

Received June 22, 2021, accepted June 27, 2021, date of publication July 5, 2021, date of current version July 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094541

Fuzzy Logic-Based Path Planning for Data Gathering Mobile Sinks in WSNs

CHATCHAI PUNRIBOON¹, CHAKCHAI SO-IN¹, (Senior Member, IEEE),
PHET AIMTONGKHAM¹, AND NUTTHANON LEELATHAKUL²

¹Applied Network Technology (ANT) Laboratory, Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

²Faculty of Informatics, Burapha University, Chonburi 20131, Thailand

Corresponding author: Nutthanon Leelathakul (nutthanon@buu.ac.th)

This work was supported in part by the Thailand Science Research and Innovation (TSRI), in part by the National Research Council of Thailand (NRCT) through the International Research Network Program under Grant IRN61W0006, and in part by Khon Kaen University, Thailand.

ABSTRACT Mobile sinks (MSs) are capable of collecting data along specified paths in wireless sensor networks (WSNs). They are deployed as a popular alternative for data loggers, to which all nodes have to send sensory data. If MSs' paths (or cycles) are not well determined, it might take a relatively long time for the MSs to make a round trip. Recent research works have proposed methods to determine rendezvous points (RPs) that the MSs must pass by to collect data, with the aim of reducing the data-collection time. Determination of the number of RPs is important, and it is challenging to make ensure that there are sufficient RPs widely located throughout a sensor network, forming a circle along which the MSs can spend limited time traveling. This research presents a method for designing paths and pinpointing RPs for MSs to collect data, as well as determining the next hop to relay data for each sensor node. Instead of reducing the MSs' travel time, the focus of this research is to preserve the energy of all sensor nodes in WSNs. Our method determines the maximum number of RPs such that the MSs can run through each RP's communication range (within a time constraint) without depleting their own energy. The method comprises three main steps. First, we calculate the number of RPs and design the MS path. Second, the exact data-collection points are determined. The last step is to specify the path along which sensory data are relayed to the MS. In our experiments, we simulate two WSNs of different sizes. The results show that our method outperforms the others by 70%-80% in terms of the sensor node uptime, power consumption, MS traveling time and the number of RPs.

INDEX TERMS Data gathering, fuzzy logic system, mobile sink, path planning, rendezvous point, wireless sensor network.

I. INTRODUCTION

One important characteristic of a wireless sensor network (WSN) is the continuous communication among a large set of sensor nodes (SNs) that are responsible for collecting various data according to their function, such as detecting objects and motion and sensing temperature, humidity, and flicker [1]–[5]. The data are then relayed via the WSN until they reach the base station (i.e., the sink node) [6], which is responsible for storing data for further usage [3]. WSNs are often large, as they are composed of a large number of sensor-node devices for deployment in large areas, e.g., forests for wildfire detection [7], [8], nature

monitoring [9]–[11], and deep sea exploration [12], [13]. WSNs are also suitable for use in hard-to-reach areas, where wiring for data transmission can be costly and troublesome. Generally, SNs send sensory data to the sink node via another SN within their transmission range. Hotspot problems are one of the most common problems in WSNs. The SNs close to the sink node tend to lose power rapidly when relaying data. As a large number of distant SNs may exist, the relaying SNs close to the sink node will run out of power before the others [14], [15], causing the entire system to fail.

Mobile sinks (MSs) are often deployed to address the abovementioned problem. MSs travel close to the SNs to collect sensory data [16]–[21] so that the data does not have to be relayed by SNs along paths to the sink node, thereby reducing the SNs transmission power consumption. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio¹.

the travel times of the MSs often exceed the delay limits (DLs) [22]. A method for planning MS travel paths is therefore important as it helps determining the rendezvous points (RPs) [23], [24], each of which is an SN selected to be in charge of collecting data from other SNs within its range and relaying the aggregated data to the MSs as they arrive. Instead of spending considerable time visiting all SNs, the MSs only need to pass by the RPs to gather the aggregated sensory data from the associated SNs [23], [25]. The RPs must be pinpointed to cover the whole coverage area to reduce the power consumption. Nevertheless, an excessive number of RPs might cause the MS round-trip times to exceed the DLs.

When selecting RPs, we must consider the number, lifetime (remaining power), and the locations of RPs. For example, if a selected RP has low power, instability in data reception may occur. In the case where the selected RP is too far from the other SNs, the other SNs will waste more power on data transmission. As the MSs can communicate with any SN if it is within the SN's communication range, [26] proposed the idea of storing data at RPs located at the boundary of the SNs' communication range to reduce the MS's traveling distances.

However, storing data at such locations might require the SNs to spend more power on transmitting data long distances. To plan relay routes, previous research adopted the minimum spanning tree (MST) method [27], which could greatly simplify the system at the expense of substantial energy spent on relaying data via multiple hops [28]. Other previous works designed RPs to be close to SNs. Such approaches yield low performance in cases when SNs are far from the RPs.

Our proposed method adopts a fuzzy logic system (FLS), which involves the usage of appropriate weights, to assist decision-making processes. When selecting RPs, FLS assists in determining each SN's weight based on its remaining energy and location. The weight reflects the suitability (also the likelihood) of the SN being selected as an RP. When calculating the data-collection distances, FLS is used to help determine the distance from each SN the MS must reach to start collecting the data. The MS should closely approach low-powered RPs before collecting data.

On the other hand, an MS can collect data immediately once it reaches the boundary of the RPs' communication range if the RPs still have a high level of remaining power. When selecting routes, the FLS is responsible for helping each SN decide which node it should send data to among the closest RP and the other SNs.

This research proposes a route planning method that focuses on efficiently planning a route for MSs under the DL constraint while attempting to reduce the energy consumption of all SNs in WSNs. Our method has three phases. The first phase involves determining the number of RPs and planning the paths for the MS. In this phase, we apply FLS to rank each SN's potential to be an RP. We then determine the minimal number of RPs that the MS will spend time visiting before returning to the sink node such that the DL is not violated. Subsequently, we increase the number of RPs by choosing

more potential SNs within the MS's communication range while they are traveling along the path.

