A Fuzzy System Based Routing Protocol to Improve WSN Performances

Bakary Hermane Magloire Sanou, Mahamadi Boulou,
and Tiguiane Yélémou(✉)

Nazi BONI University, Bobo-Dioulasso, Burkina Faso
tyelemou@gmail.com

Abstract. Wireless sensor networks (WSN) have become very popular in recent years. Once deployed, WSNs are very rigid in terms of reconfiguration. Software Defined Networking (SDN) technology is being explored by several researchers to facilitate the reconfiguration of WSN nodes. Several architectures have been proposed, among SDN-WISE. SDN-WISE separates the data plane executed by the sensor nodes and the control plane executed by a software program hosted in a controller. In SDN-WISE, the choice of data transmission path is the best path in terms of hops count. One problem with this approach is that the chosen path is used until one of its nodes exhausts its energy before a path change process may be initiated. This impacts network efficiency and reduces the network lifetime. We then propose the Fuzzy Routing Protocol (FRP) which relies on the fuzzy system to compute a cost based on the metrics residual energy, RSSI, number of packets in the queue (buffer) and number of hops to reach the sink, to calculate the cost of each node. Nodes with the high cost close to sink are chosen to form the path. When a node in the path used for data transmission has its cost decreased by K% compared to its previous cost, then a new path is computed even if it is less optimal in terms of number of hops compared to the previous one. This approach allows a better distribution of energy consumption in the network and better congestion management.

Keywords: WSN · SDN · Energy consumption · Fuzzy inference system · Routing protocol

1 Introduction

Wireless Sensor Networks (WSN) are becoming more and more indispensable in our daily life. These networks composed of sensor nodes are of great use in many areas for various applications. These autonomous sensor nodes are most often deployed in thousands in environments that are difficult to access. Unlike traditional networks that are directly connected to electricity and with fairly high resources, these nodes are limited in terms of resources such as: memory capacity, storage space, computing power and especially on-board energy [8]. However, once deployed, WSNs typically face a difficulty in reconfiguring and managing

R. A. Saeed et al. (Eds.): AFRICOMM 2022, LNICST 499, pp. 33–49, 2023.
https://doi.org/10.1007/978-3-031-34896-9_3

all of these sensor nodes. To intervene on the sensor nodes in the WSN, the administrator had to go on each sensor node and make changes. This method is very difficult to achieve if there are a large number of sensors to reconfigure or troubleshoot. To facilitate this process, a new technology has appeared, the software defined networking (SDN) [1]. The principle of this technology is the separation of the control plane from the data plane. The control of the data is thus transferred to a software entity called a controller. Thus, with a centralized view of the network, it allows an automatic, flexible and dynamic reconfiguration of the network. This technology is mainly intended for networks with an infrastructure to facilitate the administration of network equipment, including simplifying their configuration. In order to solve the problem of reconfiguration complexity in WSNs, the researchers proposed to adopt this new SDN technology. Several architectures have emerged including SDN-WISE. SDN-WISE is an SDN solution adapted to WSN [2]. Unlike existing SDN solutions for wireless sensor networks, SDN-WISE is a stateful solution. One of its goals is to reduce the amount of information exchanged between the sensor nodes and the network controller. This approach uses Dijkstra's algorithm for route calculation. The nodes on the shortest paths in terms of number of hops are more stressed and therefore discharge their energy faster than the others. This also leads to congestion at these nodes and consequently increases the packet loss rate and packet transmission delays. To improve the efficiency of the SDN-WISE architecture, we propose the Fuzzy Routing Protocol (FRP). Based on energy metrics, Received Signal Strength Indicator (RSSI), number of packets in the queue and number of hops to reach the sink, FRP relies on a fuzzy logic system to determine the cost of each node. Thus, the nodes with the highest cost that are likely to be close to the sink are chosen as the next hops to form the path. This approach not only allows a better distribution of energy consumption in the network, but also manages congestion. It improves the risk of packet loss and therefore decreases transmission delays. The remainder of the paper is organized as follows. In Sect. 2, we present related work. A brief overview of SDN-WISE is given in Sect. 3. In Sect. 4, we present and analyze the performance of our FRP routing solution compared to standard SDN-WISE. Finally, we conclude in Sect. 5.

2 Previous Work

The adoption of SDN has changed the way wireless sensor networks are designed. It is involved in architecture, deployment, operation and maintenance. Software defined networking is an emerging research topic. It focuses on facilitating network management and scalability. In these types of networks, the data plane is clearly separated from the control plane. That is, the traditional data transmission logic is defined in the control plane and implemented in the network devices. Therefore, the controller is responsible for making routing decisions and sending them to the network nodes.

Angelos-Christos G. Anadiotis and al. [3] proposed an entire SD-WISE software-defined wireless sensor network solution. Their solution integrates

Openflow features adapted to WSNs. In their work, energy consumption is reduced by efficiently managing the duty cycle in the sensor nodes and controlling the transmission power of their radio resources. They use the operating system of the wireless sensor nodes to virtualize network functions (routing protocol). Also, they ensure regulatory behavior compliance according to the remotely recognized authority (through software and mechanisms that attest the sensor nodes supported by Trusted Platform Modules (TPM)), based on the context of the nodes, exploiting only the interaction between the hardware and the trusted software. Their results showed a high packet sending delay due to the large proportion between the data size and the header size.

A new approach based on fuzzy logic has been thought by N. Abdolmaleki and al. [4], called Fuzzy Topology Discovery Protocol (FTDP). This approach was on an SDN-WISE architecture to discover the network topology and route. It takes into account several properties of the nodes in the network in order to choose the best routing node and thus improve the network performance. These are: number of neighbors, workload level (queue length), remaining energy. The results of their proposed solution use fuzzy systems in the control plane to increase the packet transfer rate, reduce packet loss and improve the energy consumption of the network.

