate at source nodes. In the present study, the queue state indicator is used to detect congestion. In a sensor node, the node queue rate for buffering, processing, and packet transmission are known, constant value. Hence, if the queued input and output rates do not match, the node buffer overflows. For instance, in the proposed network, each node is assumed to have a buffer capacity of 20 messages. This queue capacity for network nodes is then divided into two equal parts to be able to assign a different significance to each. Since the number of packets in a node queue indicates its traffic load in the network, the packets in the second half of the node queue are more important as they can lead to congestion. Node queue traffic is given by:

$$Queue\ Traffic = \frac{\sum_{i=1}^{10} queueingPacket(i) + \sum_{i=11}^{20} 2 \times queueingPacket(i)}{15} \tag{18}$$

The output of Eq. (18) gives the node queue state. If larger than 0.7, the node will probably face congestion. Hence, to use this relation for congestion probability, the node queue state should be periodically checked (every second). As such, the queued input and output rates must be computed so that in the first hop, the node is notified of the event, and in the next hop, it can warn neighboring nodes to reduce or regulate the rate of the traffic transmitted to it. This process is known as the back-pressure technique, the main advantage of which is finding low-traffic routes for data transmission. In each data forwarding, the size of the transmit queue of the neighboring node and the route traffic is checked so that the route with the least traffic can be selected. The other competitive edge of the back-pressure method is the real-time routing decision of the system and the network nodes, such that the nodes do not need to store previously-traveled routes and can lead the packets to the sink according to the current traffic pattern, consequently reducing time delay. In this method, the node queue input rate $\boldsymbol{\varphi\Delta t}$ and the node queue output rate $\boldsymbol{\mu\Delta t}$ are compared every second using the following equations:

$$(\frac{\varphi\Delta t}{\mu\Delta t}) \leq 1 \tag{19}$$

$$(\frac{\varphi\Delta t}{\mu\Delta t}) > 1 \tag{20}$$

In the first mode, the probability of congestion is low, while that of the second mode is high. In the second mode, the difference between the packet input and output rates depends on the queue used as the reference for comparison. If $\varphi\Delta t - \mu\Delta t$ is smaller than the value of the remaining queue slots, the node will not send a beacon. However, if this difference is larger, the node transmits a beacon to its single-hop neighbors in order to reduce its traffic attraction from the said neighbors to 50% of the previous attraction. If the node again reaches congestion and a high probability of queue buffer overflow, it sends a beacon to the neighbors and requires them to reduce the probability of its election in their decision system by 50% for the time period $\boldsymbol{\tau}$.

For example, Fig. 7 shows nodes ***C***, ***D***, and ***E*** transmitting data to the sink node via node ***A***. However, this node faces congestion in its queue and thus sends a backward message containing its packet attraction probability (50% of the standard flow) to all nodes from which it currently receives packets. If after the time period $\boldsymbol{\tau}$ the output of Eq. (20) is still less than $\varphi\Delta t - \mu\Delta t$, then the node will face congestion in the near future. Thus, node ***A*** transmits the $P'_A = 70\%(P_A)$ beacon to nodes ***C***, ***D***, and ***E***. Each node stores and updates the value of the fuzzy decision systems of its neighbors in its neighbor list. Clearly, after receiving the beacon message, the updated probability of node ***B*** in nodes ***C***, ***D***, and ***E*** is multiplied by 0.7 so that node ***A*** will have a lower probability of selection among its neighbors within the period $\boldsymbol{\tau}$.

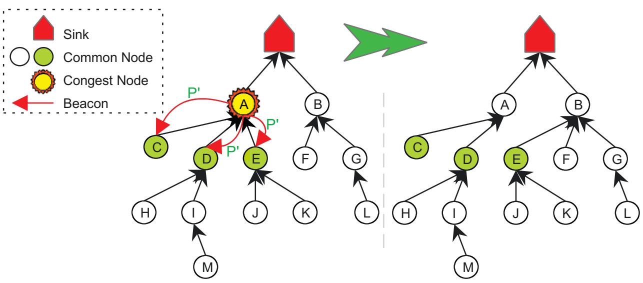

FIGURE 7. **An example of broadcasting and updating node participation probability in routing**

The parent-change mechanism in the proposed CCFDM is as follows: the child nodes, with the knowledge about the routing metric and queue state, separate themselves from the congested parent node and select a parent with a better state. Accordingly, due to the sudden migration of network nodes between parents, the new parent node itself may experience congestion and traffic load. To prevent this mass effect, a probability criterion for admission is used for parents:

$$Probabilistic\ Forwarder\ Change = max\{(\kappa(P_i)) - (P'_i))\ ,0\} \quad (21)$$

For example, as shown in Fig. 7, $\boldsymbol{P_i}$ is the node weight or the selection probability of the current node ***A***, and $\boldsymbol{P'_i}$ is the probability of selection of node ***B*** as the alternative node for the next hop in the neighborhood list of node ***E***. Parameter ***k****=0.7* is a positive coefficient, indicating at what probability the nodes transmitting messages to node ***A*** should change the weight of their decision system with regards to ***A***. In other words, in the worst-case scenario, node ***E*** changes its parent node with a probability of $\{(P_i) - (P'_i)\}$ and, in the best-case scenario ($\{\kappa(P_i) - (P'_i)\}$), 30% of the nodes sending packets to node ***A*** stop sending traffic, consequently reducing the load experienced by node ***A***.