In the second phase, we determine areas for the MS to collect data. As the MS can communicate with an RP if they are in each other's communication range, our method does not require the MS to reach an RP before it starts collecting data. By calculating weights based on each RP's energy and the number of SNs within its communication range, we leverage FLS to determine the distance from each RP for the MS to collect data. In the last phase, in consideration of each RP's communication range and remaining power, the number of SNs within the range, and the distance from the SNs to itself, FLS is adopted to determine how SNs (not selected as RPs) send data to the MS (i.e., whether to send the data via the RP or via a neighboring SN.)

The structure of this paper is as follows: Section II describes relevant literature regarding path designs for the MS to collect data; the MS has to reach each RP and can visit only the area within the RPs' communication range. Section III defines our network models and energy model for conducting experiments and performance evaluation. Section IV presents a path design method consisting of the three main phases mentioned above. Section V presents the running time analysis and comparison with previous work. The system simulation settings and the efficiency of our proposed method are shown in Section VI. We discuss our design choices, our method's limitations, and future work in Section VII. The analysis and discussion of the experimental results are presented in Section VI, and the conclusions of the experimental results are presented in Section VII. Finally, Section VIII concludes our work.

II. RELATED WORKS

This section provides a review of the literature related to path design for MSs by determining data-collection locations and adopting FLS in WSN.

In 2010, Almi'ani *et al.* [22] discussed the use of MSs to collect data in WSNs to avoid relaying data via multihop paths, thereby reducing the energy consumption of the relaying SNs. The authors proposed a cluster-based algorithm that uses binary search to determine the RP count. An algorithm for solving traveling salesman problems (TSPs) [29] was also used to determine the sequence of nodes in the path along which the MSs travel to collect data. However, cluster-based algorithms are not deterministic when they select RPs: they might occasionally choose improperly positioned SNs to be RPs.

In 2013, Salarian *et al.* [30] introduced a weighted rendezvous planning (WRP) algorithm to address the problem. The algorithm selected SNs as RPs based on weights determined by the proposed heuristic process, resulting in the distribution of RPs throughout the area. A shortest-path-tree algorithm (SPT) [23], [31] was adopted to determine the optimal RP for relaying the data of each SN. An algorithm for solving the TSP was used to calculate a cycle for the MS to collect the data from all RPs in order. Although this

RP-selection method is efficient, SPT routing may result in excessive data transmission/receipt at specific nodes.

In 2016, Cheng and Yu [32] presented a method to reduce the path lengths given by TSP by allowing the MS to collect data once it reaches the SN's communication range. The authors suggested data collection in the area where the MSs could collect data from multiple RPs (where RPs' communication ranges overlap). The method involved determining the number of RPs whose communication ranges overlap and selecting data-collection locations in the overlapping areas. However, RPs might have to transmit data to an MS located farther away, thereby consuming more energy. This method yielded deterministic paths based solely on the RPs' locations, which could lead to inefficient paths.

In Nayak and Devulapalli [33], the base station is able to move to collect sensory data. FLS is applied to assign each SN's weight value, based on battery power, mobility and centrality. The weights are used to cluster SNs and select super cluster heads, aiming to efficiently distribute the load among the SNs, thereby increasing the network lifetime. Nevertheless, the proposed method did not address the path design problem for data collection.

In 2017, Kaswan *et al.* [34] introduced methods called reduced k-means (RkM) and delay bound reduced k-means (DBRkM) for route planning. The work focused on clustering RPs based on weights and reducing the number of nodes within the RPs' communication ranges to obtain a wider distribution of RPs. A TSP algorithm was also adopted to arrange data-collection locations for the MS to travel along in order. However, MSs were not fully utilized, as the work focused solely on SN management. Cheng *et al.* [35] presented a data-collection path planning method that depended on area partitioning and the density of SNs in each area that could increase the efficiency in terms of energy consumption. However, the work did not address the case where data might have to be relayed via multihop paths.

In 2018, Alsaafin *et al.* [26] presented three path-planning methods: reduced energy path (REP), reduced delay path (RDP), and delay bound path (DBP). REP was the most energy efficient; RDP yielded paths along which the MS took the least amount of time to travel; and DBP planned traveling paths for the MS that satisfied the DL constraint by choosing data-collection locations located between paths obtained from REP and RDP. This research allowed data collection once the MS traveled inside the RPs' communication range, thereby reducing the travel path distances. Wen *et al.* [36] proposed a non-TSP-based routing method called path construction to determine round-trip routes for the MS. The method first created a convex-polygon path formed by all the outer RPs and then adjusted the path by including all the remaining inner RPs.

Amgoth and Annavarapu [37] proposed a method based on the technique called ant colony optimization (ACO) for path design. The method also relies on a directed spanning tree representing the network relay of sensor data. For efficient path design, pheromones are used as weights for selecting

RPs. The method gradually selects one RP at a time until the whole path requires the MS to travel along more than the DL. However, the method depends on the tree and selects the next RP only from the neighbors of the current RP, and it also requires the MS to reach the node before gathering the data. Thus, in scenarios where the DL is limited, the path might cover only the nodes lying on certain connected branch of the trees. In contrast, in our technique, we consider the whole WSN topology and are able to select RPs that are distributed across the network. In scenarios where the DL is large, the ACO-based method does not consider increasing the network lifetime.

Our proposed method increases RPs as long as the path does not exceed the DL so that the path could be longer and closer to SNs, thus saving more network lifetime. Sert *et al.* [38] addressed the problem of data gathering in WSNs, and proposed a method called the two-tier distributed fuzzy logic-based protocol (TTDFP). The authors suggested the use of FLS, relative node connectivity, distance to the sink, and remaining energy for clustering and selecting the clusters' heads. Fuzzy rule parameters were adaptive (instead of being fixed) to avoid inefficient trial and error methods and any human bias. The FLS was also applied to routing based on the average link remaining energy and relative distances. However, the work required head nodes to relay sensory data to the base station. Once the head nodes are selected, their energy might be depleted rapidly, thus reducing the network lifetime.

In addition, Qadori *et al.* [39] proposed an MS-based data collection method called fuzzy-based mobile agent migration (FuMAM). The FLS-based method arranged the order of SNs, from which the MS has to travel to collect data by taking into account the remaining energy, the distance to the source node, and the number of neighbors. However, based solely on FLS (and not on TSP), it is possible that the designed path was not the shortest path that passes through all SNs. The work also did not select any SNs as RPs.