Due to the channel bandwidth for data and control traffic in SDN-based WSNs, control messages are a bottleneck, disrupt network performance and controller responsiveness. So, a flow configuration request control algorithm (FR_CMQ) was proposed by M. Ndiaye and al [5] to reduce the duplication of flow configuration request control messages. The results showed a significant reduction of control messages, thus leading to a decrease in energy consumption. But FR_CMQ affects the packet delivery rate and delay. This could be due to the lack of a loss control or delivery success mechanism for initial flow configuration request messages.

The implementation of QoS brings challenges such as the difficulties of adaptability and implementation in traditional network architecture. For this purpose, X. Tan and al [6] presented QSDN-WISE, a hierarchical software defined network architecture for wireless sensor networks. It enables complex network management and makes the system more adaptable. The DCHUC clustering algorithm in QSDN-WISE uses a non-uniform clustering mechanism. It ensures that the cluster head nodes are (as close as possible) to the sink node. This avoids energy holes and increases the lifetime of the network. In addition, DCHUC determines two cluster head nodes, one with the least congestion and good link stability for the other, based on criteria of congestion, link stability and residual energy of the nodes. These two types of cluster heads not only form two hierarchical network topologies, but also reduce the load on each cluster head and maintain the quality of intra-cluster services. The QSDN-WISE centralized routing algorithm considers residual energy, node congestion, link stability and distance between nodes as parameters for choosing the next hop. Based on the two topologies and data classification, it constructs two heterogeneous routing paths for nodes to meet the requirements of different data classes. The results show that QSDN-WISE can not only balance the energy consumption of WSNs, but also provide QoS support for data with different QoS requirements. And it performs better in energy saving, end-to-end delay, packet loss rate and message count control compared to SDN-WISE, SDN-DMRP and IRPL.

To improve traffic distribution, J. Schaerer and al. [1] implemented the dynamic traffic aware routing protocol DTARP. It takes into account the centrality and dynamic traffic statistics of nodes when computing the path. With this information, more active central nodes are recognized and are less eligible for retransmissions. Therefore, less active nodes are chosen even if they have a slightly higher hop count. In addition, it is possible to have several paths of the same length between two nodes. Therefore, a cost is determined and then associated to each link by a cost function defined in the algorithm, based on the average traffic on the link and the radio signal strength indicator (RSSI). The link with a low cost is chosen. The results showed that DTARP could reduce the network activity of the most active node by 25% compared to the SDN-WISE protocol. This traffic distribution will increase the overall network lifetime, as the lifetime of the most active nodes is increased.

N. Q. Hieu and al. [7] proposed the idea of using an automatic timer mechanism. It adjusts the communication rate between the controller and the sensor nodes. The generation of control packets (Beacon and Report) is regulated by this timer called “Trickle Timer” in order to optimize the network performance. The Trickle Timer allows the sensor nodes to exchange a few packets during a given time if the network state is stable. Trickle Timer is inspired by the Trickle algorithm in RPL. It runs at defined time intervals. At the first half of the interval called the listening period, each node looks at the transmissions of its neighbors. If the information received from the neighbors is consistent, a counter is incremented. And in the second half of the time interval the node sends its information to its neighbors only if the counter value is below a predefined redundancy constant. Otherwise, the packets in its queue are dropped. When an inconsistent transmission (discovery of a new neighbor) or an external event (defined by the controller) is detected, the Trickle algorithm resets. The results show that the implementation of the Trickle timer in SDN-WISE provides better performance in terms of power consumption and transmission and reception duty cycle.

3 SDN-WISE Approach

In this approach the exchange and processing of the appropriate packet called TD packet is paramount [2]. This packet carries information about the battery level and the number of hops from the (nearest) sink. Each time a node receives this packet, it compares it with the current best hop and then chooses the next best hop. The choice is made according to the priority to the number of hops, then to the RSSI value received with the message and finally to the residual energy level. This information also feeds a list containing the WISE neighbors.

In this list we have the addresses of the neighbor nodes, their RSSI and their battery levels. It is sent periodically to the topology management (TM) layer in order to establish a graphical representation of the network. Then, the table is completely emptied and rebuilt with the incoming TD packets in order to always have an up-to-date view of the network topology. One of the controllers acts as a proxy between the physical network and the other controllers. It is called WISE-Visor like FlowVisor in traditional OpenFlow networks. The controllers define the network management policies to be implemented by the WSN and are often application dependent. As a result, controllers can interact with the application. Sensor nodes have limited memory capacity which makes it important to choose the size of the different data structures. This size depends on several deployment-specific characteristics defined by WISE-Visor during the initialization phase. To extend the life of the network, traffic should be distributed as evenly as possible across the network. With SDN-WISE, packets are sent on the shortest path. This approach uses the Djikstra algorithm for route calculation. Nodes on the shortest paths in terms of hop count are found to be more stressed and therefore deplete their energy faster than nodes on the edge of the network.