IV. Simulation

The simulations in this study are implemented on the base routing protocols in an environment with randomly distributed sensor nodes and the sink node at the center. The simulation is conducted in NS2.34, and the results presented for 10 reruns of the algorithms and variance calculation are calculated with respect to the test type. To proceed with the simulation, different parameters have been considered. A total of 200 homogeneous nodes have been randomly distributed which include source and sink nodes. The region of the distribution is a square of the size 1000 x 1000 m having one sink centered inside the area of distribution. The communication radius of each node is 120m which is called an active coverage area of each sensor node. When constant bit rate traffic and burst traffic are concerned, the transmission rate will be 250 kbps. The channel congestion window size varies from 1 to 63. 10 data packets, each of the size 50 Bytes, Can be in the queue which indicates the buffer size. What has been used as the standard at the MAC and physical layers is IEEE 802.15.4 standard. The traffic information collected from the neighbors of each node is preserved retained periodically by the node using the Fuzzy Decision system and the proposed algorithm is applied to send (forward) the packets to the sink using the optimal paths.

TABLE 4. **Simulation parameters.**

Parameter	Value
Number of nodes	200 Homogeneous sensor nodes
Distribution of nodes	Randomly
Network size	Square region of 1000 × 1000 m
The nodes' radius	120 m
The transmission rate	250 kbps
Traffic Type	UDP (CBR)
Traffic	Constant bit rate traffic / burst traffic
CW size	1-63
The total buffer size	10 data packets
Size of packets	50 Bytes
MAC and PHY Layer	IEEE 802.15.4 standard version
Back-pressure type	Send probability packet to upstream sources'
Simulation time	1000 second

A. Average network lifetime test

This parameter is used to evaluate the lifetime of the proposed protocol with respect to its counterparts in order to select an efficient route and congestion control in the network. Smart application and proper use of the fuzzy decision system and factoring in the effective parameters balances the energy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

consumption by network nodes. As a result, the topology of the network nodes is more stable compared to other methods, allowing appropriate network links to lose energy over a longer period. One of the solutions for calculation of average network lifetime is the criterion of the first node death. The longer it takes for the first node to die, the higher the efficiency of the solution in balancing and resolving the problem of energy consumption bottleneck (hotspot). The first node death and average network lifetime (ALTN) are defined as [52]:

$$ALTN = \frac{\sum_{i=1}^{N-m} t_i + (m \times T)}{N} \quad (22)$$

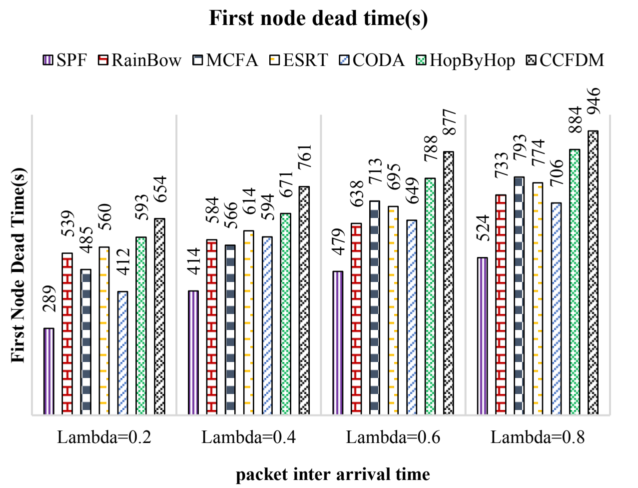

FIGURE 8. Diagram of first node death time

Where t_i is the death time of the i^{th} node, N is the total number of network nodes, m is the number of alive nodes at the end of the simulation, and T is the predefined network lifetime. Figure 8 shows that compared to other approaches, the first node death takes longer to occur in the proposed CCFDM protocol at the packet inter-arrival time of $\lambda = 0.2 \sim 0.8$ per second.

