Congestion-Aware Routing and Fuzzy-based Rate Controller for Wireless Sensor Networks

Majid HATAMIAN [1], *Maryam ALMASI BARDMILY* [2], *Mojtaba ASADBOLAND* [1],
Mehdi HATAMIAN [3], *Hamid BARATI* [1]

[1] Dept. of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
[2] Dept. of Computer Engineering, Ashtiyan Branch, Islamic Azad University, Ashtiyan, Iran
[3] Dept. of Information Technology Engineering, Payame Nur University, Tehran, Iran

majid.hatamian.h@ieee.org, {almasi.maryam, mojtaba.asadboland, mehdi.hatamian}@hotmail.com, hbarati@iaud.ac.ir

Manuscript received July 21, 2015

Abstract. *In this paper, congestion-aware routing and fuzzy-based rate controller for wireless sensor networks (WSNs) is proposed. The proposed method tries to make a distinction between locally generated data and transit data by using a priority-based mechanism which provides a novel queueing model. Furthermore, a novel congestion-aware routing using greedy approach is proposed. The proposed congestion-aware routing tries to find more affordable routes. Moreover, a fuzzy rate controller is utilized for rate controlling which uses two criteria as its inputs, including congestion score and buffer occupancy. These two parameters are based on total packet input rate, packet forwarding rate at MAC layer, number of packets in the queue buffer, and total buffer size at each node. As soon as the congestion is detected, the notification signal is sent to the offspring nodes. As a result, they are able to adjust their data transmission rate. Simulation results clearly show that the implementation of the proposed method using a greedy approach and fuzzy logic has done significant reduction in terms of packet loss rate, end-to-end delay and average energy consumption.*

Keywords

Wireless sensor networks, congestion control, fuzzy logic, greedy routing, buffer occupancy

1. Introduction

Wireless sensor networks (WSNs) consist of a large number of nodes which are distributed geographically. Nodes gather information, sense environmental changes and send the collected data to a remote destination for data analysis and decision making. Nowadays, WSNs play an increasingly important role in our life, since they can be used in a variety of monitoring application scenarios. WSNs can be deployed in forests to detect and help control the bush fires. They will also play a crucial role in healthcare where they can be used to monitor and assist elderly people and people with chronic diseases, especially those that live in more remote communities without any nearby medical facilities. In WSNs, some nodes (which are called sink nodes) are responsible for collecting data from other sensor nodes. For this reason, data collection is a crucial issue in WSNs [1, 2]. In WSNs, routing is a very complex problem for a number of reasons. Nodes in WSNs have very limited resources (e.g. battery power), which necessitates complicated schemes to maximize the use of these limited resources. Moreover, Quality-of-Service (QoS) is important in disaster detection and healthcare applications, as high latency or high packet loss may directly affect the health of people (patients) or ultimately even cost lives. Because of the nature of the many-to-one traffic in these networks, congestion may affect a WSN in some cases. This situation leads to the drop of packets with important information, as well as the waste of the energy. To this end, in order to avoid these problems, a congestion control protocol must be used [3, 4]. Congestion increases packet drops, delay, unreliability, wastage of communication resources, and decreases lifetime of the network. Congestion control mechanism includes a set of methods for monitoring and regulating the total amount of packets entering the network to keep traffic levels at a desirable value. For this purpose, decision making methods are suitable for making intelligent decisions with respect to the existent congestion. In addition, an optimized congestion control protocol plays a crucial role for properly receiving packets [5, 6, 7, 8, 9].

Our Work: In this paper, congestion-aware routing and fuzzy-based rate controller for WSNs is proposed. In order to distinguish between the whole traffic of the network, we divide it into two kinds, including locally generated data and transit data. In WSNs, the transit data has to travel many hops to reach the sink. In this particular case, dropping of such data leads to more energy dissipation. As a result, it is an essential need to give priority to these data during routing procedure. For this purpose, we define a novel queueing

DOI: 10.13164/re.2016.0114

model. Additionally, we propose a parameter which is called congestion score. By this parameter, we will be able to estimate the congestion level at every offspring node. Moreover, the proposed congestion-aware routing uses congestion score and delay as two parameters to perform the routing procedure by considering less congested paths. Finally, by applying congestion score and buffer occupancy to the proposed fuzzy rate controller, it is possible to control the data transmission rate of offspring nodes. Our main contributions can be summarized as follows:

- In the proposed method, we use a parameter by considering the total packet input rate and the packet forwarding rate by combination of transit data and generated data queues at a specific node. This is in contrast to the traditional congestion control mechanisms.
- Using a greedy approach as a congestion-aware method for routing among nominated nodes leads to selection of the most appropriate routes in terms of congestion level which is in contrast to the previous work.
- Using fuzzy logic as a decision making method for data transmission rate controlling by applying two inputs which has not been used in any previous work as a method for congestion controlling.

The rest of this paper is organized as follows. Section 2 investigates some previous work for congestion controlling. Section 3 explains system model which is used in the proposed method. Section 4 presents the proposed congestion-aware routing using a greedy approach. Section 5 explains the steps of the rate adjustment using fuzzy rate controller. In Section 6 the proposed method has been evaluated in terms of the following criteria: packet loss rate, end-to-end delay and energy consumption. Finally, Section 7 concludes this paper.

2. Related Work

Heretofore, several studies have been proposed on developing efficient congestion control protocols for WSNs.