4 Fuzzy Routing Protocol Based on SDN-WISE

4.1 Principle of Our FRP Approach

Our FRP approach is a new packet routing approach that enables load balancing, optimization of sensor node energy consumption, reduction of transmission delay and packet loss rate. The Dijkstra algorithm in SDN-WISE, in order to estimate the shortest routing path, prioritizes the number of hops as the routing metric, followed by the received RSSI value and finally the residual energy level. Instead of this protocol, we propose the centralized routing approach called Fuzzy Routing Protocol (FRP) to determine the best paths by a combination of several metrics. With FRP, the choice of a path will be conditioned by a computed cost of each node of this path. Thus, a relatively long path may be chosen for data transmission at the expense of a shorter path. To enable efficient balancing of energy consumption and achieve optimal network load balancing, the best cost (highest cost node) is required for route selection. FRP uses the fuzzy system that calculates a cost corresponding to each node. This cost is determined based on the reconciliation of four (04) decision parameters that are the residual energy, the number of packets in the queue of the node, the RSSI between the node and each of its neighbors and the number of hops to reach the sink node. These four parameters define for us the performance level of the nodes for their eligibility as intermediate node of the path for data transmission. Once these costs are calculated, the controller establishes the different routing paths in the network to reach the destination. This information is sent back to the source nodes and other intermediate nodes so that the intermediate nodes simply relay the packets along pre-calculated paths. The state of the nodes varies over time due to changes in the values of the decision parameters, often at a high frequency (the number of packets in the queue, etc.). To mitigate the impact of these dynamic changes on routing decisions, a threshold K defining the level of performance degradation of the nodes is set. So, if the new calculated cost of a node on a path is less than K% of its current cost, this means that the node has lost performance, but the change has not yet occurred. The controller then searches again for a path whose cost degradation has not yet reached K% among the possible paths to choose the new best path.

4.2 FRP Flowchart

The following variables provide a better understanding of the proposed flow chart:

- K: constant of the degradation level of a node in %;
- Ni: designates a given node;
- Nv: designates a neighboring node of Ni;
- V: the set of neighbors of a node Ni or next jumps;
- P: the set of nodes in a given path;
- C(Ni): cost of a node Ni;
- C(Nv): cost of a neighboring node Nv.

After the initialization of the constant K, the following sequences of actions are performed by the controller to find an optimal routing path:

- First, it looks for the one with the highest cost C(Nv) among the neighbors V of the source node. This neighbor node Nv is then chosen and will be the first intermediate node in the set P of the routing path for the source node.
- Then for this first intermediate node, it also searches among its neighbors V for the node that has the highest cost C(Nv). This new neighbor node Nv is chosen as the second intermediate node in the routing path set P.
- Then the process is repeated in this way until the set P of the complete path between the source node and the sink is established.
- The fuzzy inference system is executed each time, after receiving the control packets from the sensor nodes.
- If the cost of a node belonging to a routing path decreases by K%, this means that the node has lost performance. Then the controller searches again for the best path among the possible paths including the nodes with the highest costs. Otherwise the same path is maintained for data routing.

FRP flow chart is illustrated in Fig. 1.

4.3 Calculating the Cost of a Node

Fuzzy Inference System. The fuzzy logic reasoning system allows us to transform several input metrics (residual energy, RSSI, number of packets in the queue and number of hops) into a single output value (cost). The operation of the fuzzy inference system can be summarized in these main steps: fuzzification, fuzzy inference, aggregation and defuzzification. We use the fuzzy inference model of Mamdani because of its simplicity and efficiency [8,9].

To illustrate the FRP routing mechanism, we consider the following network architecture. Node N1 wants to send data to the sink. We assume that the values of the decision parameters taken from node N1 are as follows: energy 7, RSSI 80, number of packets in the queue 5 and number of hops 2.

Architecture of the fuzzy inference engine To directly combine the four linguistic input variables into a single output, would entail the management of 3 * 3 * 3 * 4 = 108 possible combinations of the different membership functions of the inference engine. To avoid the complexity associated with this management, we organize the inference system into three functional blocks as illustrated in the figure below. In the first step, energy and RSSI are combined to produce the linguistic variable ER (Energy and RSSI). In a second step, the number of packets in the queue and the number of hops are combined to produce the linguistic variable FS (Queue and Hops). Subsequently, these last two are combined together to produce a single "cost" output. This last linguistic variable provides information about the ability of a node to serve as a relay.

Fuzzification. Fuzzification is the step in which we make sense of or interpret the input variables of our decision model [8]. Instead of belonging to the "true" or "false" set of traditional binary logic, fuzzy logic admits degrees of belonging to a given fuzzy set. For this purpose, several functions are available (triangular, sinusoidal, trapezoidal, etc.), but we will use the trapezoidal function to measure the degree of membership of input variables to the corresponding fuzzy set.

First step fuzzification In the first step, we combine the residual energy and RSSI to calculate ER which in turn is considered in the next step. The initial energy of each node is considered to be 10 and the RSSI varies between 0 and 100. The energy variable can belong depending on the value of the scalar input to the fuzzy sets: low, medium and full (for a low, medium or full battery respectively). The RSSI variable can belong according to the value of the scalar input to the fuzzy sets: weak, moderate and strong (for a weak, moderate or strong signal respectively). Their membership functions are represented in Fig. 8 and Fig. 9.

As assumed for node N1, with an energy equal to 7, the projections made on the fuzzy sets indicate us a full value at 100% and a medium value at 0%. As for the RSSI = 80, a similar reasoning leads us to find for the very high and high fuzzy sets, the degree of membership of 50% each.

Table 1 summarizes the relationship between these two input variables and the output ER (Energy and RSSI). It is described by the fuzzy sets: very low, low, medium, high, very high. The table is based on the idea that "the lower the RSSI and energy, the lower the ER output.

Table 1. Output linguistic variable ER

RSSI/Energy	LOW	MEDIUM	FULL
Very low	Very low	Low	Low
Low	Low	Medium	Medium
Medium	Low	Medium	High
High	Medium	High	High
Very high	Medium	High	Very high

To determine the output, we use Mamdani's inference model, and use the probabilistic operator "AND" as the composition function and the maximum as the aggregation operator. For node N1, the controller computes two non-zero ER membership functions: very high (ER) = 0.50 and high (ER) = 0.50. These values are defuzzified (using the procedure described below) into a single ER output (ER = 0.81), and then used in step 3 of the fuzzification phase.