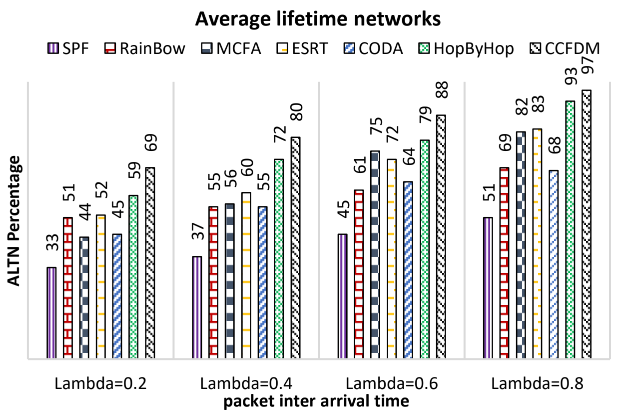

FIGURE 9. Diagram of average network lifetime

By maintaining a balance in selecting the next hop with respect to the other network nodes, the slope of energy consumption is further reduced at energy hotspots. In normal routing, mostly the routing parameters are used for selection of the next hop, which cannot alone guarantee to preserve the network links. Hence, in the proposed method, the parameters of remaining energy, traffic load, ETX rate, node delay, and packet attraction (min hops) were used to intelligently select the node with the minimum remaining energy at each sector. Accordingly, the first node death time in the proposed method takes longer to occur compared to the other protocols. Since the primary factor of calculating network lifetime is the first node death time, the average lifetime of network nodes is considerably higher for the proposed method compared to the other methods, as shown in Fig. 9.

B. Packet delivery ratio test

This parameter is used to evaluate the number of packets transmitted from the source node and soundly received at the destination. Avoiding congested routes can increase the packet delivery ratio. In the proposed method, selecting the proper route before transmission and updating route information can increase this ratio. Further, the use of a fuzzy decision system and activity of nodes participating in the route logically increases the probability of link repair, consequently making it more likely for the transmitting nodes to reuse the route selected in the previous cycle. Besides, using link quality parameters as the key metrics with a high weight in the proposed evaluation system leads to the formation of stable links with a minimum packet loss (Figure 10). Another reason contributing to the improved results is the consideration of queue state in the back-pressure method for congestion detection and control mechanism which has successfully reduced the packet loss rate (buffer overflow).
The percentage of delivery ratio of data packets in the network is expressed as follows:

$$PDR = \frac{\sum Recieved\ Packets}{\sum Send\ Packets} \quad (23)$$

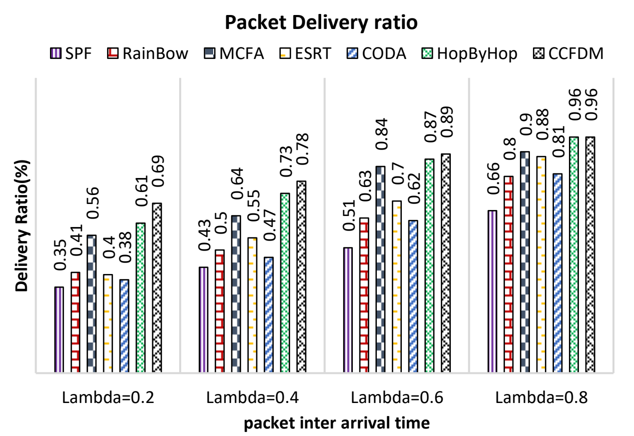

FIGURE 10. Diagram of delivery rate for the routing packets.

C. Average end-to-end network delay test

Figure 11 indicates that the proposed method is more aware of the link delay and routes as it uses a delay parameter in its fuzzy decision system and the queue delay rate in the congestion detection and control phase. Delay-aware route management and balance facilitate the network traffic, which

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

both increases the delivery ratio and reduces the average end-to-end network delay from the source to the sink. Accordingly, generating stable links, maintaining quality links, and removing low-quality links from the route can eliminate link delay as much as possible. Since network congestion can cause interruptions, longer processing and queuing time, and reduced transmission rate in network nodes, the earlier it is detected and managed, the lower the end-to-end network delay rate of packets.

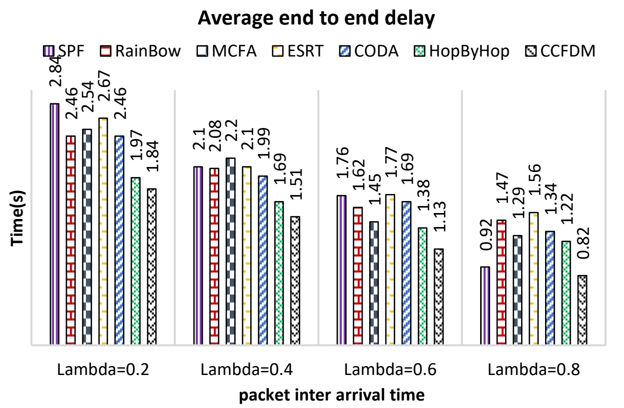

FIGURE 11. Average end-to-end network delay from the source node to the sink

Figure 11 shows a direct relationship between packet inter-arrival time and end-to-end delay rate, meaning that the higher the packet inter-arrival time, the higher the end-to-end delay rate. The proposed protocol outperformed the other methods in this paper and offered higher efficiency. The following relation gives the average end-to-end network delay:

$$Avg\ End\ to\ End\ Delay = \frac{\sum (\ Arrive\ Time - Send\ Time\ )}{\sum Numberof\ Arrived\ Packets} \quad (24)$$