In 2019, Wang and Chen [24] introduced a technique for planning paths along which the MS could travel and collect data with the limitation of buffers. The technique, called EARTH, created a tree formed by all SNs and selected RPs based on the hop counts, tree height, and the amount of relay data. The method examined the suitability of SNs before promoting them as RPs. The author also attempted to reduce the amount of data required for transmission per RP. This research demonstrated a technique in which RP selections were revalidated by considering nearby SNs. As the focus of this work was on managing data traffic, the travel paths for MSs were significantly longer than others.

In 2020, Donta *et al.* [40] presented a method called hierarchical agglomerative clustering-based data collection (HACDC) to address the path design problem in 3-dimensional WSNs. The work is based on hierarchical agglomerative clustering to determine SN groups and on dendrogram statistical methods to determine RPs. The virtual RPs are determined to increase the efficiency and reduce the

RPs' load before designing the paths based on the method called MS tour planning. However, once the method specifies the RPs and the paths, it does not perform path fine-tuning and does not address the DL constraint.

In this paper, our research work selects RPs by leveraging a FLS to determine weights, in contrast to other studies that calculate weights based on equations. The FLS allows multiple membership functions (MFs). Each variable value is transformed by the MFs into multiple membership levels, which are used as inputs in the FLS's rules, thereby providing more flexible value interpretations. We take into account SN energy in the path determining process, while previous works consider the quantity of neighboring SNs and distances.

While previous works (except [32]) are based only on the TSP algorithm to determine the traveling path for the MS, we propose a method to reduce the path obtained via TSP to determine data-collection locations (within the RPs' communication ranges), allowing more RPs on the path under the same delay-time constraint and reducing the amount of data transmitted by each RP. When selecting the relaying RP for each SN, previous studies choose the RP closest to the SN and depend on MST for determining the relaying paths, while our work is based on each RP's weight (not just the distance). Thus, the loads can be distributed among all RPs with remaining energy, increasing the overall RP longevity.

III. SYSTEM MODELS

We provide a brief description of network and energy models in this section.

A. NETWORK MODELS

To simulate WSNs to study the performance of MSs' data collection, we define variables and model networks based on relevant research [26], [30], [34]. Each network consists of a single sink node that acts as a base station that receives data from a MS. SNs, with sufficient energy at appropriate locations are chosen to be RPs. The unselected SNs (non-RP SNs) send their data to specific RPs, which then relay the data to the MS. After the algorithm plans an RP route, the MS travels a round trip along the route to collect data from all the SNs.

In contrast to SNs with always-on power sources, all the SNs are sensitive to energy loss (while the MS is sensitive to travel time). Our method also allows an RP to transmit data once it reaches the MSs' communication range without reaching the RPs' exact locations. Certain non-RP SNs are also responsible for receiving data from nearby SNs far from any RPs before forwarding the data to the specific next hop (whether it is an RP or another non-RP SN).

Note that Table 1 presents all the notations used in this paper.

Figure 1 shows the network model, which consists of a sink node (shown as the black square) and SNs (shown as the circles). The black circles represent SNs selected as RPs, and the white circles represent the SNs not chosen as RPs

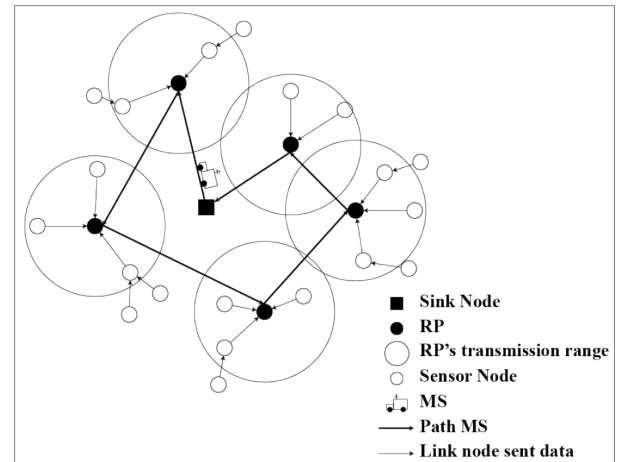


FIGURE 1. Data gathering in a WSN using a MS.

TABLE 1. Notations and their definitions.

Notation	Definition
DL	Delay limit
RP	Rendezvous point
R	Transmission range
E_t	Transmission energy
E_r	Receiving energy
E_{elec}	Energy used for 1 bit computation
E_{amp}	Energy required for the transmission of 1 bit
k	Data in bits
d	Distance
RE	Remaining energy
AEC	Average energy consumption
NNR	Number of nodes in transmission range
$D_{SN,C}$	Distance from SN to the centroid of nodes in the SN's communication range
CRP	Candidate RP
MF	Membership functions
$STPath$	Shortest traveling path for MS
$TxEPath$	Transmission-energy-efficient path for SN
$avgRP$	Average number of RPs on $STPath$ and on $TxEPath$
TSP	Traveling salesman problem
RBR	Reduce path length based on radius
$FinPath$	Final RP path
$CandidateNode_{i,j}$	The j^{th} SN in the communication range of SN_i
$CandidatePath_{i,j}$	Set of nodes in the candidate path from the $CandidateNode_{i,j}$ to its associated RP (the set includes $CandidateNode_{i,j}$ and the RP)
$AvgE_{i,j}$	Average energy of nodes in $CandidatePath_{i,j}$
$Dist2RP_{i,j}$	Distance of the path from SN_i to its j^{th} RP via $CandidatePath_{i,j}$
$NNG_{i,j}$	Number of nodes in $CandidatePath_{i,j}$
E_{init}	Initial energy

(i.e., non-RP SNs). Our model also makes the following assumptions [22], [26], [30], [34]:

- All SNs have the same initial energy level and operate under a limited power constraint.
- Every SN is always able to send data to the sink node via hop-by-hop transmission.
- After node deployment, all nodes are immobile and know their own exact coordinates and the location of the sink node (e.g., from GPS or various localization methods).
- The MS must travel to collect the information from every RP.
- The MS wastes no time collecting data from RPs.

- The MS has enough energy to travel a round trip to collect the data. The battery can be either swappable or a fast-charge battery based on current battery technology.
- If the MS travels to collect data within an RP’s communication range but does not truly reach the RP, the data transmission requires more power (due to the longer distance), as in the case when SNs send data to their associated RP.