Second stage fuzzification In the second step, we will combine the number of packets in the queue and the number of hops to calculate FS, which is also taken into account in the next step. We assume that the buffer of our nodes can only hold 10 packets and as for the number of hops, it is 100 at most. The variables, number of packets in the queue and number of hops, vary from small, medium to large and short, medium to large respectively. The Fig. 10 and Fig. 11 represent their membership functions.

Now for the number of packets in the queue and the number of hops of node N1, we obtain respectively the degrees of membership in the medium 100 The relationship between these two input variables and the FS output (Queue and Jumps) is shown in Table 2 below.

Table 2. Output linguistic variable FS

Queue — Number of hops	Short	Medium	Large
Small	Very high	high	low Very
Medium	high	Medium	low
Large	Low	Very low	Very low

Here the controller calculates a non-zero FS output membership function: high (FS) = 1. This value is defuzzified (according to the defuzzification procedure described below) into a single FS output (FS = 0.70), then used in step 3 of the fuzzification phase.

Third stage fuzzification In this step, we combine ER (Energy and RSSI) and FS (Queue and Hops) according to the relationships established in Table 3 below to determine the node cost. The fuzzification of the variables ER and FS corresponds to the degrees of membership in the very high (ER) 50%, high (ER)

50% and high (FS) 100% fuzzy sets, respectively. These variables vary from very low, low, medium, high, very high fuzzy sets.

Table 3. Output linguistic variable cost

FS — ER	Very low	Low	Medium	High	Very high
Very low	Very low	Very low	low	low	medium
Low	Very low	low	low	medium	medium
Medium	low	medium	medium	high	high
High	low	medium	high	high	Very high
Very high	medium	medium	high	Very high	Very high

Defuzzification. As the name suggests, defuzzification is the reverse operation of fuzzification. All fuzzy values obtained after the inference and aggregation steps are converted into a single concise result [4]. Among the many defuzzification methods (average of maxima, centers of gravity) proposed in the literature, the center of gravity method is the most widespread and attractive of all defuzzification methods. The defuzzification process of the linguistic variable "cost" is presented in Fig. 12 below.

The output value indicates what is the level of cost to choose a node as the next hop when computing routing paths, according to the selected metrics. For the proposed topology, two cost output membership functions are enabled at node N1: very high 0.50 and high 0.50. The center of gravity of the represented region is 0.81.

4.4 Analytical Study of Our Approach Performance

In this section, we considered a WSN where similar routing cost calculations performed for all nodes in the network. As illustrated in the Sect. 4.3, the calculated costs are shown in the following Figure 2

As a preliminary evaluation, we perform an analytical study comparing the FRP and the standard SDN-WISE mode of operation. Node N1 must be able to transmit data to the sink. To do so, we apply the two routing approaches: the SDN-WISE standard and the FRP on an SDN-WISE network architecture, then we present the chosen paths. Case of Dijkstra-based SDN-WISE routing. In this first case, the choice of the route is based on the shortest path in terms of number of hops. The path used by node N1 would therefore be N1-N2-N6-sink. For the case of FRP-based routing, we assume K = 25% as the level of node performance degradation. That is, if a node on the routing path were to have a cost lower than the previous one by 25%, then a new routing path is computed. The FRP looks for paths consisting of nodes close to the sink in terms of number of hops and with a high cost. Step 1: the calculated path from node N1 to reach the sink would be N1-N2-N4-N7-sink (see Fig. 3). To choose this path, the controller first checks which of the neighbors of node N1 has the highest cost. Thus, N2 is chosen to be the first node of the path. The same process is performed to then choose node N4 as the next hop in the path. At this level, the choice of the next hop of the path is between the nodes N6 and N7 which are closest to the sink. And N7 having the highest cost is then chosen to reach the sink.

Step 2: at this step, the new calculated cost of node N2 = 0.59. We see that this cost is less than the initial cost by 25%, i.e. a difference of 22. Then the control recalculates a new path for node N1. Thus, the new path would be N1-N3-N5-N7-sink (see Fig. 4).