D. Queue efficiency test

When transmitting data packets in a network, data packet loss can occur due to factors such as noise, signal loss, low signal-to-noise (SNR) ratio, communication channel traffic, and above all, queuing issues and inefficient queue and buffer management in the network nodes. As such, algorithms more aware of a node, link, and route QoS can further reduce the packet loss ratio. As discussed in Section III, where the congestion control, collision probability calculations, queue management and state in the proposed method were explained, although the ETX metric can increase transmission efficiency and, in a sense, link stability, without the queue state metric it is not possible to gain the expected improvements.
Figure 12 shows the test results. As predicted, by reducing the packet inter-arrival of the network ($\lambda = 0.2 \sim 0.8$) in 200 nodes, the packet loss ratio in the node queues increases. According to the variance of 20 reruns of the test, the CCFDM method managed not to remove almost any packet for the packet inter-arrival intervals of $\lambda = 0.6$ and $\lambda = 0.6$ due to buffer overflow. However, by increasing the traffic rate, i.e. reducing packet transmission interval to $\lambda = 0.4$, nearly 3% of the total packets are lost in the node queue buffer.

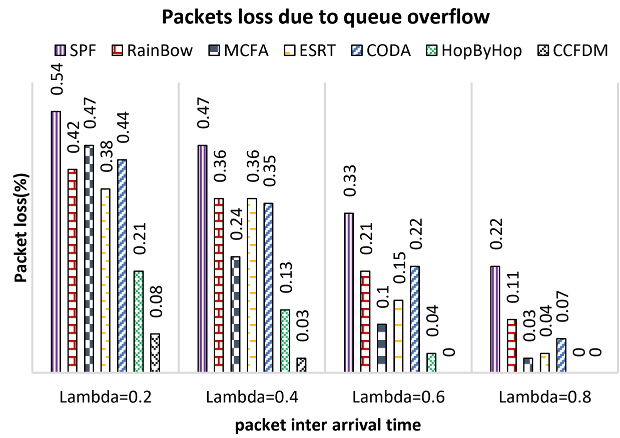

FIGURE 12. The packet loss ratio in the node queue

This value reaches a maximum of 8% for the packet inter-arrival interval of $\lambda = 0.2$, which is still considerably different from other compared methods. The HopbyHop method was only able to offer similar performance in the low-traffic test compared to the proposed method but failed to catch up as the traffic rate increased.

E. Jain's fairness index (JFI) test

To calculate JFI for network throughput rate, the throughput rate was first obtained using Eq. (25) and then the JFI definition in [2], [15] was used for the calculations. In defining the throughput of 6LowPAN communication networks, the average successful data delivery rate in a channel is called the operating power or throughput rate of the network. The data may be calculated using a logical link or through passing across the wireless medium of the sensor network. Operating power is normally measured as bps, packet/s, or packet/time interval:

$$Throughput = \frac{Total\ data\ transported}{Time\ interval} \quad (25)$$

System power or total operating power is the total rate of data delivered to the network base station in a certain time period. System power can be analyzed mathematically using the queue theory where the packet load in a time unit is denoted as the input rate $\boldsymbol{\lambda}$, and the packet power in a time unit is denoted as the output rate $\boldsymbol{\mu}$. In JFI, the objective is to find to what extent the network fairly utilizes its bandwidth resources (wireless medium) in time unit when subject to variable traffic. These calculations indicate the extent of utilization of the network capacity (available bandwidth). Evidently, a higher traffic rate or input rate $\boldsymbol{\lambda}$ indicates the lower ability of the network to utilize its nominal or actual capacity in a time unit. Equation 26 can be used to calculate the fair allocation of network resources such as bandwidth. This calculation is conducted for the sink node [53]. Parameter $\boldsymbol{th_i}$ is the throughput of node $\boldsymbol{i}$, and $\boldsymbol{n}$ is the number of network nodes in the JFI equation:

$$JFI = \frac{[\sum_{i=1}^{n} th_i]^2}{n \sum_{i=1}^{n} (th_i)^2} \quad (26)$$

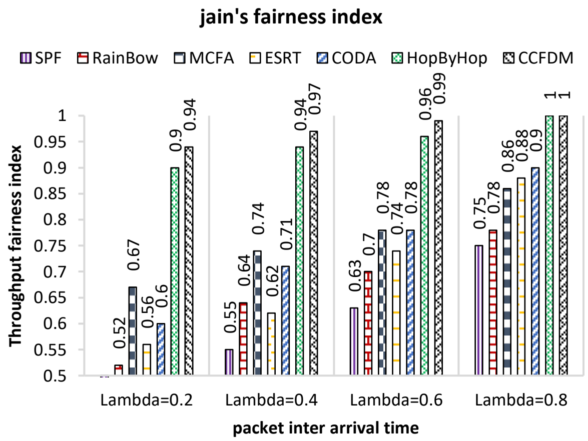

FIGURE 13. Results of JFI test for network throughput

F. Power efficiency test

The last comparison test in this paper addresses the network power efficiency, that is, the ratio of total network throughput to total energy consumption. The test attempts to show—compared to the other methods discussed to what extent the proposed method has been able to utilize the channel capacity for data exchange and transmission with respect to energy consumption. According to [54], network energy efficiency is given by:

$$PE = \frac{Throughput}{Total\ energy\ consumption} = \frac{S}{E} \quad (27)$$

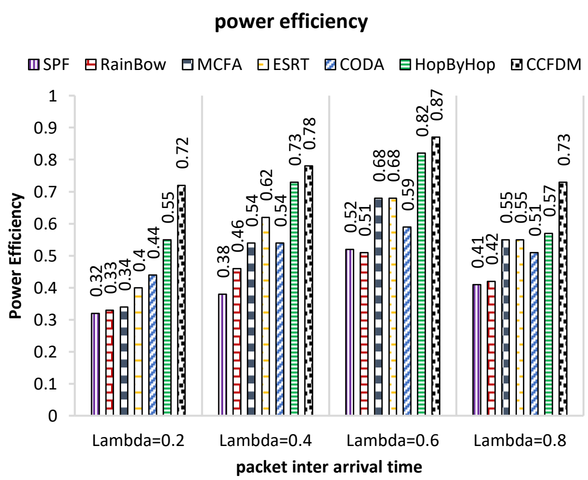

FIGURE 14. Energy efficiency test results m

Although network throughput can be optimal under certain situations, this optimality does not necessarily mean a minimum or maximum packet inter-arrival time. That is, there is a certain threshold for the input traffic and its useful throughput. According to Fig. 14, the energy efficiency of the optimum rate of CCFCM for the input traffic rate of $\lambda = 0.6$ is 100 packets per minute. At lower traffic of $\lambda = 0.8$, the network resources are not utilized to their best capacity, which is also true for $\lambda = 0.2$, where the dense network traffic reduces the network efficiency and performance.

V. Conclusion

The present paper proposed a congestion control solution for routing in classic IoT protocols based on layering network regions. This approach uses a fuzzy decision system to prevent, detect, and mitigate congesting. Since allocating equal weights for network nodes cannot lead to correct decisions due to their different geographical positions, in the proposed CCFDM method, a sectoring method was used centered on the sink node. This method groups the network nodes as sectors and offers the advantage of dynamic evaluation with effective network quantities and parameters when facing congestion and data transmission from network nodes to the sink. Thus, it assigns different significance and weights to network nodes at different regions to facilitate more accurate decisions for energy-efficient and high-throughput routing. In the second phase of the CCFDM, the parameters of remaining energy, traffic load, link ETX, and link delay rate were used as the inputs of the fuzzy decision system. The output of this system is a congestion-aware routing protocol that can prevent congestion in network nodes and reduce it. Additionally, the proposed method employed a congestion-detection method based on the queue state to detect and manage congestion. Because of the funneling effect in the network graph, the nodes closer to the sink are more prone to congestion. To solve this, the back-pressure approach was used based on the queue state. In other words, the higher the traffic rate of the node, the lower its probability of data exchange with its single-hop neighbors. This dynamic system causes the intermediary nodes to participate in routing on a per-need basis and according to the QoS criteria. The simulation results indicate that the proposed method offers better performance considering the network lifetime, packet delivery rate, end-to-end delay, queue efficiency, JFI, and energy efficiency compared to its counterpart protocols. For future works, the authors will attempt to adapt the proposed solution with the constraints of RPL-based methods to enhance its performance.

Acknowledgment

This research is sponsored by the Project: "Support of research and development activities of the J. Selye University in the field of Digital Slovakia and creative industry" of the Research & Innovation Operational Programme (ITMS code: NFP313010T504) co-funded by the European Regional Development Fund.

References

[1] C. Lim, “A Survey on Congestion Control for RPL-Based Wireless Sensor Networks,” *Sensors*, vol. 19, no. 11, p. 2567, 2019.

[2] H. A. A. Al-Kashoash, H. Kharrufa, Y. Al-Nidawi, and A.

H. Kemp, "Congestion control in wireless sensor and 6LoWPAN networks: toward the Internet of Things," *Wirel. Networks*, pp. 1–30, 2018.

[3] A. A. Khan, S. Ghani, and S. Siddiqui, "A Study on Channel Sharing for Congestion Control in WSN MAC Protocols," *Int. J. Wirel. Networks Broadband Technol.*, vol. 6, no. 1, pp. 15–33, 2017.

[4] A. U. Rajan, K. R. SV, A. Jeyasekar, and A. J. Lattanze, "Energy-efficient predictive congestion control for wireless sensor networks," *IET Wirel. Sens. Syst.*, vol. 5, no. 3, pp. 115–123, 2015.

[5] G. Daneels *et al.*, "Accurate Energy Consumption Modeling of IEEE 802.15. 4e TSCH Using Dual-Band OpenMote Hardware," *Sensors*, vol. 18, no. 2, p. 437, 2018.

[6] D. Raposo, A. Rodrigues, S. Sinche, J. Sá Silva, and F. Boavida, "Industrial IoT monitoring: Technologies and architecture proposal," *Sensors*, vol. 18, no. 10, p. 3568, 2018.