Congestion Detection and Avoidance in Sensor Networks (CODA) [10] tries to detect the potential congestion based on a combination of the present and past channel loading conditions, and the current buffer occupancy. Once the congestion is detected, nodes inform their upstream neighbors via a backpressure message. The destination node regulates its data transmission rate or drops packets based on the local congestion situations once it receives the backpressure message. CODA is an energy-efficient method, but successful delivery of the packets to the destination is not guaranteed. Sensor Transmission Control Protocol (STCP) [11] is a generic transport protocol which uses information of the transport layer for congestion controlling. Majority of STCP functionalities are implemented at the base station (BS). Each node might be the source of multiple data flows with different characteristics such as flow type, transmission rate and required reliability. It should be noted that STCP is able to support multiple applications, controlled variable reliability and congestion detection and avoidance mechanisms. Although this method is capable of reducing energy consumption, but it leads to high packet loss during congestion in the network. Event to Sink Reliable Transport (ESRT) [12] protocol tries to achieve reliable event detection regarding minimum energy consumption. ESRT operates based on event reliability and reporting frequency. It can be argued that reliability can be defined as the number of data packets received at the decision interval at the sink. The proposed congestion detection mechanism in ESRT is triggered with regard to the local buffer level of the sensors nodes. If the event-to-sink reliability is lower than required, ESRT adjusts the reporting frequency of the source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively in order to conserve the energy while still maintaining reliability. In addition, the sink node periodically configures the source data transmission rate by broadcasting the congestion state to all sensor nodes. After receiving the broadcast message, each sensor node tries to reduce the data transmission rate. Due to the reduced data traffic, the traffic congestion problem can be alleviated gradually. In [13], a hierarchical tree based congestion control approach using fuzzy logic for heterogeneous traffic (HTCCFL) is proposed. In the first phase of HTCCFL, a hierarchical tree is constructed using topology control algorithm. By using a control packet, each node is able to be aware of the congestion state of all the neighbors. In the second phase, congestion detection is done using fuzzy logic. The state of congestion is then estimated by the outcome of the fuzzy rules. If rate adjustment is not feasible, then alternate path can be selected from the established hierarchical tree. In [14], a hop-by-hop traffic-aware routing (HTR) has been proposed. The proposed method is able to adjust the data transmission rate of nodes for a multi-sink WSN. This method is designed through constructing a hybrid virtual gradient field using depth and normalized traffic loading to provide a balance between optimal paths and possible congestion on routes toward the sinks. By considering the number of packets in the queue, the congestion degree and the average cumulative queue length, a node informs its neighbors about both the distance cost from itself to a sink and possible congestion. In [15], an efficient fuzzy based congestion control algorithm (FCC) has been proposed. The proposed method uses three parameters for congestion detection procedure, including the node degree, queue length and data arrival rate. By applying these parameters to a fuzzy inference system, authors tried to estimate the level of congestion. By considering this estimation, they tried to adjust the data transmission rate in order to mitigate the potential congestion.

3. System Model

3.1 Network Model

Consider one sink node and the set of sensor nodes which are defined as follows:

$$S = [N_{S_1}, N_{S_1}, N_{S_1}, ..., N_{S_n}]. \quad (1)$$

Each sensor node is shown by N_{S_i}, where $1 \leq i \leq n$. In the model of studied network, we assumed that sensor nodes are placed randomly and independently in a given environment and are homogeneous (in terms of computation power, processing and storage space). Additionally, the network structure consists of several sink and sensor nodes that communicate with each other. When a sensor node transmits its data to the sink, then it is called a child node and its upstream node is called its parent. The number of child nodes for a parent node S is shown by $C(S)$ as follows:

$$C(S) = [C_1, C_2, C_3, ..., C_c]. \quad (2)$$

3.2 Node Model

The energy consumption of nodes is defined by the energy which is consumed for sending and receiving the messages. Like [16], energy is consumed on data sending/receiving for a sensor node is calculated as follows:

$$E_{tx}(l,d) = E_{elec} \times l + \begin{cases} \varepsilon_{mp} \times l \times d^2, d \leq d_0, \\ \varepsilon_{mp} \times l \times d^4, d > d_0, \end{cases} \quad (3)$$

where $E_{tx}(l,d)$ is the consumed energy when transmitting a l-bit message through the distance d. E_{elec} is the electronic energy consumed per bit for coding, modulation, filtering and spreading. Also the distance threshold (d_0) is calculated as follows:

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}, \quad (4)$$

where ε_{fs} represents the amplifier parameter in a free space model when the transmission distance is shorter than d_0 and ε_{mp} represents the amplifier parameter in a multi-path fading channel model when the transmission distance is longer than d_0. Moreover, the energy consumption on receiving data is calculated as follows:

$$E_{rx}(l) = E_{elec} \times l. \quad (5)$$

Finally, the total energy consumption for transmitting a l-bit message from a source node S to a destination node D through the distance d is obtained as follows:

$$E_{S,D}(l,d) = E_{tx}(l,d) + E_{rx}(l), \quad (6)$$

where, $E_{tx}(l,d)$ is the energy consumption for transmitting a l-bit message through distance d, and $E_{rx}(l)$ is the energy consumption for receiving a l-bit message. In the model of studied network there are two types of data: 1 – locally generated data (The data generated from any source node) and 2 – transit data (the data transmitted via intermediate nodes). It is evident that transit data has to travel many hops to reach the sink. In this particular case, dropping of such data leads to more energy dissipation. As a result, it is important to give priority to these data during routing procedure. In the proposed method, each node supports two types of traffic, including locally generated data and transit data. Since the data link (MAC) layer is responsible for controlling the forwarding rate of packets from the network layer, the proposed queuing model is placed between these two layers. Fig. 1 shows the proposed queuing model [17].

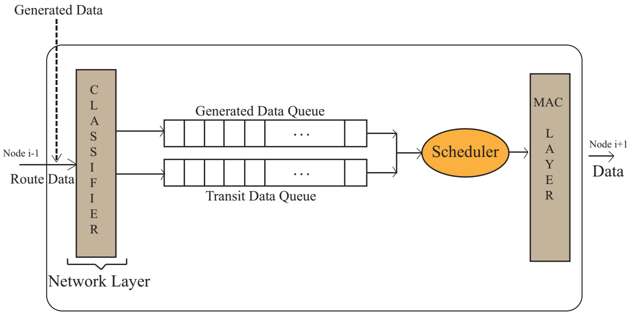

Fig. 1. Dividing the whole traffic of the network into two types by a novel queuing model.

It should be noted that generated data at nodes and transit data are queued separately in these two queues. The transit data queue has a threshold level. When the queue space is occupied above the threshold level, both queues are utilized for storing transit data and all the packets in generated data queue are dropped. This is done in order to provide sufficient space for transit traffic (since they have high priority). When the space of transit data queue falls down below the threshold level, both queues are used in normal way. We use the following equation to set a desirable priority:

$$E_{D_{N_{S_i}}} = \alpha Q_{T_{N_{S_i}}} + \beta Q_{L_{N_{S_i}}}, \quad (7)$$

where $E_{D_{N_{S_i}}}$ is the data priority estimator for node N_{s_i}, $Q_{T_{N_{S_i}}}$ and $Q_{L_{N_{S_i}}}$ indicate transit and locally generated data queues at each node N_{s_i}, respectively. Additionally, α and β are the coefficients for $Q_{T_{N_{S_i}}}$ and $Q_{L_{N_{S_i}}}$, respectively. Moreover, they are determined in a tentative way. It should be noted that:

$$\alpha + \beta = 1, \quad (8)$$

where $\alpha > \beta$. More information about the influence of α and β on the performance of the proposed queueing model will be explained in Section 6. The innovation in the queuing model is that the transit data always has a higher chance of reaching sink. In addition, since upstream traffic plays an important role in many WSNs applications, it is an essential need to provide a solution to act against the potential congestion. For this reason, the proposed method is suitable for using in upstream applications.