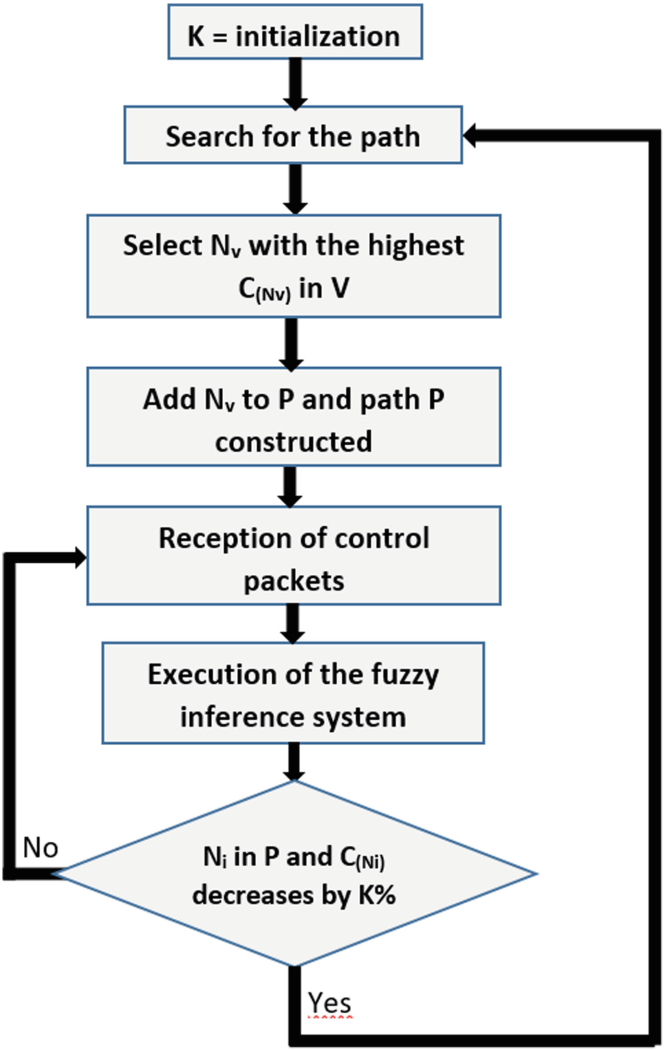

Fig. 1. FRP flow chart

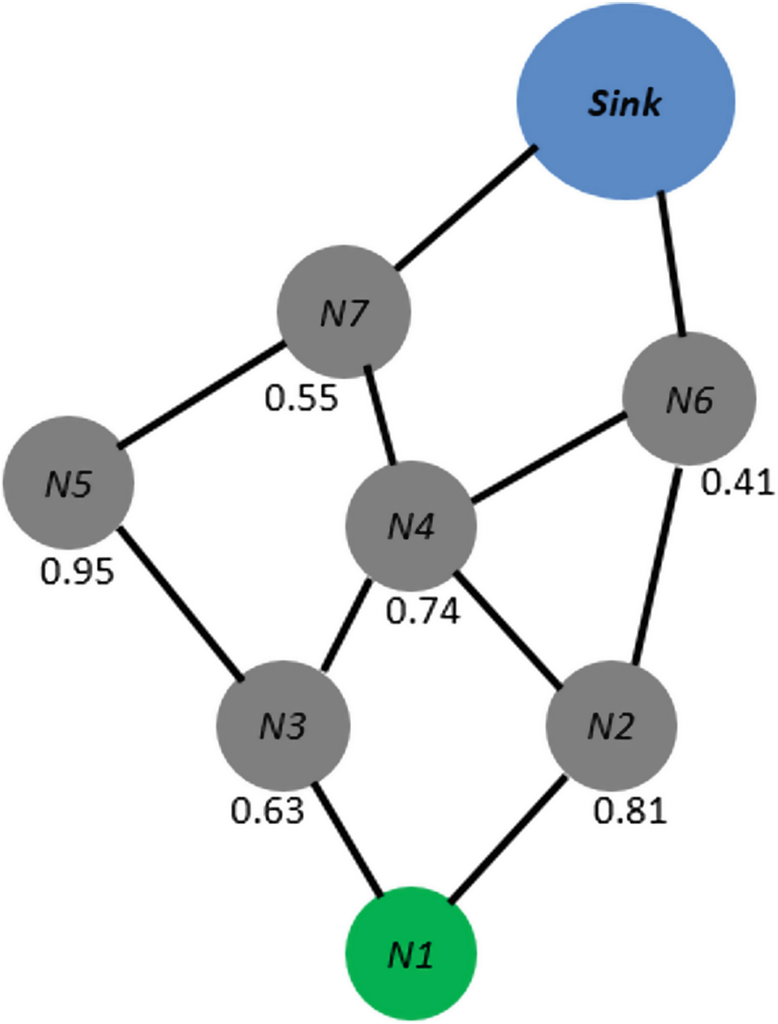

Fig. 2. Calculated costs of nodes

In a standard SDN-WISE, we note that when an optimal path (shortest path) is used to send data, that path remains until a node in the path exhausts all of its energy, before moving to another path.

In this approach, core nodes discharge faster since they belong to the shortest paths. While others with a large amount of energy are less used for data transmission, due to belonging to less optimal paths. This route selection method reduces the lifetime of the network. Indeed, we assume that if one of the nodes of the used path falls (runs out of energy), the system is no longer efficient and we conclude the end of the network mission.