[7] Y. Bin Zikria, H. Yu, M. K. Afzal, M. H. Rehmani, and O. Hahm, "Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques." *Elsevier*, 2018.

[8] S. Liu, K. Wang, K. Liu, and W. Chen, "Noncoherent Decision Fusion over Fading Hybrid MACs in Wireless Sensor Networks," *Sensors*, vol. 19, no. 1, p. 120, 2019.

[9] T. Le and S. Moh, "Hybrid Multi-Channel MAC Protocol for WBANs with Inter-WBAN Interference Mitigation," *Sensors*, vol. 18, no. 5, p. 1373, 2018.

[10] J. Maisuria and S. Mehta, "An overview of medium access control protocols for cognitive radio sensor networks," in *Multidisciplinary Digital Publishing Institute Proceedings*, 2018, vol. 2, no. 3, p. 135.

[11] L. Oliveira, J. J. P. C. Rodrigues, S. A. Kozlov, R. A. L. Rabêlo, and V. H. C. de Albuquerque, "MAC Layer protocols for internet of things: A survey," *Futur. Internet*, vol. 11, no. 1, p. 16, 2019.

[12] K. Bhandari, A. Hosen, and G. Cho, "CoAR: Congestion-Aware Routing Protocol for Low Power and Lossy Networks for IoT Applications," *Sensors*, vol. 18, no. 11, p. 3838, 2018.

[13] M. H. Homaei, E. Salwana, and S. Shamshirband, "An Enhanced Distributed Data Aggregation Method in the Internet of Things," *Sensors*, vol. 19, no. 14. MDPI, 2019.

[14] E. K. Zavadskas, Z. Turskis, and S. Kildienė, "State of art surveys of overviews on MCDM/MADM methods," *Technol. Econ. Dev. Econ.*, vol. 20, no. 1, pp. 165–179, 2014.

[15] H. Al-Kashoash, *Congestion Control for 6LoWPAN Wireless Sensor Networks: Toward the Internet of Things*. Springer, 2017.

[16] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 52, pp. 101–115, 2015.

[17] S. A. Shah, B. Nazir, and I. A. Khan, "Congestion control algorithms in wireless sensor networks: Trends and opportunities," *J. King Saud Univ. Inf. Sci.*, vol. 29, no. 3, pp. 236–245, 2017.

[18] M. A. Jan, S. R. U. Jan, M. Alam, A. Akhunzada, and I. U. Rahman, "A Comprehensive Analysis of Congestion Control Protocols in Wireless Sensor Networks," *Mob. Networks Appl.*, pp. 1–13, 2018.

[19] A. M. Ahmed and R. Paulus, "Congestion detection technique for multipath routing and load balancing in WSN," *Wirel. Networks*, vol. 23, no. 3, pp. 881–888, 2017.

[20] 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, 2018, pp. 485–495.

[21] H. A. A. Al-Kashoash, M. Hafeez, and A. H. Kemp, "Congestion control for 6LoWPAN networks: A game theoretic framework," *IEEE internet things J.*, vol. 4, no. 3, pp. 760–771, 2017.

[22] H. D. Nikokheslat and A. Ghaffari, "Protocol for controlling congestion in wireless sensor networks," *Wirel. Pers. Commun.*, vol. 95, no. 3, pp. 3233–3251, 2017.

[23] M. Gholipour, A. Haghighat, and M. Meybodi, "Hop-by-hop traffic-aware routing to congestion control in wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2015, no. 1, p. 15, 2015.

[24] M. Gholipour, A. T. Haghighat, and M. R. Meybodi, "Hop-by-Hop Congestion Avoidance in wireless sensor networks based on genetic support vector machine," *Neurocomputing*, vol. 223, pp. 63–76, 2017.

[25] Y. Kabalci, "IEEE 802.15. 4 Technologies for Smart Grids," in *Smart Grids and Their Communication Systems*, Springer, 2019, pp. 531–550.

[26] F. Masud, A. H. Abdullah, A. Altameem, G. Abdul-Salaam, and F. Muchtar, "Traffic Class Prioritization-Based Slotted-CSMA/CA for IEEE 802.15. 4 MAC in Intra-WBANs," *Sensors*, vol. 19, no. 3, p. 466, 2019.

[27] C. M. G. Algora, V. A. Reguera, N. Deligiannis, and K. Steenhaut, "Review and Classification of Multichannel MAC Protocols for Low-Power and Lossy Networks," *IEEE Access*, vol. 5, pp. 19536–19561, 2017.

[28] M. Kumaraswamy, K. Shaila, V. Tejaswi, K. R. Venugopal, I. SS, and P. L M, "Multihop Multi-Channel Distributed QOS Scheduling MAC Scheme for Wireless Sensor Networks," *IOSR J. Comput. Eng.*, vol. 17, no. 2, pp. 01–10, 2015.

[29] W. Rehan, S. Fischer, and M. Rehan, "Anatomizing the robustness of multichannel MAC protocols for WSNs: An evaluation under MAC oriented design issues impacting QoS," *J. Netw. Comput. Appl.*, 2018.