4. The Proposed Congestion-Aware Routing

In this section, the congestion-aware routing parameters and the procedure of the congestion-aware routing will be explained.

4.1 Routing Parameters

4.1.1 Congestion Score (CS)

Congestion Score (CS) describes the current congestion level at each node N_{s_i}, and is calculated for all the offspring nodes. $CS_{N_{s_i}}$ is defined as follows:

$$CS_{N_{s_i}} = \left(1 - \frac{r_{in}^{N_{s_i}}}{r_{for}^{N_{s_i}}}\right) T, \tag{9}$$

where $r_{in}^{N_{s_i}}$ represents the total packet input rate to node N_{s_i}, and $r_{for}^{N_{s_i}}$ indicates the packet forwarding rate at MAC layer by node N_{s_i} during time period T. Furthermore, $r_{in}^{N_{s_i}}$ can be defined as follows:

$$r_{in}^{N_{s_i}} = R_s^{N_{s_i}} + R_{tr}^{N_{s_i}}, \tag{10}$$

where $R_s^{N_{s_i}}$ represents the source traffic rate (generated by node N_{s_i}), and $R_{tr}^{N_{s_i}}$ represents the transit traffic rate of node N_{s_i} from its child nodes. We can define the occurrence or the non-occurrence of congestion as follows:

$$CS = \begin{cases} \text{Congestion does not happen} & \text{if } 0 \le CS \le 1, \\ \text{Congestion happens} & \text{if } CS < 0. \end{cases} \tag{11}$$

If $CS < 0$, then $r_{for}^{N_{s_i}} < r_{in}^{N_{s_i}}$ and congestion happens. Moreover, if $0 \le CS \le 1$, then $r_{for}^{N_{s_i}} \ge r_{in}^{N_{s_i}}$ and congestion does not happen. As soon as the congestion is detected, the notification signal is sent to the offspring nodes. There are two types of congestion notification signal: 1 – Implicit Congestion Notification (ICN) and 2 – Explicit Congestion Notification (ECN). In ICN, the notification signal is sent along with the data packet. But in ECN, the notification signal is sent as a separate packet. In the proposed method, ICN is used to send the notification signal to all the offspring nodes. When offspring nodes received the notification signal, adjust their transmission rate (the details will be explained in Section 5). ICN is more effective than ECN. Since in ECN the notification signal is sent as a separate packet and this imposes a heavy traffic overhead in the network.

4.1.2 Delay

In order to calculate the delay among neighboring nodes, each node sends a control packet which is called *Advertisement Delay*(ADVD) packet to all the neighboring nodes. Neighboring nodes will respond to ADVD packet by sending a reply packet which is called *Advertisement Reply* (ADVD-REP) packet. Delay is considered as one of the criteria of congestion-aware routing because it leads to the selection of the paths which have less delay. Delay between two nodes N_{s_i} and N_{s_j} is represented by $D_{N_{s_i},N_{s_j}}$ and is obtained as follows:

$$D_{N_{s_i},N_{s_j}} = t_{s_{N_{s_i}}} + t_{s_{N_{s_j}}}, \tag{12}$$

where $t_{s_{N_{s_i}}}$ denotes the time it takes to ADVD packet reach its destination (from node N_{s_i} to node N_{s_j}), and $t_{s_{N_{s_j}}}$ denotes the time it takes to ADVD-REP packet reach its destination (from node N_{s_j} to node N_{s_i}).

4.1.3 Buffer Occupancy

Consider two neighboring nodes. For example, node a wants to send a packet to another node b ($b \in neighbors(a)$). This is done only when node b has enough storage space to store the packet from a. With regard to the buffer occupancy criterion in the routing procedure, packets will not be dropped at the receiver because of the buffer overflow. We define function $BO_{N_{s_i}}$ as the function to denote the amount of buffer occupancy at node N_{s_i} as follows:

$$BO_{N_{s_i}} = \frac{N_Q^{N_{s_i}}}{N_T^{N_{s_i}}}, \tag{13}$$

where, $N_Q^{N_{s_i}}$ represents the number of packets in the queue buffer, and $N_T^{N_{s_i}}$ represents the total buffer size at node N_{s_i}. It is clear that the value of $BO_{N_{s_i}}$ is always in the range of $[0, 1]$. At the worst case scenario, $BO_{N_{s_i}}$ is 1. Hence, it means $N_Q^{N_{s_i}} = N_T^{N_{s_i}}$.

4.2 Routing Procedure

In the proposed method, each node N_{s_i} is aware of the following properties of its neighboring nodes (i.e. node N_{s_j}):

$$[CS_{N_{s_j}}, D_{N_{s_i},N_{s_j}}, BO_{N_{s_j}}]. \tag{14}$$

This information will be exchanged periodically. Source node checks the value of the buffer occupancy ($BO_{N_{s_j}}$) to select a relay node for transmitting the packet. If $BO_{N_{s_j}} \neq 1$, it will be gone to the next phase of the routing procedure. Otherwise, node N_{s_j} will be disregarded. As a result, the nodes whose BO values are not equal to 1 (between [0,1] and less than 1), are always selected for the next phase. Source node N_{s_0} is assumed. Set S includes all the neighbors of node N_{s_0}. It consists of ordered pairs which denote the congestion score and the delay between node N_{s_0} and its neighbors, respectively (neighbors of node N_{s_0} are shown as $N_{s_1}, N_{s_2}, N_{s_3}, \ldots$). In this case, we have:

$$S = \left\{(D_{N_{s_0},N_{s_1}}, CS_{N_{s_1}}), (D_{N_{s_0},N_{s_2}}, CS_{N_{s_2}}), \ldots\right\}, \tag{15}$$

where $(D_{N_{s_0},N_{s_1}}, CS_{N_{s_1}})$ means that the delay between node N_{s_1} and source node is $D_{N_{s_0},N_{s_1}}$ and the value of its congestion score is $CS_{N_{s_1}}$. In order to select or reject node N_{s_n} as the next-hop, we have two options:

Definition 1: $N_{s_n} = 0$; in this state, the node will not be selected. In fact, the delay between the source node and the sink node is less than the delay between the source node and the node N_{s_n}. As a result, there is no need to add the value of the congestion score of the mentioned node to the set.

Definition 2: $N_{s_n} = 1$; in this state, the delay between the source node and the node N_{s_n} should be subtracted from the sink node (if the delay between node N_{s_0} (source node) and node N_{s_n} is denoted by $D_{N_{s_0},N_{s_1}}$ and the delay between node N_{s_0} and the sink node is denoted by $D_{N_{s_0},sink}$, then we have $D_{N_{s_0},sink} - D_{N_{s_0},N_{s_1}}$. When $N_{s_n} = 1$, this node can be a suitable option for transferring the information. Obviously, since the information is transmitted to this node (from source node), we should subtract the value of the delay between N_{s_0} and N_{s_n} from the value of the delay between N_{s_0} and the sink node. We have to do this because our final objective is to transmit the data packets to the sink node and we have already passed a portion of the path (the path between N_{s_0} and N_{s_n}). Consequently, the value of the congestion score of node N_{s_n} will be added to the congestion score set (it means that $CS_{N_{s_n}}$ is added to the set).