B. ENERGY MODELS

The most popular fundamental means of modeling WSNs is to focus on SNs’ power consumption. According to [41], [42], the power consumption depends on the amount of transmitted data (k) and the distance (d) between the sender and the receiver.

$$E_t(k, d) = (E_{elec} \times k) + (E_{amp} \times k \times d^2) \quad (1)$$

$$E_r(k) = E_{elec} \times k \quad (2)$$

Equations (1) and (2) are used to calculate the power consumed by an SN while transmitting and receiving data, respectively. As presented in Table 7, E_{elec} and E_{amp} are set to 50 nJ/bit, and 10 pJ/bit/m², respectively, according to [34].

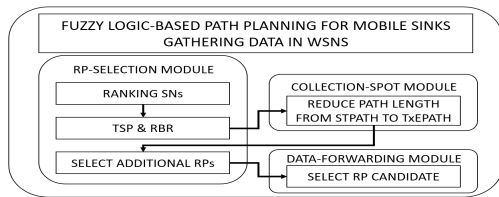


FIGURE 2. Modules in FLPP.

IV. FUZZY LOGIC-BASED PATH PLANNING (FLPP) FOR DATA GATHERING MSs IN WSNs

Figure 2 shows an overview of the proposed system, which consists of three modules described in the subsections RP-selection module, Collection-spot module, and Data-forwarding module.

- 1) **RP-selection module.** This module determines the appropriate number of RPs. Our proposed method starts from considering all SNs according to their weight. SNs with higher weight values are considered first. We also consider the number of RPs along two paths: the one close to all SNs (called $TxEPath$, to preserve SNs’ energy when transmitting data) and the other (called $STPath$, which is the shortest path for the MSs to collect data at the exact locations of all RPs). Both paths require the MS to travel without exceeding the DL .
- 2) **Collection-spot module.** This module pinpoints spots (within the RPs’ communication ranges) for the MS to collect data. It considers the $TxEPath$ and $STPath$ to determine a shorter MS traveling path. The FLS weights help in determining how close to the RPs the data-collection locations should be.

- 3) **Data-forwarding module.** This module specifies which neighboring node of each SN should be designated the next hop to relay the SN’s data. The designated node can be chosen from 1) the RPs within the SN’s communication range and 2) other non-RP SNs that are also within the SN’s communication range and already have their own designated node. We use FLS to select a designated node based on its remaining power, the number of SNs having the node as their designated node, and its distance from the SN.

The results from the FLPP modules can be improved by redesigning the paths (as discussed in detail in Section IV-C.2). In Section VI, we compare previous works with our proposed method without path redesign (i.e., the paths are determined only once). After the comparisons, we present the experimental results of our enhanced FLPP.

A. RP-SELECTION MODULE

This module adjusts the MS traveling path so that the MS takes a shorter or equal amount of time.

1) **RANKING SNS (FOR DETERMINING CANDIDATE RPS)**

This step ranks SNs in terms of their ability to be RPs. FLS is adopted to determine each SN’s weight based on the following characteristics.

- **Remaining energy (RE).** The SNs that are effective RPs are the ones with high remaining power levels to support data transmission.
- **Number of nodes in the transmission Range (NNR).** Effective RPs are in an area of dense SNs and could have a large number of associated SNs, thereby reducing the overall number of data relays. However, an excessive number of associated SNs could overload the RPs: a suitable area is not overcrowded nor too sparse.
- **Average energy consumption (AEC).** AEC is the average energy consumed by each node per the MS’s traveling trip. If an RP consumes a relatively large amount of power, the RP will run out of power before other SNs. Thus, rotation of SNs as RPs, (based on their AEC) is necessary.
- **Distance from the SN to the centroid of all SNs in its communication range ($D_{SN,C}$).** SNs closer to their associated centroid (thus being closer to the SNs within their communication range) should be chosen as RPs. All the associated SNs would consume a similar amount of power when transmitting data and thus run out of energy almost simultaneously.

After the weights are determined, we rank the SNs as illustrated in Figure 3. Figure 3(a) shows the weight of each node. Figures 3(b-e) and 3(f-h) show the node ranking in the first and second iteration, respectively. In Figure 3(b), SN_5 is the first node to be considered as an RP as its weight is highest. SN_6 is in SN_5 ’s communication range and is not ranked nor selected until the next iteration. (To differentiate

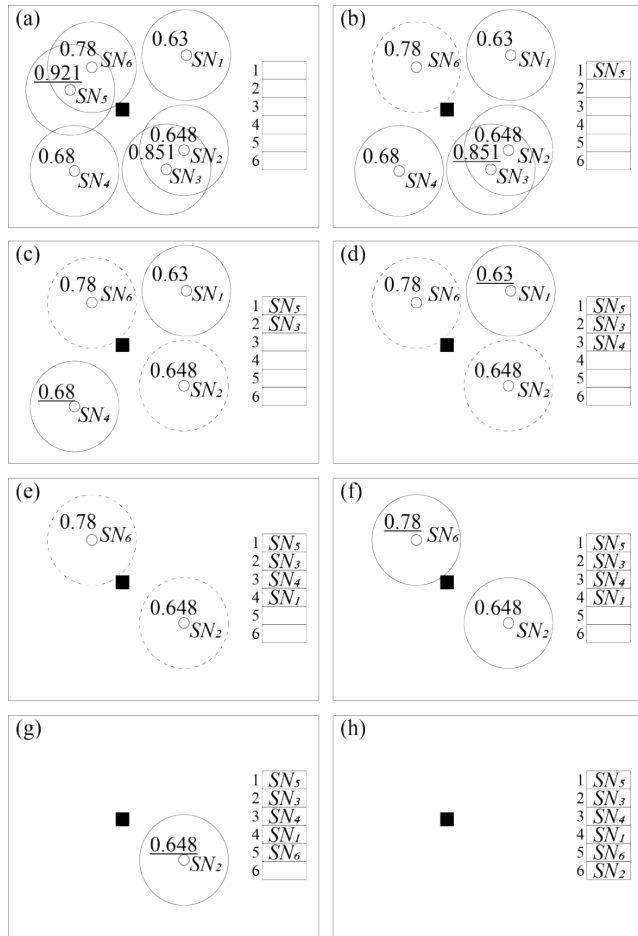


FIGURE 3. Example of RP ranking.

nodes ranked in the current and the next iteration, the former's communication range is represented by a solid circle while the latter's is represented by a dash circle.) The process continues ranking the next maximal-weight SN and excluding the nodes in its radius (for the next iteration) until no SNs are available for ranking, as shown in Figure 3(e). Then, the process is repeated for the second iteration, as shown in Figure 3 (f-h).