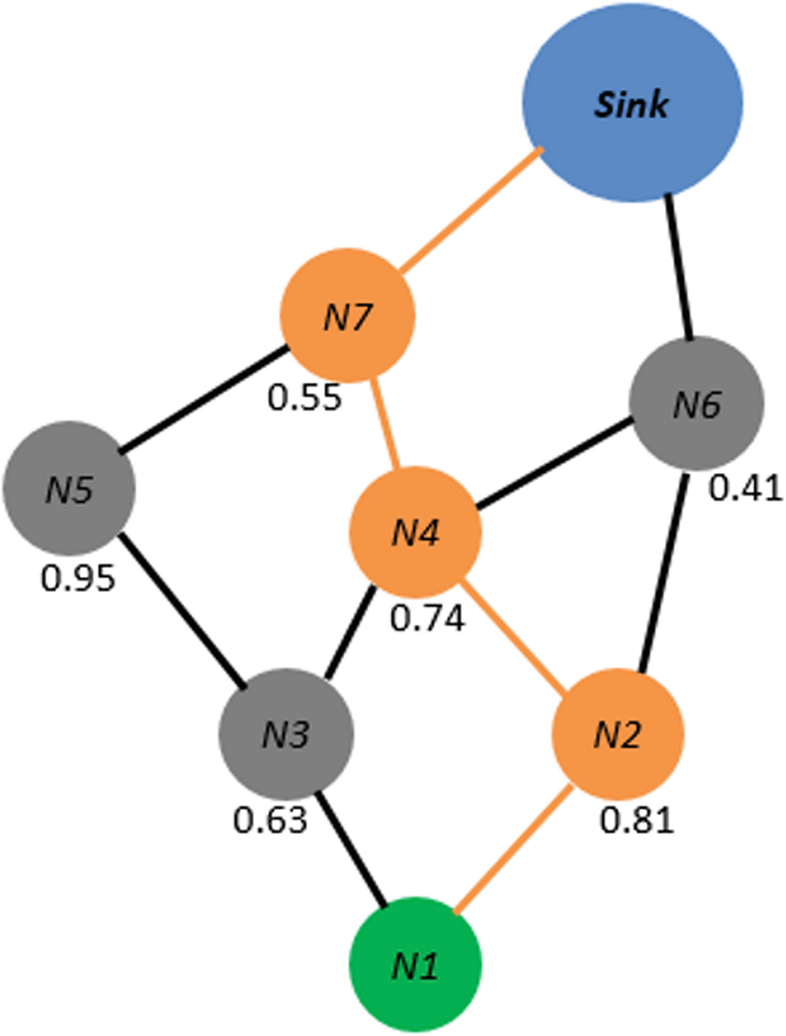

Fig. 3. Initial routing path for N1

In FRP routing, the paths chosen for data transmission are not necessarily the shortest. Instead, the nodes in the paths are the ones with the highest costs among the eligible nodes during the formation of the path. When a node in the path has its cost reduced by K% compared to the old one, then a new path is computed even if it is less optimal in terms of number of hops compared to the previous one.