Suppose the delay and the congestion score of each neighboring node as an ordered pair. If we have k ordered pairs, we can use them to define $S^i = (X_i, Y_j)|1 \leq j \leq k$. It should be noted that $S^0 = (0,0)$. Each pair of S^i is defined as $(D_{N_{s_i},N_{s_j}}, CS_{N_{s_j}})$. Therefore, we can write:

$$S_1^i = \{(D + D_{N_{s_i},N_{s_j}}, CS + CS_{N_{s_j}})|(D, CS) \in S^{i-1}\}. \quad (16)$$

Furthermore, the members of S^{i+1} can be obtained by merging ordered pairs of S^i and S_1^{i+1}. It should be mentioned that if we have $(D_{N_{s_i},N_{s_j}}, CS_{N_{s_j}})$ and $(D_{N_{s_i},N_{s_k}}, CS_{N_{s_k}})$ and also $D_{N_{s_i},N_{s_j}} \geq D_{N_{s_i},N_{s_k}}$ and $CS_{N_{s_j}} \geq CS_{N_{s_k}}$, we should omit $(D_{N_{s_i},N_{s_j}}, CS_{N_{s_j}})$ according to (16). In other words, if the delay between neighboring node A and the source node is higher than the delay between neighboring node B and the source node and meanwhile its congestion score is more than node B, node A should not be considered in the final set. For example, consider Fig. 2. In this figure, the values of the delay between each two nodes and the congestion score for each node are given (the values are normalized between [0,1]).

Node N_{s_0} wants to select one of its neighbors ($N_{s_1} - N_{s_4}$) as the next-hop. According to the values of delay and congestion score of each node, we apply set approach to nodes N_{s_1}, N_{s_2}, N_{s_3} and N_{s_4} as follows:

$$\begin{aligned} N_{s_1} &= (D_{N_{s_0},N_{s_1}}, CS_{N_{s_1}}) = (0.5, 0.2) \\ N_{s_2} &= (D_{N_{s_0},N_{s_2}}, CS_{N_{s_2}}) = (0.4, 0.5) \\ N_{s_3} &= (D_{N_{s_0},N_{s_3}}, CS_{N_{s_3}}) = (0.2, 0.1) \\ N_{s_4} &= (D_{N_{s_0},N_{s_4}}, CS_{N_{s_4}}) = (0.7, 0.8). \end{aligned} \quad (17)$$

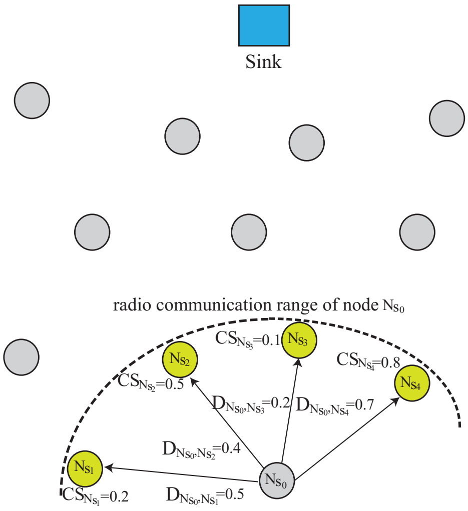
Sink
radio communication range of node N_{S_0}
$CS_{N_{S_3}}$=0.1
N_{S_3}
$CS_{N_{S_4}}$=0.8
N_{S_2}
N_{S_4}
$CS_{N_{S_2}}$=0.5
$D_{N_{S_0},N_{S_3}}$=0.2
$D_{N_{S_0},N_{S_4}}$=0.7
$D_{N_{S_0},N_{S_2}}$=0.4
N_{S_1}
N_{S_0}
$CS_{N_{S_1}}$=0.2
$D_{N_{S_0},N_{S_1}}$=0.5

Fig. 2. An example of the proposed routing approach.

The first component of each ordered pair represents the delay between node N_{s_0} and its neighboring nodes and the second component represents the congestion score of neighboring nodes. With respect to what has already been said, we can write:

$$\begin{aligned} S^0 &= \{(0,0)\}, \\ S_1^1 &= \{(0.5, 0.2)\}, \\ S^1 &= \{(0,0), (0.5, 0.2)\}, \\ S_1^2 &= \{(0.4, 0.5), \xcancel{(0.9, 0.7)}\}, \\ S^2 &= \{(0,0), \xcancel{(0.4, 0.5)}, \xcancel{(0.5, 0.2)}\}, \\ S_1^3 &= \{(0.2, 0.1), \xcancel{(0.6, 0.6)}, \xcancel{(0.7, 0.3)}\}, \\ S^3 &= \{(0,0), (0.2, 0.1)\}, \\ S_1^4 &= \{\xcancel{(0.7, 0.8)}, \xcancel{(0.9, 0.9)}\}, \\ S^4 &= \{(0,0), (0.2, 0.1)\}. \end{aligned} \quad (18)$$

Ordered pair $(0.9, 0.7)$ is omitted while merging S^1 and S_1^2. As previously mentioned, when $D_{N_{s_i}} \geq D_{N_{s_j}}$ and $CS_{N_{s_i}} \geq CS_{N_{s_j}}$, we must omit $(D_{N_{s_i}}, CS_{N_{s_i}})$ (also this process is done for ordered pairs $(0.4, 0.5)$, $(0.5, 0.2)$, $(0.6, 0.6)$ and $(0.7, 0.3)$ while merging S^2 and S_1^3. As well as $(0.7, 0.8)$ and $(0.9, 0.9)$ while merging S^3 and S_1^4). Now, we should see whether the last component of S^4 is a member of its previous set S^3 or not:

$$\begin{aligned} &(0.2, 0.1) \in S^3 \rightarrow N_{s_4} = 0, \\ &(0.2, 0.1) \notin S^2 \rightarrow N_{s_3} = 1, \\ &(0.2, 0.1) - (0.2, 0.1) = (0,0), \\ &(0,0) \in S^1 \rightarrow N_{s_2} = 0, \\ &(0,0) \in S^0 \rightarrow N_{s_1} = 0. \end{aligned} \quad (19)$$

As it can be seen, node N_{s_3} is selected as the next-hop (according to **Definition 2**).