To rank the potential of SNs to be RPs, Algorithm 1 requires the weight of each SN, which is obtained via FLS.

Lines 1-5 initialize the input variables. *listSNs* is the variable used to store the lists of SNs. *listRE* stores the remaining energy of all SNs. *listAEC* stores the average energy consumed by each node per the MS's traveling trip (average over all previous trips); it initially has a value of 0. *list_of_ListNinR* is a variable that stores a list of SNs within the communication range of each SN. *listDSN,C* is used to store the distance from each SN to its centroid. *listCRP* stores the sequence of SNs to be considered as RPs obtained by considering the FLS-based weights according to lines 7-9. In lines 10-11, the highest-weight SN is determined. The list of SNs within range of the highest-weighted SN are stored in *listNinR* (line 11). The SN is removed from *listSNs* (line 13) once

Algorithm 1 Select Candidate RPs

INPUT: *SNs*

OUTPUT: Candidate RPs (*listCRP*)

- 1: *listSNs* = *SNs*
- 2: *listRE* = the energies of each SN
- 3: *listAEC* = the list of the average energy consumption of each SN (per the MS's traveling trip)
- 4: *list_of_ListNinR* = the list of the lists of nodes within the communication range of each SN
- 5: *listDSN,C* = the distances from each SN to the centroid point of the nodes within its area
- 6: FOR *i* = 1 to size(*SNs*)
- 7: $W1 = \text{CALL:FLS}(\text{listRE}, \text{list_of_ListNinR})$
- 8: $W2 = \text{CALL:FLS}(\text{listAEC}, \text{listDSN,C})$
- 9: $W3 = \text{CALL:FLS}(W1, W2)$
- 10: $\text{max_i} = \text{max_index}(W3)$
- 11: $\text{listNinR} = \text{list_of_ListNinR}[\text{max_i}]$
- 12: ADD *listSNs*[*max_i*] in *listCRP*
- 13: DELETE *listSNs*[*max_i*]
- 14: DELETE *list_of_ListNinR* [*max_i*]
- 15: FOR EACH node in *listNinR*
- 16: $\text{tmp_index} = \text{listSNs.index}(\text{node})$
- 17: DELETE *list_of_ListNinR* [*tmp_index*]
- 18: DELETE *listSNs* [*tmp_index*]
- 19: END FOR
- 20: IF (*listSNs* == \emptyset & *i* < size(*SNs*))
- 21: *listSNs* = SNs without nodes in *listCRP*
- 22: *list_of_ListNinR* = the list of the lists of the nodes in the range of each SN in *listSNs*
- 23: END IF
- 24: END FOR

it is inserted into *listCRP* (line 12). In lines 15-19, the SNs in *listNinR* are removed from *list_of_ListNinR* and *listSNs*. Lines 19-22 reinitialize *list_of_ListNinR* and *listSNs* for the SNs that have not been ranked.

Lines 7-9 show the weight calculation based on the dual FLS method. This method takes into account two factors at a time (i.e., considering *listRE* with *listNinR*, and *listAEC* with *listDSN,C*), producing two weights *W1* and *W2*. The final weight (*W3*) is determined by considering *W1* and *W2*. The determination of the weights starts with normalizing the *RE* and *NINR* values into the range [0,1] and using these values to construct the MFs.

In this paper, we choose triangular MFs, as shown in Figure 4, as they can represent values whose degrees of membership (membership levels) are different. We adjust the MF model based on the results of preliminary experiments and then define the conditions of the ally fuzzy rules. The number of rules depends on the number of inputs and the input MF stage. For example, the *W1* weighting requires two inputs: *RE* and *NINR*. Each input consists of three MF stages (low: L, medium: M, and high: H). Thus, there are 9 possible fuzzy rules (3²), which output 5 MF stages (i.e., very low: VL, low: L, medium: M, high: H and very high: VH),

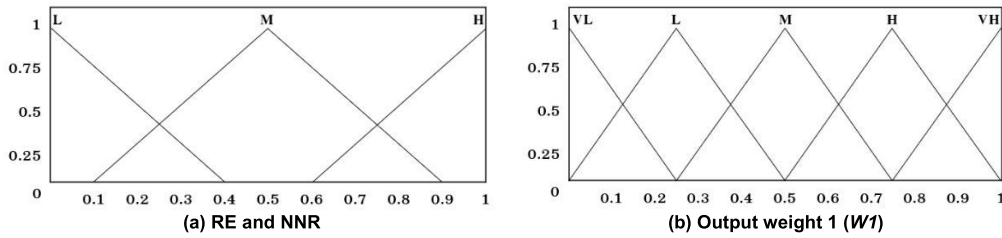


FIGURE 4. MFs of RE, NNR and W1.

TABLE 2. Fuzzy rules to determine W1.

Rule No.	IF (RE)	AND (NNR)	THEN (W1)
1	H	H	M
2	H	M	VH
3	H	L	M
4	M	H	L
5	M	M	H
6	M	L	L
7	L	H	VL
8	L	M	M
9	L	L	VL

as shown in Table 2. According to the rules, if an SN has more remaining energy (RE), it is assigned a greater W1 value, while the SN’s NNR should be at a moderate level.

The proposed fuzzy inference engine (based on an aggregation method called intersection operation) is used to obtain W1 as an output. Figure 5 shows an example of using the fuzzy inference engine according to the 9 rules in Table 2. We assume RE and NNR are 0.8 and 0.6, respectively. In each rule, the membership degrees of RE and NNR are compared: the smaller one is used as the output’s membership degree. The outputs of all rules are then aggregated, and defuzzified by determining the center of gravity (CoG) of the aggregated output, yielding the output MF in the first stage of FLS (W1).

In short, the output is obtained by taking the normalized RE and NNR values, which are then transformed by MFs into membership degrees. The degrees are then used for inference by means of each fuzzy rule. The inferred results are weighted and finally aggregated. In other words, the output (W1) is the sum of all membership degrees obtained by applying the MF to the normalized RE and NNR values (according to all the rules) based on the aggregation method.