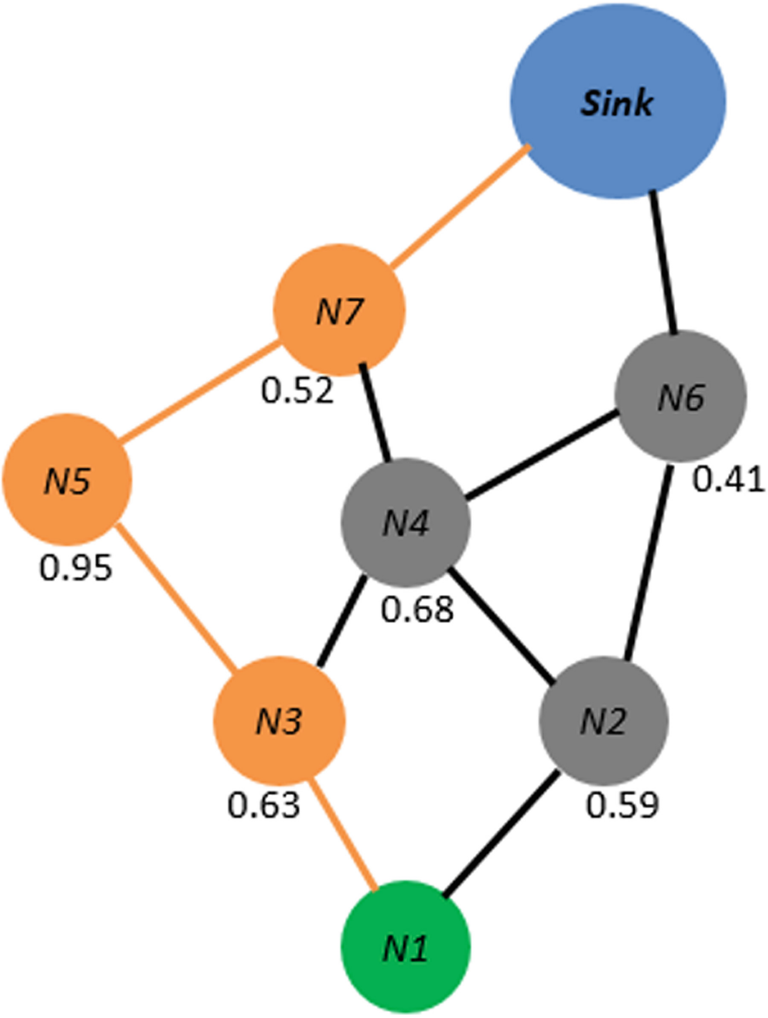

Fig. 4. New routing path for N1

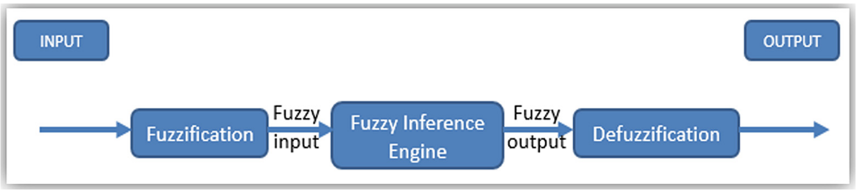

Fig. 5. Fuzzy inference system

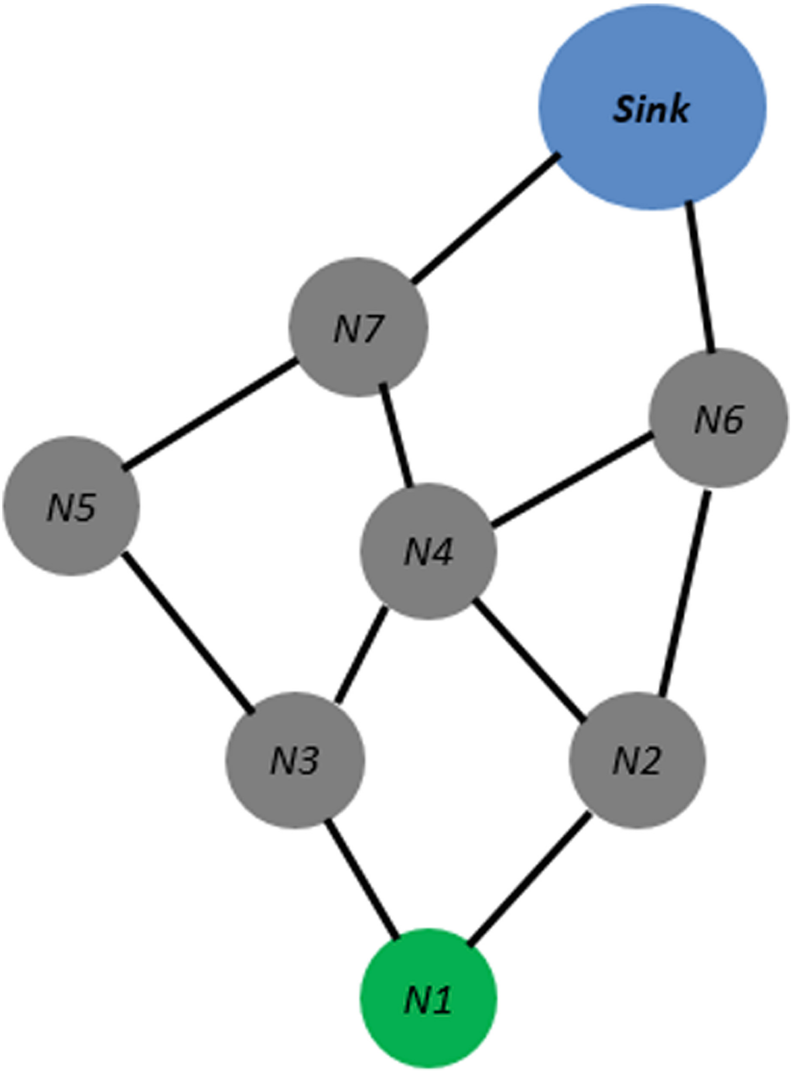

Fig. 6. Network architecture

From this analytical study, we can state that FRP increases the network lifetime compared to the standard SDN-WISE routing mode. However, intensive simulations of its enhanced route selection process should confirm this claim (Figs. 5, 6 and 7).