5. Rate Adjustment using Fuzzy Rate Controller

The structure of the proposed fuzzy rate controller consists of four parts: fuzzifier, Fuzzy Rate Adjustment Inference System (FRAIS), rule base, and defuzzifier. This structure has been shown by Fig. 3. Fuzzifier maps crisp inputs to corresponding fuzzy sets by assigning a degree of membership for each fuzzy set. In a crisp set, an element is either a member of the set or not. Also a membership function is the relationship between the values of an element and its degree of membership in a set. Rule base is an important part of every fuzzy controller. In this paper, we show the rule base in Tab. 1 by considering the relations between congestion score and buffer occupancy. Additionally, defuzzifier is responsible for mapping fuzzy values to the corresponding crisp values. As highlighted in the previous sections, those nodes that their CS values are less than 0, are congestion prone nodes. In this particular case, the offspring nodes will be informed using a notification signal. Once the offspring nodes receive the notification signal, they adjust their data transmission rate using proposed fuzzy rate adjustment inference system.

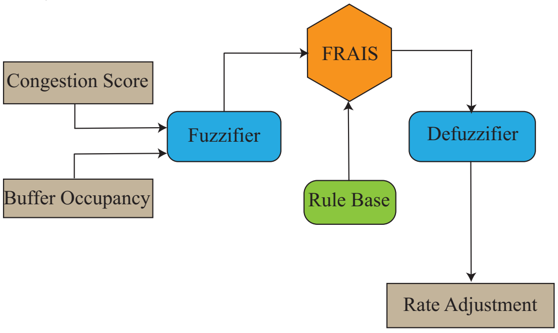

Fig. 3. The structure of the proposed fuzzy rate controller.

5.1 Fuzzification Process

Assume that A_1, A_2, and A_3 are crisp sets. Then $\mu_{B_1} : A_1 \to [0,1]$, $\mu_{B_2} : A_2 \to [0,1]$ and $\mu_{B_3} : A_3 \to [0,1]$ are called the membership functions of B_1, B_2 and B_3, which define the fuzzy sets B_1, B_2 and B_3 of A_1, A_2 and A_3. In order to perform the fuzzification process, we should map two crisp sets in fuzzy sets as follows:

$$B_1 = \text{Congestion Score} = C_i \in \{\text{low, medium, high}\}, \quad (20)$$

$$B_2 = \text{Buffer Occupancy} = B_i \in \{\text{low, medium, high}\}. \quad (21)$$

We mapped C_i and B_i in B_1 and B_2 fuzzy sets, respectively. After applying inputs to the Fuzzy Inference System (FIS), the output fuzzy set is obtained as follows:

$$B_3 = \text{Rate Adjustment} = R_i \in \{\text{DVL, DL, DM, DH, DVH}\}. \quad (22)$$

- The fuzzy variable Congestion Score has three fuzzy states, including low, medium and high, and its membership function is shown by Fig. 4(a).
- The fuzzy variable Buffer Occupancy has three fuzzy states, including low, medium and high, and its membership function is shown by Fig. 4(b).
- The output represents the rate adjustment value and has five fuzzy states, including DVL (Decrease Very Low), DL (Decrease Low), DM (Decrease Medium), DH (Decrease High), and DVH (Decrease Very High). Also, its membership function is shown by Fig. 4(c). It should be noted that all values have been normalized between [0,1].

In order to obtain the fuzzy system output (according to the inputs applied to the system), we should define the fuzzy rules. In this case, we have 9 fuzzy rules. Since we have two fuzzy sets and each of them has three states. Accordingly, the total number of possible fuzzy inference rules is 9 ($3^2 = 9$). These 9 rules are shown in Tab. 1.

No.	B_1	B_2	B_3
1	low	low	DVL
2	low	medium	DL
3	low	high	DM
4	medium	low	DL
5	medium	medium	DM
6	medium	high	DH
7	high	low	DM
8	high	medium	DH
9	high	high	DVH

Tab. 1. FRAIS Rules.

5.2 Defuzzification Process

In order to obtain the numerical result of our fuzzy system and discover the crisp output $b_{3_{\text{crisp}}}$, center of sums (COS) defuzzification method is used. In this method, fuzzy logic controller first calculates the geometric center of area (COA) for each membership function as follows: [18, 19, 20]:

$$\text{COA} = \frac{\int \mu_{b_3}(b_3)\, b_3 \mathrm{d}b_3}{\int \mu_{b_3}(b_3)\, \mathrm{d}b_3}. \quad (23)$$

Fuzzy logic controller calculates the area under the scaled membership functions and within the range of the output variable, then uses the above equation to calculate the geometric center of this area. Finally, the following equation is used to calculate a weighted average of the geometric center of area for all the membership functions [18, 19, 8]:

$$b_{3_{\text{crisp}}} = \frac{\sum_{i=1}^{5} \text{COA}_i.\text{Area}_i}{\sum_{i=1}^{5} \text{Area}_i}. \quad (24)$$

Since we have five states for the output, first we should calculate COA_1 to COA_5 and then Area_1 to Area_5. It should be mentioned that COA_1 to COA_5 are the geometric centers of areas of the scaled membership functions and Area_1 to Area_5 are the scaled membership functions.

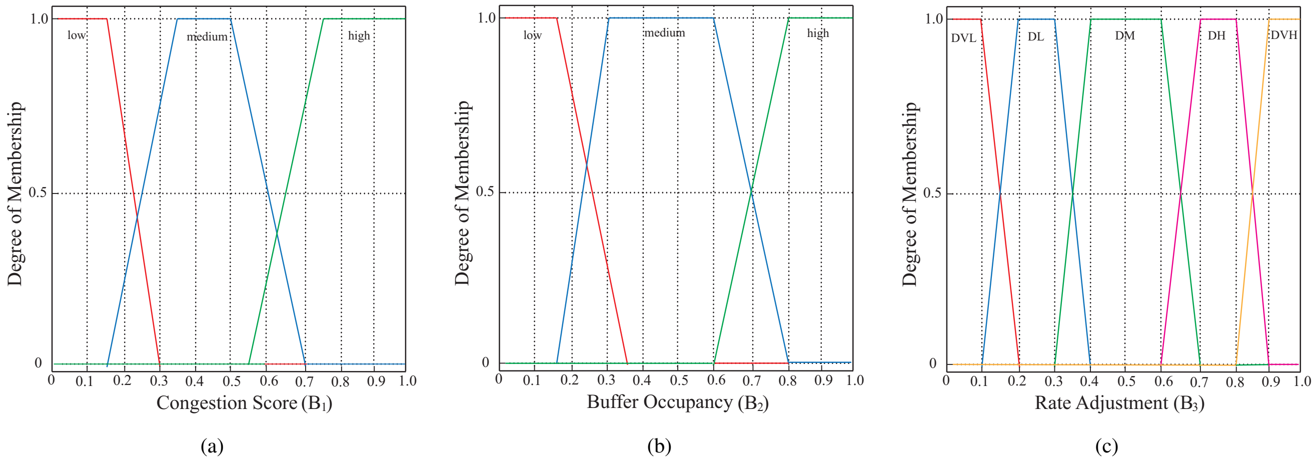

Fig. 4. Fuzzy membership functions for (a) congestion score, (b) buffer occupancy, and (c) rate adjustment.