After obtaining W1, we determine W2 by applying FLS to the values of AEC and D_{SN,C}. This step is similar to the previous one. The MF for AEC and D_{SN,C} is shown in Figure 4(a), while that for W2 is shown in Figure 4(b). Note that if AEC (average consumed power) and D_{SN,C} (distance between the SN and its centroid) are large, then the SN’s potential to be an RP decreases. Table 3 shows all 9 fuzzy rules, whose outputs are aggregated later to determine W2.

To calculate the W3 of each SN, both W1 and W2 are used by FLS according to the fuzzy rules in Table 4. The MF for W1 and W2 is shown in Figure 4(a), while that for W3 is shown in Figure 4(b). The SNs with the largest W3 are stored

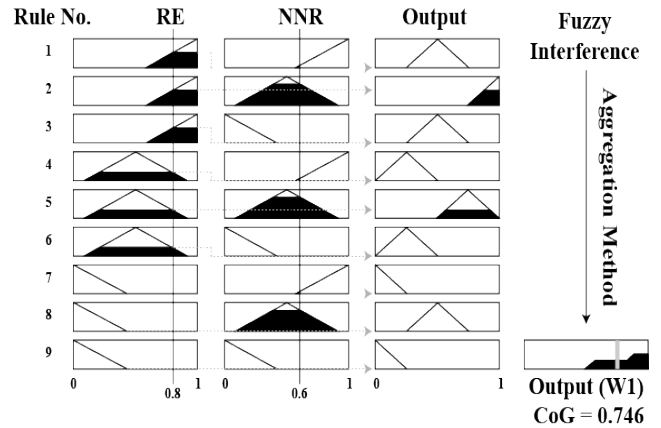


FIGURE 5. Example of fuzzy inference.

TABLE 3. Fuzzy rules to determine W2.

Rule No.	IF (AEC)	AND (D _{SN,C})	THEN (W2)
1	H	H	VL
2	H	M	L
3	H	L	M
4	M	H	L
5	M	M	M
6	M	L	H
7	L	H	M
8	L	M	H
9	L	L	VH

in CRP. The SN and the other SNs within its communication range are excluded from the weight calculation in the current iteration to avoid selecting RPs that are close to each other. Then, the SN with the second highest W3 is determined. The process is repeated until there are no more SNs to consider. At the end of each iteration, if the number of SNs in the CRP is still not equal to the number of all SNs, the process starts the next iteration by reconsidering the SNs (excluded in the current iteration but not yet in the CRP) to be selected as RPs.

2) TSP TO REDUCE PATH LENGTH BASED ON RADIUS (RBR)

After the CRP is obtained, we focus on determining the appropriate number of RPs (avgRP) to be deployed on the path the MS travels to collect sensor data. However, there are two other outcomes to address: TxEPath (the path that conserves the RPs’ transmission energy) and STPath (the path that conserves the MS’s energy).

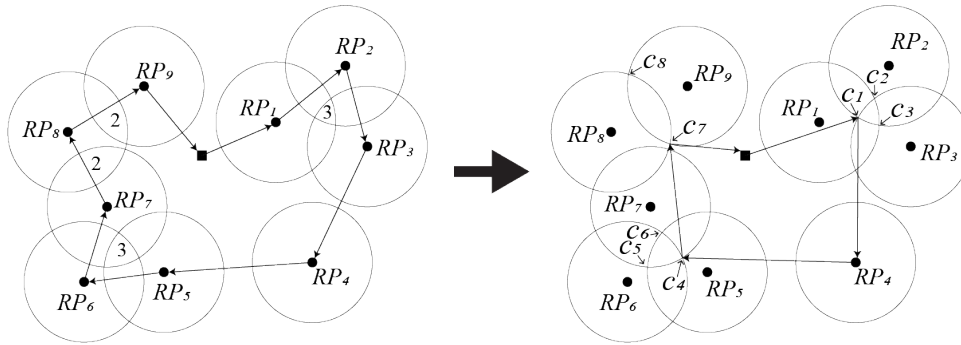


FIGURE 6. Path reduction when RPs' communication ranges overlap (from TSP to RBR (part 1)).

TABLE 4. Fuzzy rules to determine W3.

Rule No.	IF (W1)	AND (W2)	THEN (W3)
1	H	H	VH
2	H	M	H
3	H	L	M
4	M	H	H
5	M	M	M
6	M	L	L
7	L	H	M
8	L	M	L
9	L	L	VL

The TSP to RBR procedure has two phases. Phase I (TSP): Determine $TxEPath$ based on TSP [29] to provide cycles visiting all given RPs. Phase II (RBR): Determine $STPath$, which is the path obtained by applying our distance-shortening process, called RBR (reduce path length based on radius), to the $TxEPath$. RBR attempts to reduce the MS's traveling distance by redirecting the MS path through the area where the RP's communication ranges overlap because the MS only has to reach the RPs' communication ranges, not the RPs' locations, to collect data. This phase is described in detail below.

1) **Path reduction when RPs' communication ranges overlap.** This step starts by determining the overlapping areas through which the $TxEPath$ passes. We then rank the areas based on their number of overlapping communication ranges ($Noverlap(i)$ is the number of overlapping ranges of the i^{th} area, for $1 \leq i \leq m$, where m is the number of overlapping areas). The area with the largest $Noverlap$ is the first to be considered for path reduction.

Note that to determine $Noverlap(i)$, we count only the overlapping communication ranges of the RPs in the traveling sequence along $TxEPath$. For example, let the traveling sequence of RPs along $TxEPath$ be RP_1, RP_2, RP_3 , and RP_4 . Although the communication ranges of RP_1, RP_2 , and RP_4 overlap, we count only the ranges in the sequence (i.e., those of R_1 and R_2); thus, $Noverlap(1) = 2$. Data collection in the areas where RPs' communication ranges overlap allows the MSs to collect data from multiple RPs at the same location.

A data-collection location is selected from the intersections of the circumferences of the RPs' overlapping communication ranges.