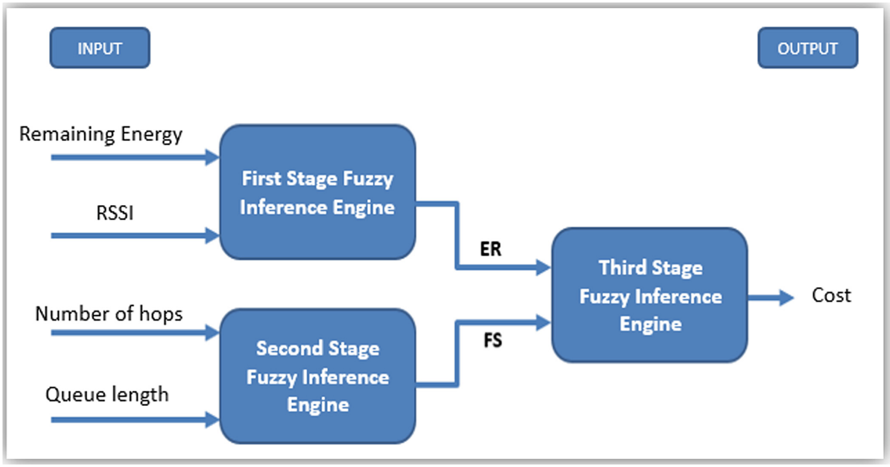

Fig. 7. Fuzzy inference engine

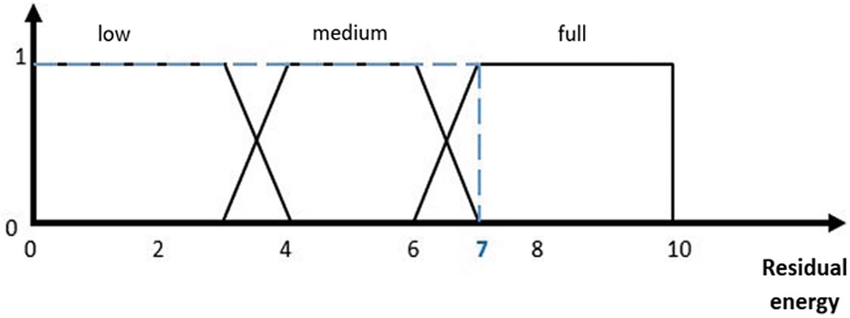

Fig. 8. Residual energy membership function

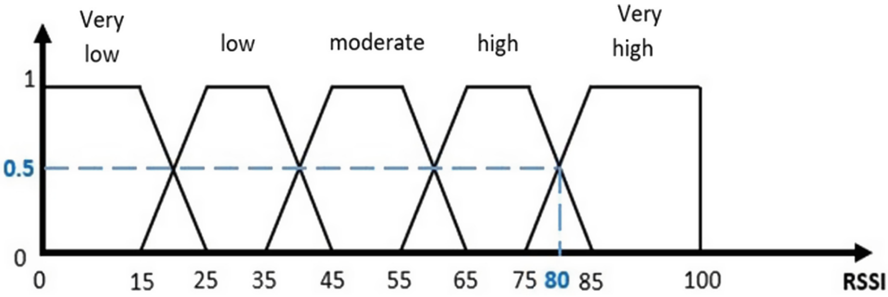

Fig. 9. RSSI membership function

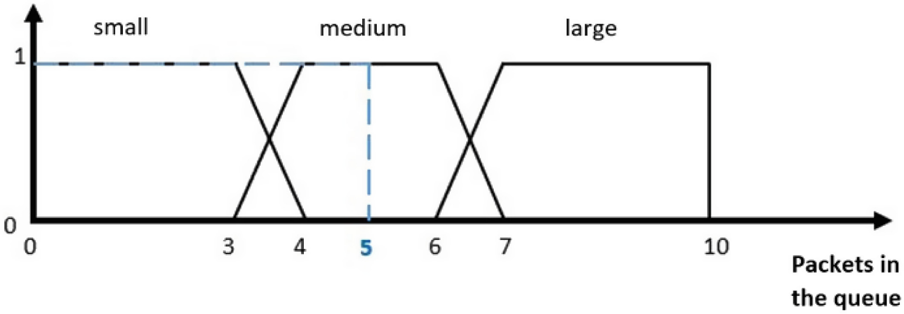

Fig. 10. Membership function for the number of packets in the queue

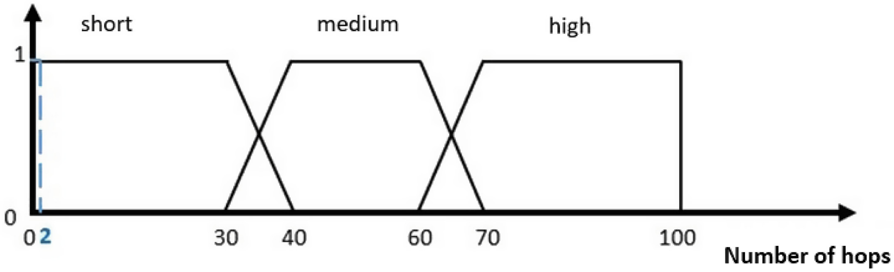

Fig. 11. Number of hops membership function

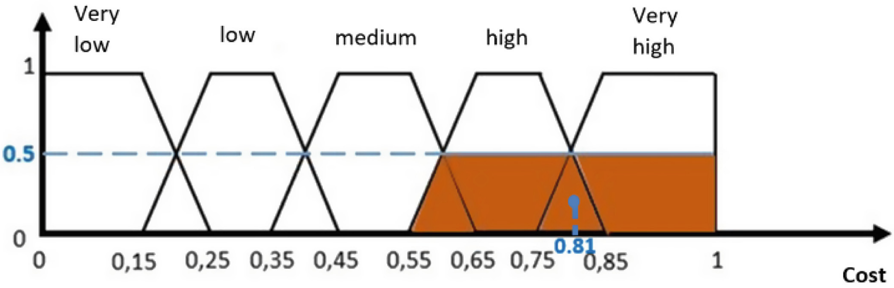

Fig. 12. Cost defuzzification

5 Conclusion

In our work, we have focused on the issue of optimizing energy efficiency in routing and congestion management of nodes in WSN especially SDN-WISE. We have seen that wireless sensor networks consist of sensor nodes with energy, storage and processing constraints. Also, we have highlighted the uneven distribution of energy consumption, network instability and the risk of overloading the sensor nodes in the centralized routing approach based on SDN-WISE. In view of these, we then proposed the Fuzzy Routing Protocol (FRP) which relies on the fuzzy system to compute a cost based on the metrics energy, RSSI, number of packets in the queue and number of hops to reach the sink. The nodes with high costs and close to the sink are chosen as the next hops to constitute the path. The analytical study shows that this new approach improves the lifetime of the network. It not only allows a better balancing of the energy consumption in the network, but also manages the congestion. It then improves the risk of packet loss and thus decreases the transmission delays. Simulation tests are needed to confirm the performance of this new approach.

References

1. Schaerer, J., Zhao, Z., Braun, T.: DTARp: a dynamic traffic aware routing protocol for wireless sensor networks. In: RealWSN 2018 - Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks, Part of SenSys 2018, pp. 49–54 (2018)
2. Galluccio, L., Milardo, S., Morabito, G., Palazzo, S.: SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In: Proceedings - IEEE INFOCOM, vol. 26, pp. 513–521 (2015)
3. Anadiotis, A.-C., Galluccio, L., Milardo, S., Morabito, G., Palazzo, S.: SD-WISE: a software-defined wireless sensor network. Comput. Netw. **159**, 84–95 (2019). https://www.sciencedirect.com/science/article/pii/S1389128618312192
4. Abdolmaleki, N., Ahmadi, M., Malazi, H.T., Milardo, S.: Fuzzy topology discovery protocol for SDN-based wireless sensor networks. Simul. Model. Pract. Theory **79**(Dec), 54–68 (2017)
5. Ndiaye, M., Abu-Mahfouz, A.M., Hancke, G.P., Silva, B.: Exploring control-message quenching in SDN-based management of 6LoWPANs. In: IEEE International Conference on Industrial Informatics (INDIN), vol. 2019-July, pp. 890–893 (2019)
6. Tan, X., Zhao, H., Han, G., Zhang, W., Zhu, T.: QSDN-WISE: a new QoS-based routing protocol for software-defined wireless sensor networks. IEEE Access **7**, 61070–61082 (2019)
7. Hieu, N.Q., Thanh, N.H., Huong, T.T., Thu, N.Q., Van Quang, H.: Integrating trickle timing in software defined WSNs for energy efficiency. In: 2018 IEEE 7th International Conference on Communications and Electronics, ICCE 2018, pp. 75–80 (2018)
8. Kamgueu, P.O.: Configuration dynamique et routage pour l ’ internet des objets. To cite this version: HAL Id: tel-01687704 soutenance et mis à disposition de l ’ ensemble de la Contact: ddoc-theses-contact@univ-lorraine.fr (2018)
9. Kipongo, J., Esenogho, E.: Efficient topology discovery protocol for software defined wireless sensor network. Int. J. Electr. Comput. Eng. (IJECE) **9**, 19 (2020)