Parameter	Value
D_0	87.7 m
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
R_C	30 m
E_{DA}	5 nJ/bit/signal
Initial energy of each node	0.5 J
Packet size	500 bytes

Tab. 2. Simulation parameters.

6. Simulation Results and Discussions

The performance of the proposed method is compared with CODA [10], HTR [14], and FCC [15] in terms of packet loss rate, end-to-end delay, and the average energy consumption. The number of nodes in the simulated network is 100 and they are randomly deployed in 100 × 100 m^2 field, including source nodes and sink nodes. The available buffer size of each node is 50 packets. We assume that all nodes have no mobility since the nodes are fixed in the most applications of WSNs. In addition, the data traffic generator tries to generate a packet every 10 ms. Additionally, Tab. 2 shows the simulation parameters.

During the tests, we changed the value of α in different steps of the simulations. In fact, we tried to realize the efficacy of this coefficient on the performance of the proposed method. For this reason, we considered two scenarios for implementation of our method. In these scenarios, we assumed that $\alpha = 0.8$ and $\alpha = 0.4$, respectively. Then, we measured the performance of the proposed method with respect to these two scenarios. Furthermore, the structure of the simulated network which consists of several sink and regular nodes has been shown by Fig. 5.

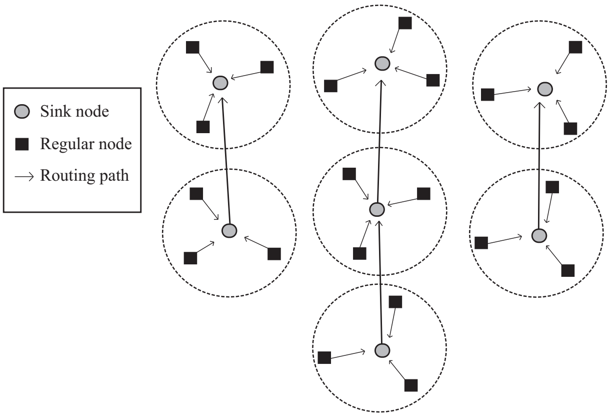

Fig. 5. The structure of the simulated networks (by considering many-to-one traffic).

6.1 First Scenario

In the first scenario, we assume that $\alpha = 0.8$. In other words, when the transit data queue is filled to four fifth of its capacity, both generated and transit data queues are utilized for storing transit data. With regard to this assumption, we intend to measure the impact of α on the performance of the proposed method.

Fig. 6 shows the performance of CODA, HTR, FCC, and the proposed protocol in terms of packet loss rate. As a matter of fact, the amount of data packets lost within a certain time period is an important index for measuring the quality of the network service. As it is evident from Fig. 6, the simulation results clearly show that the proposed protocol has a better functionality in comparison with CODA, HTR, and FCC protocols in terms of packet loss rate. Accordingly, the performance of the proposed method is much better than the previous methods in terms of packet loss rate in most of the seconds of the conducted simulations. For example, in second 60, packet loss rate of CODA is 42 packets per second, while this value is 35 and 28 for HTR and FCC, respectively. On the other hand, this value is 21 for the proposed method.

This is due to the fact that the proposed queuing model tries to prevent dropping of transit data. Since in our proposed method, we used a novel concept for assigning priority to the data packets using a priority-based approach. Additionally, it can be argued that if more nodes are near to the sink, the time of data transmission will be less.

In Fig. 7, the performance of CODA, HTR, FCC, and the proposed protocol is evaluated in terms of end-to-end delay. End-to-end delay indicates the average time taken by a data packet to arrive at the destination. It should be mentioned that only the data packets that successfully delivered to the destinations are considered in our measurement. As a matter of fact, the lower value of end-to-end delay represents the better performance of the protocol. It is obvious from Fig. 7 that the end-to-end delay of CODA, HTR, and FCC is much higher than the proposed method. For example, in second 130, the values of end-to-end delay for CODA, HTR, and FCC are 1.611, 1.232, and 1.118, respectively. On the other hand, this value is 0.825 for the proposed method. As a result, the delay dispersion of the proposed method is much lesser than CODA, HTR, and FCC. The delay dispersion of the proposed method is between 0 and 1.201. In the meanwhile, the delay dispersion of other methods is between 1 and 1.833. Accordingly, we can conclude that the performance of the proposed method is better than CODA, HTR and FCC in terms of end-to-end delay.

Fig. 8 shows the average energy consumption per bit for CODA, HTR, FCC, and the proposed method. The average energy consumption per bit is required to transmit a packet to the sink successfully. As it is evident in Fig. 8, the proposed method consumes less energy than CODA, HTR, and FCC. It can be seen that the energy consumption is 22 J for CODA in second 50, for HTR in second 58, for FCC in second 59, and for the proposed protocol in second 71. The energy consumption is 28 J for CODA in second 61, for HTR in second 72, for FCC in second 68, and for the proposed method in second 90. In fact, the simulation results obviously show that the proposed protocol has significant reduction in terms of the energy consumption. Since our method uses a greedy approach and considers the delay and the congestion score as two parameters for greedy routing, consumes less energy than other methods. Furthermore, fuzzy rate controller controls the amount of data transmission rate and handles the potential congestion. On the other hand, by using a novel queuing model, our method is able to give high priority to transit data. As a result, these kind of data have more chance to reach the sink. In other words, there is a lesser chance for them to be lost. Since dropping of such data leads to more energy dissipation. In fact, our method tries to keep transit data alive for a longer time.

6.2 Second Scenario

In second scenario, we assume that $\alpha = 0.4$. In other words, when the transit data queue is filled to two fifth of its capacity, both generated and transit data queues are utilized for storing transit data. With regard to this assumption, we intend to measure the impact of α on the performance of the proposed method.

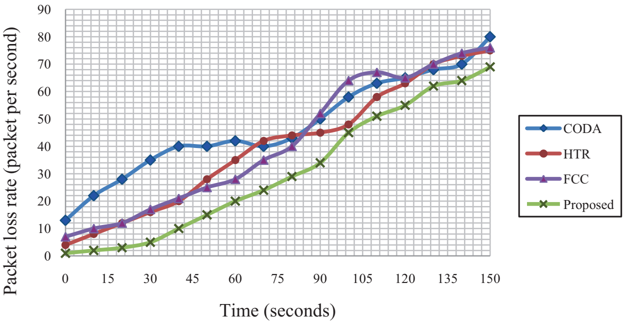

Fig. 6. Comparison in terms of packet loss rate (α = 0.8).