[30] C.-M. Chao and H.-H. Wang, "The bottleneck problem in large scale IEEE 802.15. 4/ZigBee networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 26, no. 2, pp. 115–128, 2017.

[31] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, pp. 1193–1201.

[32] S. Lin *et al.*, "Toward stable network performance in wireless sensor networks: A multilevel perspective," *ACM Trans. Sens. Networks*, vol. 11, no. 3, p. 42, 2015.

[33] A. Kumar, M. Zhao, K.-J. Wong, Y. L. Guan, and P. H. J. Chong, "A Comprehensive Study of IoT and WSN MAC Protocols: Research Issues, Challenges and Opportunities," *IEEE Access*, vol. 6, pp. 76228–76262, 2018.

[34] H. Cinar, M. Cibuk, and I. Erturk, "HMCAWSN: A hybrid multi-channel allocation method for erratic delay constraint WSN applications," *Comput. Stand. Interfaces*, vol. 65, pp. 92–102, 2019.

[35] M. C. Raja, "The Minimum Cost Forwarding Using MAC

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Protocol for Wireless Sensor Networks," *Int. J. Mod. Eng. Res.*, vol. 2, no. 4, pp. 4122–4127, 2012.

[36] J. Kaur, R. Grewal, and K. S. Saini, "A survey on recent congestion control schemes in wireless sensor network," in *Advance computing conference (IACC), 2015 IEEE international*, 2015, pp. 387–392.

[37] O. Chughtai, N. Badruddin, A. Awang, and M. Rehan, "Congestion-aware and traffic load balancing scheme for routing in WSNs," *Telecommun. Syst.*, vol. 63, no. 4, pp. 481–504, 2016.

[38] Ö. B. Akan and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1003–1016, 2005.

[39] M. A. Mahmood, I. Welch, and P. Andreae, "Enhanced Event Reliability in Wireless Sensor Networks," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, 2018, pp. 93–100.

[40] K. Singh, K. Singh, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Comput. Networks*, vol. 138, pp. 90–107, 2018.

[41] D. D. Tan, N. Q. Dinh, and D.-S. Kim, "GRATA: gradient-based traffic-aware routing for wireless sensor networks," *IET Wirel. Sens. Syst.*, vol. 3, no. 2, pp. 104–111, 2013.

[42] L. Hai, Q. Gao, J. Wang, H. Zhuang, and P. Wang, "Delay-Optimal Back-Pressure Routing Algorithm for Multihop Wireless Networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2617–2630, 2018.

[43] 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*, 2003, pp. 266–279.

[44] H. Tall *et al.*, "Load balancing routing with queue overflow prediction for WSNs," *Wirel. Networks*, vol. 25, no. 229, pp. 229–239, Jul. 2019.

[45] H. Lamaazi and N. Benamar, "A Comprehensive Survey on Enhancements and Limitations of the RPL protocol: A focus on the Objective Function," *Ad Hoc Networks*, p. 102001, 2019.

[46] H.-Y. Zhou, D.-Y. Luo, Y. Gao, and D.-C. Zuo, "Modeling of node energy consumption for wireless sensor networks," *Wirel. Sens. Netw.*, vol. 3, no. 1, p. 18, 2011.

[47] P. Veeraraghavan, G. Khomami, and F. Fontan, "The Relation between the Probability of Collision-Free Broadcast Transmission in a Wireless Network and the Stirling Number of the Second Kind," *Mathematics*, vol. 6, no. 7, p. 127, 2018.

[48] S. Bindel, S. Chaumette, and B. Hilt, "F-ETX: an enhancement of ETX metric for wireless mobile networks," in *International Workshop on Communication Technologies for Vehicles*, 2015, pp. 35–46.

[49] A. Bildea, "Link quality in wireless sensor networks," de Grenoble, 2013.

[50] W. Liu, D. Zhao, and G. Zhu, "End-to-end delay and packet drop rate performance for a wireless sensor network with a cluster-tree topology," *Wirel. Commun. Mob. Comput.*, vol. 14, no. 7, pp. 729–744, 2014.

[51] P. Pinto, A. Pinto, and M. Ricardo, "End-to-end delay estimation using RPL metrics in WSN," in *Wireless Days (WD), 2013 IFIP*, 2013, pp. 1–6.

[52] V. Moghiss, M. R. Meybodi, and M. Esnaashari, "An intelligent protocol to channel assignment in wireless sensor networks: Learning automata approach," in *2010 International Conference on Information, Networking and Automation (ICINA)*, 2010, vol. 1, pp. V1-338.

[53] D. Aguirre-Guerrero, R. Marcelín-Jiménez, E. Rodriguez-Colina, and M. Pascoe-Chalke, "Congestion control for a fair packet delivery in WSN: from a complex system perspective," *Sci. World J.*, vol. 2014, 2014.

[54] M. Ram, S. Kumar, V. Kumar, A. Sikandar, and R. Kharel, "Enabling Green Wireless Sensor Networks: Energy Efficient T-MAC Using Markov Chain Based Optimization," *Electronics* , vol. 8, no. 5. 2019.