Specifically, the intersections that minimize the MS traveling path are chosen. As shown in Figure 6, the MS initially starts at the rectangular point. Let $(RP_i, RP_j, \dots, RP_k)$ be the area where RP_i, RP_j, \dots, RP_k overlap. There are 4 overlapping areas: $(RP_1, RP_2, \text{ and } RP_3)$, $(RP_5, RP_6, \text{ and } RP_7)$, $(RP_7 \text{ and } RP_8)$, and $(RP_8 \text{ and } RP_9)$, and $Noverlap(1), Noverlap(2), Noverlap(3)$, and $Noverlap(4)$ are 3, 3, 2, 2, respectively. $(RP_1, RP_2, \text{ and } RP_3)$ and $(RP_5, RP_6, \text{ and } RP_7)$ are chosen first as their numbers of overlapping ranges are the greatest. Next, $(RP_8 \text{ and } RP_9)$ is chosen.

Note that RP_7 is in both $(RP_5, RP_6, \text{ and } RP_7)$ and $(RP_7 \text{ and } RP_8)$; therefore, $(RP_7 \text{ and } RP_8)$ is not chosen because $(RP_5, RP_6, \text{ and } RP_7)$ is already selected. The intersections of the first area are c_1, c_2 , and c_3 ; the second area has intersections at c_4, c_5 , and c_6 . The intersections of (RP_8, RP_9) are c_7 and c_8 . As shown in Figure 6, c_1, c_4 , and c_7 reduce the path distance the most.

2) **Path reduction based on shortcuts.** In this step, we consider reducing the path length when the MS has to travel to collect data from the remaining RPs whose communication range does not overlap with others. Instead of visiting the exact RPs' locations to collect their data, the MS collects data when it is within the RPs' communication range; see Figure 6 as an example. The MS does not have to travel from c_1 to RP_4 and from RP_4 to c_4 because it just needs to reach the boundary of RP_4 's communication range to collect RP_4 's data.

To reduce the path length based on shortcuts to collect data from RP_n , we first consider the dashed line from RP_{n-1} to RP_{n+1} . If the line passes through RP_n 's communication range, the line is included in the MS traveling path. (The data-collection location lies on the straight line from RP_{n-1} to RP_{n+1} and closest to RP_n , shown as a cross in Figure 7a.) Otherwise, the new data-collection location is the intersection between the boundary and the line parallel to the direct line from RP_{n-1} to RP_{n+1} , as shown in Figure 7b.

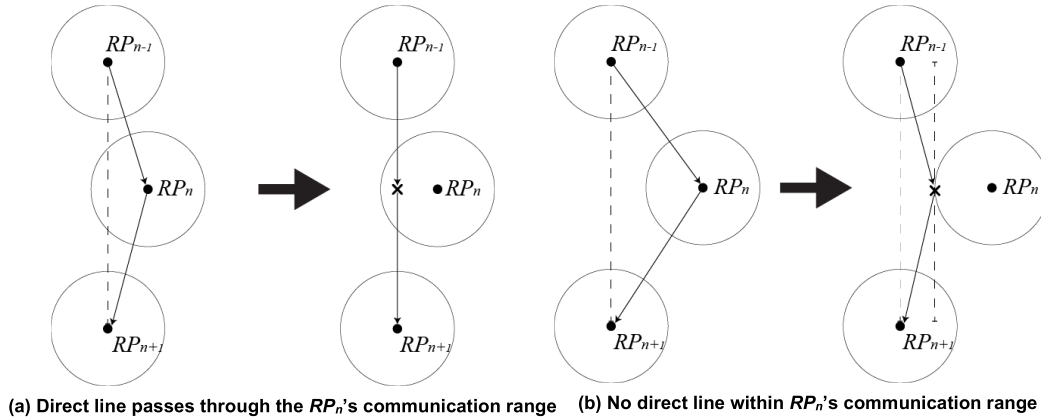


FIGURE 7. Path reduction based on shortcuts.

The two steps in RBR are represented by `reduce()` in line 7 of Algorithm 3 and Equation (5). Algorithm 2 and 3 illustrate the detailed processes for determining $TxEPath$ and $STPath$, respectively.

The greater the number of RPs on the $TxEPath$ is, the less energy the SNs consume during data transmission. Algorithm 2 determines the upper bound of the number of RPs. The binary-search algorithm is used to determine the maximum number of RPs (on the $TxEPath$) possible while the round-trip traveling time is still less than DL .

Algorithm 2 differs from Algorithm 3 in that the former applies TSP to determine circles (which are the MS traveling paths) to specify the visitation sequence of the RPs, while the latter is based on RBR to yield paths with reduced length. Algorithm 3 also provides the maximum number of SNs possible under the DL constraint.

After $TxEPath$ and $STPath$ (with the highest number of RPs possible) are obtained such that the round-trip MS's traveling time does not exceed DL , $avgRP$ can be obtained according to Equation (3). The average value is the number of RPs, which can be increased or decreased to adjust the MS traveling path. $TxEPath$ is determined using $avgRP$ according to Equation (4). $STPath$ is obtained by applying RBR to $TxEPath$, as shown in Equation (5). For our preliminary results, the number of RPs, equal to the average of $maxRP_{Tx}$ and $maxRP_{ST}$, yields best results. The number of RPs close to $maxRP_{ST}$ is insufficient: RPs waste energy transmitting data to the MS via longer distances. The number of RPs close to $maxRP_{Tx}$ is excessive, as it results in premature death of the RPs receiving data from the large number of SNs.

$$avgRP = \text{round} \left(\frac{maxRP_{Tx} + maxRP_{ST}}{2} \right) \quad (3)$$

$$TxEPath = TSP(sink_Node, RP_1, RP_2, \dots, RP_{avgRP}) \quad (4)$$

$$STPath = \text{reduce}(TxEPath) \quad (5)$$

3) SELECTION OF ADDITIONAL RPS

This is the last step of Module 1 designed to help in adding additional RPs selected from SNs within the MS's

Algorithm 2 Find $TxEPath$, $maxRP_{Tx}$ While Ensuring the Path'S Distance Is Less than DL