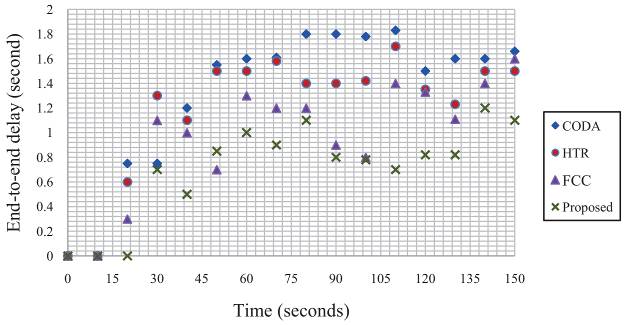

Fig. 7. Comparison in terms of end-to-end delay (α = 0.8).

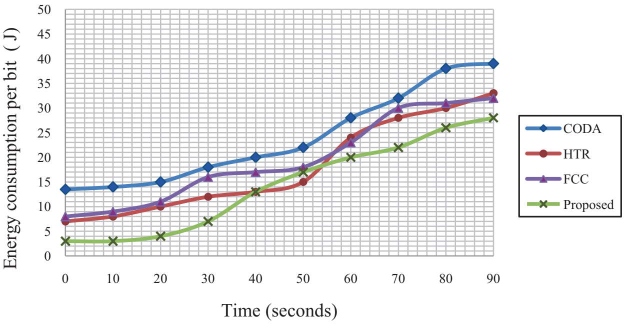

Fig. 8. Comparison in terms of the average energy consumption (α = 0.8).

Fig. 9 illustrates the performance of CODA, HTR, FCC, and the proposed protocol in terms of packet loss rate. As it is evident from Fig. 9, the simulation results clearly reveal that the proposed protocol has a better performance in comparison with CODA, HTR, and FCC protocols in terms of packet loss rate. Although, the performance of the proposed method decreased (as a result of decreasing in the value of α), but packet loss rate of the proposed method is less than previous methods in most of the seconds of the simulations. For example, in second 50, packet loss rate of CODA is 41 packets per second. While this value is 28 and 25 for HTR and FCC, respectively. On the other hand, this value is 20 for the proposed method. With respect to this fact, it is evident that our proposed queueing model is capable of reducing packet drops. In this case, this queueing model plays an increasingly important role in reducing the packet loss rate.

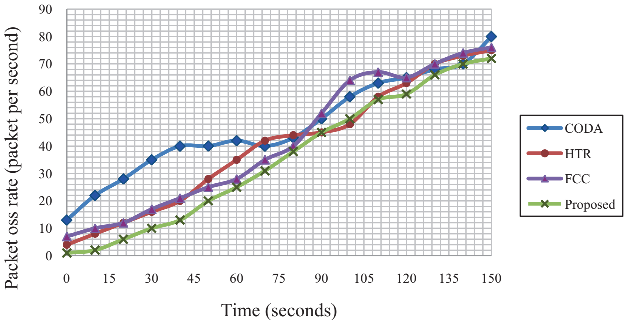

Fig. 9. Comparison in terms of packet loss rate (α = 0.4).

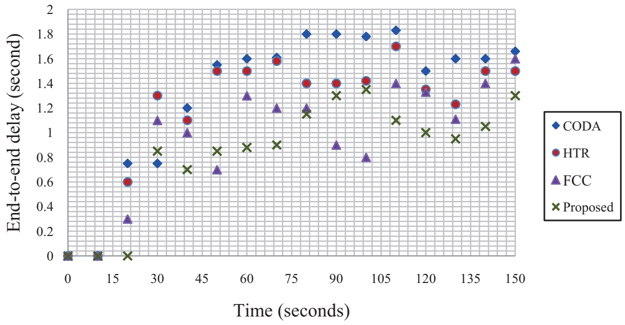

Fig. 10. Comparison in terms of end-to-end delay (α = 0.4).

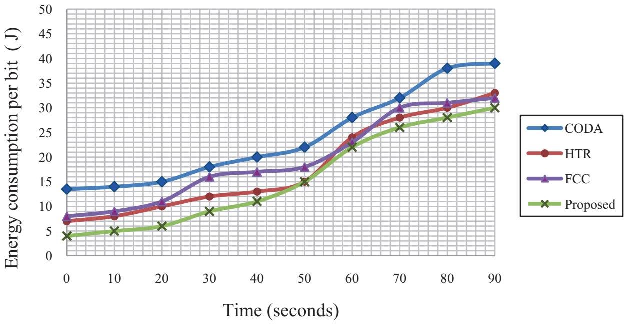

Fig. 11. Comparison in terms of the average energy consumption (α = 0.4).

Fig. 10 demonstrates the performance of CODA, HTR, FCC, and the proposed protocol in terms of end-to-end delay. It is obvious from Fig. 10 that the end-to-end delay of CODA, HTR, and FCC is higher than the proposed method. One important thing that should be considered here is that the end-to-end delay of the proposed method has increased (in second scenario) in comparison with first scenario. However, the end-to-end delay of the proposed method is still less than other methods but this increasing reveals the importance of our novel queueing model. For example, in second 140, the values of end-to-end delay for CODA, HTR, and FCC are 1.665, 1.520, and 1.412, respectively. On the other hand, this value is 1.051 for the proposed method.

Fig. 11 shows the average energy consumption per bit for CODA, HTR, FCC, and the proposed method. With respect to Fig. 11, the proposed method consumes less energy than CODA, HTR, and FCC. It can be argued that the energy consumption is 29 J for CODA in second 63, for HTR in second 72, for FCC in second 68, and for the proposed protocol in second 81. The energy consumption is 22 J for CODA in second 51, for HTR in second 53, for FCC in second 57, and for the proposed method in second 61.

7. Conclusion

In this paper, congestion-aware routing and fuzzy-based rate controller for WSNs have been proposed. Obviously, upstream traffic is prone to congestion. Hence, in most of the WSNs applications we are confronting with upstream traffic, it is crucial to provide an efficient solution to mitigate the potential congestion. To this end, we proposed a queueing model for assigning priority to packets based on the data value. Furthermore, a greedy congestion-aware routing approach was used for selecting the most appropriate paths. Additionally, by applying congestion score and buffer occupancy to a fuzzy rate controller, the offspring nodes were able to adjust and regulate their data transmission rate to their parent nodes. In addition, the simulation results clearly showed that the proposed method is superior to the previous methods, including CODA, HTR, and FCC in terms of packet loss rate, end-to-end delay and the average energy consumption.