Mohammad Hossein Homaei (M19). received his B.Sc. and M.Sc. degrees in Information Technology (Networking) at UAST and IAU of Hamedan and Malayer (IRAN) respectively in 2014 and 2017. Currently, he is currently a visiting researcher at the Institute of Automation, Kalman Kando Faculty of Electrical Engineering, Obuda University, Budapest, Hungary, and also, at the Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia. He has authored 10 academic papers in referred journals and conferences and 3 patents. His research interests include Wireless Communications, Wireless Sensor Networks, Internet of Things, Algorithms, Machine Learning, Big Data, and Routing. He has been working on various projects in Networking, Distributed Systems Applications. He has recently focused on Algorithms, Protocols and Developing Sensor Networks, Autonomous Underwater Vehicles, and the Internet of Things (IoT).

Faezeh Soleimani. is a Ph.D. student in Applied Mathematics at the University of Texas at Arlington (UTA). She received a B.S. in Applied Mathematics from Razi University, Kermanshah, Iran, and M.S. in Applied Mathematics (Operations Research) from Khwarizmi University, Tehran, Iran. She previously conducted research on the Super-Efficient Decision Making Units in the field of Data Envelopment Analysis for her master thesis. She is also interested in Optimization, Computational Mathematics, and Machine Learning..

Shahaboddin Shamshirband. Shahab is an Adjunct Professor at TON DUC THANG University, Vietnam, (2) Adjunct Faculty at Iran Science and Technology University, IRAN, (3) Academic faculty at IAUC, IRAN, (4) Faculty member at University of Malaya, Malaysia, and (5) PostDoc research fellow. He received his Ph.D. in computer science from the University of Malaya (Kuala Lumpur, Malaysia) and MSc in Artificial Intelligence. He has published more than 200 papers, in refereed international SCI-IF journals (100), international conference proceedings (25), books (10) with more than 3000 citations in Google Scholar (with h-index of 29), and ResearchGate RG Score of 47. He has worked on various funded projects, with grants worth more than 50000 USD. He is on the editorial board of journals and has served as Guest Editor for journals. His major academic interests are in Computational Intelligence, Data mining in multidisciplinary fields. His articles are ranked in the highly cited papers and most downloaded papers from the top 10 % (2013 till now) in Computer Science according to the WoS.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2968524, IEEE Access

Amir Mosavi. received his MSc and Ph.D. degrees in applied informatics from London Kingston University, U.K. He was a Senior Research Fellow with the School of the Built Environments, Oxford Brookes University, and also with Norwegian University of Science and Technology. Dr. Mosavi collaborated with the Institute of Automation, Kalman Kando Faculty of Electrical Engineering, Obuda University, Budapest, Hungary, and also, and at the Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia as a data scientist. He is an Alexander von Humboldt Research Fellow Alumni for big data, the IoT, and machine learning at Bauhaus University, Germany. He is a data scientist for climate change, sustainability, and hazard prediction with more than 60 peer-reviewed articles on machine learning prediction models. He was a recipient of the Green-Talent Award, the UNESCO Young Scientist Award, the ERCIM Alain Bensoussan Fellowship Award, the Bauhaus Postdoc PROFIL, the Campus France Fellowship Award, the Slovak National Research Award, the Campus Hungary Fellowship Award, and the Endeavour-Australia Leadership Award..

Narjes Nabipour. is a senior researcher at Duytan University, Vietnam. She is a complex systems data scientist with a focus on climate data and IoT systems. She has taught advanced statistics and data science for more than ten years. Her main research interests are prediction models, causal discovery and causal inference based on graphical models and deep learning. She has so far coauthored in various journals. Her graduate studies have been in computer science with expertise in machine learning modeling.

Annamária R.Várkonyi-Kóczy (M'94–SM'97–F'07). was born in Budapest, Hungary, in 1957. She received the M.Sc. degree in electrical engineering, the M.Sc. Degree in mechanical engineer-teacher, and the Ph.D. degree from the Technical University of Budapest, Budapest, in 1981, 1983, and 1996, respectively, and the D.Sc. degree from the Hungarian Academy of Sciences, Budapest, in 2010. She was a Researcher with the Research Institute for Telecommunication, Budapest, for six years, followed by four years with the Group of Engineering Mechanics, Hungarian Academy of Sciences. From 1991 to 2009, she was with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest. Since 2009, she has been a Full Professor with the Institute of Mechatronics and Vehicle Engineering, Óbuda University, Budapest. She is also a founding professor and project leader with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Budapest. Her research interests include digital image and signal processing, uncertainty handling, soft computing, anytime, and hybrid techniques in complex measurement, diagnostics, and control systems. Dr. Várkonyi-Kóczy was the past Vice-President of the Hungarian Fuzzy Association, is an elected member of the Hungarian Academy of Engineers, a member of the John von Neumann Computer Society and the Measurement and Automation Society (Hungary).