References

[1] SHEN, X., HUANG, J., LIU, P., PAN, Y., XU, B., RAO, Z. Analysis of collaborative beamforming for wireless sensor networks with phase offset. *Radioengineering*, 2014, vol. 23, no. 1, p. 421–429.

[2] SIMEK, M., MORAVEK, P., KOMOSNY, D., DUSIK, M. Distributed recognition of reference nodes for wireless sensor network localization. *Radioengineering*, 2012, vol. 21, no. 1, p. 89–98.

[3] KAKITANI, M., BRANTE, G., SOUZA, R. Energy efficiency analysis of a two dimensional cooperative wireless sensor network with relay selection. *Radioengineering*, 2013, vol. 22, no. 2, p. 549–557.

[4] HATAMIAN, M., BARATI, H., MOVAGHAR, A., NAGHIZADEH, A. CGC: centralized genetic-based clustering protocol for wireless sensor networks using onion approach. *Telecommunication Systems*, 2015, p. 1–18. DOI: 10.1007/s11235-015-0102-x

[5] HATAMIAN, M., BARATI, H., MOVAGHAR, A. A new greedy geographical routing in wireless sensor networks. *Journal of Advances in Computer Research*, 2015, vol. 6, no. 1, p. 9–18.

[6] KAFIA, M. A., DJENOURIB, D., OTHMANC, J. B., QUADJAOUTA, A., BADACHEA, N. Congestion detection strategies in wireless sensor networks: A comparative study with testbed experiments. *Procedia Computer Science*, 2014, vol. 37, p. 168–175. DOI: 10.1016/j.procs.2014.08.026

[7] HATAMIAN, M., AHMADPOOR, S. S., BERENJIAN, S., RAZEGHI, B., BARATI, H. A centralized evolutionary clus-

tering protocol for wireless sensor networks. In *Sixth International Conference on Computing, Communications and Networking Technologies (ICCCNT15)*. Texas (USA), 2015, p. 1–6. DOI: 10.1109/ICCCNT.2015.7395190

[8] RAZEGHI, B., OKATI, N., HODTANI, G. A. A novel approach to mathematical multiple criteria decision making methods based on information theoretic measures. In *Iran Workshop on Communication and Information Theory (IWCIT)*. Iran, 2015, p. 1 - 6.

[9] HATAMIAN, M., BARATI, H., BERENJIAN, S., NAGHIZADEH, A., RAZEGHI, B. Error control coding in optical fiber communication systems: an overview. *Advances in Computer Science: An International Journal*, 2015, vol. 4, no. 14, p. 70–80.

[10] WAN, C. Y., EISENMAN, S. B., CAMPBELL, A. T. CODA: Congestion detection and avoidance in sensor networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems*. Los Angeles (CA, USA), 2003, p. 266–279.

[11] IYER, Y. G., GANDHAM, S., VENKATESAN, S. STCP: A generic transport layer protocol for wireless sensor networks. In *Proceedings of 14th International Conference on Computer Communications and Networks (ICCCN)*. San Diego (CA, USA), 2005, p. 449–454.

[12] AKAN, O. B., AKYLDIZ, I. F. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 2005, vol. 13, no. 5, p. 1003–1016. DOI: 10.1109/TNET.2005.857076

[13] SAYYADA, J., CHOUDHARI, N. K. Hierarchical tree based congestion control using fuzzy logic for heterogeneous traffic in WSN. *International Journal of Current Engineering and Technology*, 2014, vol. 4, no. 6, p. 4136–4143.

[14] GHOLIPOUR, M., HAGHIGHAT, A. T., MEYBODI, M. R. Hop-by-hop traffic-aware routing to congestion control in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2015, vol. 2015, no. 15, p. 1–13. DOI: 10.1186/s13638-015-0241-5

[15] SWATHIGA, U. U. A. S., CHANDRASEKAR, C. An efficient fuzzy based congestion control technique for wireless sensor networks. *International Journal of Computer Applications*, 2012, vol. 40, no. 14, p. 47–55.

[16] HEINZELMAN, W. B., CHANDRAKASAN, A. P., BALAKRISHNAN, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 2002, vol. 1, no. 4, p. 660–670. DOI: 10.1109/TWC.2002.804190

[17] HATAMIAN, M., BARATI, H. Priority-based congestion control mechanism for wireless sensor networks using fuzzy logic. In *Sixth International Conference on Computing, Communications and Networking Technologies (ICCCNT15)*. Denton (TX, USA), 2015, p. 1-5. DOI: 10.1109/ICCCNT.2015.7395203

[18] CHEN, G., PHAM, T. T. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems.* CRC press, 2001.

[19] ROJAS, R. *Neural Networks: A Systematic Introduction.* Berlin: Springer-Verlag, 1996.

[20] RAZEGHI, B., HATAMIAN, M., NAGHIZADEH, A., SABETI, S., HODTANI, G. A. A novel relay selection scheme for multi-user cooperation communications using fuzzy logic. In *Proceedings of Networking, Sensing and Control (ICNSC)*. Taipei (Taiwan), 2015, p. 241–246.

About the Authors . . .

Majid HATAMIAN received his B.S. degree in Computer Hardware Engineering in 2013, Iran. Furthermore, he received his M.S. degree in Computer Systems Architecture Engineering with first class honours from Azad University in Iran in 2015. His major research interests include wireless networks and mobile communications, efficient routing protocols, cryptography & data protection, security/privacy issues in computer networks and machine learning systems.

Maryam ALMASI BARDMILY is currently working toward the M.S. degree in Computer Systems Architecture Engineering at Ashtiyan Branch, Azad University, Iran. His major research interests include computer networks, artificial intelligence, intrusion detection systems and cloud computing.

Mojtaba ASADBOLAND is currently working toward the M.S. degree in Computer Systems Architecture Engineering at Dezful Branch, Azad University, Iran. His major research interests include wireless communications, networks-on-chips and computer architecture.

Mehdi HATAMIAN is currently working toward the B.S. degree in Information Technology Engineering at Payame Nur University, Iran. His major research interests include wireless communications, dynamic routing, network managements and resource allocation.

Hamid BARATI received his B.S., M.S., and Ph.D. degrees in Computer Systems Architecture Engineering in 2005, 2007 and 2015, respectively. Currently he is a faculty member at the Department of Computer Engineering, Islamic Azad University, Dezful Branch, Iran. His major research interests include mobile ad-hoc networks, interconnection networks & energy-efficient routing and security issues in wireless sensor networks.