INPUT: CRP // the list of RP nodes sorted (in descending order) according to the FLS weights
OUTPUT: $TxEPath$, $maxRP_{Tx}$ // $TxEPath$: the path for conserving sensors' transmission energy $maxRP_{Tx}$: max number of RPs on the $TxEPath$ whose length is less than DL

```

1: low = 0
2: high = size(CRP)
3: mid = (low + high)/2
4: ADD Sink_Node, RP1, RP2, ..., RPmid to TxEPath
5: WHILE (low < high)
6:     IF (Dist(TSP(TxEPath)) ≤ DL)
7:         low = mid
8:         mid = (low + high)/2
9:         CLEAR TxEPath
10:        ADD Sink_Node, RP1, RP2, ..., RPmid to
        TxEPath
11:        ELSE IF
12:            high = mid
13:            mid = (low + high)/2
14:        CLEAR TxEPath
15:        ADD Sink_Node, RP1, RP2, ..., RPmid to
        TxEPath
16:        END IF
17:    END WHILE
18:    IF (Dist(TSP(TxEPath)) > DL)
19:        CLEAR TxEPath
20:        ADD Sink_Node and RP closest to the sink node to
        TxEPath
21:    END IF
22:    maxRPTx = size(TxEPath) - 1

```

communication range while it travels along the path to collect data. As shown in Figure 2, this process starts after $FinPath$ (i.e., Final RP Path) is obtained from Module 2, as described Section in IV-B. The selection of an SN as an RP is based on two conditions:

- $FinPath$ passes by the SN (i.e., the MS is within the SN's communication range while it travels along the path).

Algorithm 3 Find $STPath$, $maxRP_ST$ When the Path's Distance Is Less than DL

INPUT: CRP , $TxEPath$

OUTPUT: $STPath$, $maxRP_ST$ // $STPath$: the path for decreasing the MS's traveling time $maxRP_ST$: max number of RPs on the $STPath$ whose length is less than DL

- 1: $low = size(TxEPath)$
- 2: $high = size(CRP)$
- 3: $mid = (low + high)/2$
- 4: **ADD** $Sink_Node$, RP_1 , RP_2 , ..., RP_{mid} to $STPath$
- 5: **WHILE** ($low \leq high$)
- 6: $STPpath = TSP(STPpath)$
- 7: $reduced = reduce(STPpath)$ // length of $STPpath$ is reduced based on radius according to 4.1.2
- 8: **IF** ($dist(reduced) \leq DL$)
- 9: $low = mid$
- 10: $mid = (low + high)/2$
- 11: **CLEAR** $STPath$
- 12: **ADD** $Sink_Node$, RP_1 , RP_2 , ..., RP_{mid} to $STPath$
- 13: **ELSE IF**
- 14: $high = mid$
- 15: $mid = (low + high)/2$
- 16: **CLEAR** $STPath$
- 17: **ADD** $Sink_Node$, RP_1 , RP_2 , ..., RP_{mid} to $STPath$
- 18: **END IF**
- 19: **END WHILE**
- 20: **IF** ($dist(reduced) > DL$)
- 21: **CLEAR** $STPath$
- 22: **ADD** $Sink_Node$ and RP closest to the sink node to $STPath$
- 23: **END IF**
- 24: $maxRP_ST = size(STPath) - 1$

- The distance from the SN to $FinPath$ must be less than that from the SN to its closest RP.

Figure 8 shows an example of including more RPs after $FinPath$ has been obtained. $FinPath$ is shown as the dashed line. The distances from the candidate SNs (SN_5 and SN_6 in the figure) to their closest RP are the lengths of the bold lines. The shortest distances from the candidate SNs to $FinPath$ (within its communication range) are equal to the thin lines' lengths. SN_1 , SN_2 , SN_4 , and SN_7 are not candidates for RPs as they are farther from $FinPath$ than their closest RP. SN_3 has no $FinPath$ in its communication range. As being closer to the $FinPath$ than their closest RPs, SN_5 and SN_6 are chosen as additional RPs.

B. COLLECTION-SPOT MODULE (REDUCE PATH LENGTH FROM $STPath$ TO $TxEPath$)

After the previous module, we obtain $STPath$ (reduced by means of the proposed RBR method). In this section, we attempt to form $FinPath$ based on the locations where the MS collects data from each RP. The locations are determined

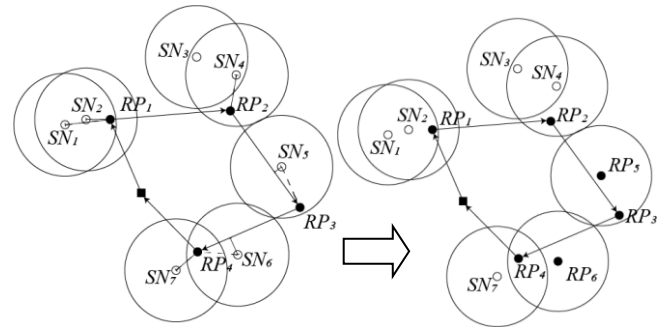


FIGURE 8. Considering additional RPs.

based on the RP's SN density and remaining energy with the aim of balancing the load among RPs with different SN densities.

If RPs are heavily loaded, the MS should approach closer to the RPs before collecting their data so that the RPs' energy lasts longer. In contrast, if the RPs' load is light, we can adjust the MS traveling path (i.e., the RPs transmit data at longer distances) to reduce the MS traveling time. The data-collecting positions depend on the weights, each of which is based on each the RP's energy trend. The weight values are calculated from a fuzzy system and are converted into percentages for adjusting $STPath$ towards $TxEPath$ and finally attaining $FinPath$ (which is still under the DL constraint), as shown in Figure 9.

To determine the RPs' energy-trend weights ($W4$), FLS is used in a similar way as when determining the RP-candidate weights, as previously described in the Section RP-selection module. Two factors are considered: RP's remaining energy (RE) and the number of nodes in the transmission range (NNR), based on Algorithm 4. The MF of RE and NNR is depicted in Figure 4(a), while that of $W4$ is shown in Figure 4(b). The FLS rules are shown in Table 5.

Lines 1-2 in Algorithm 4 initialize the variables. $W4$ in Line 3 represents the energy-trend weight obtained from FLS based on the values of RE and NNR . Line 5 shows $ratioRP$, which is the greatest possible distance percentage the data-collection locations can be adjusted away from each RP's position (in each iteration). Lines 6-14 show how $FinPath$ is determined by gradually adjusting each RP's data-collection location in $TxEPath$ towards the one in $STPath$ until the new path has a round-trip traveling time less than DL , as illustrated in Figure 9.

C. DATA-FORWARDING MODULE

1) SELECT RP CANDIDATES

After RPs are obtained, this step determines the designated nodes for each SN. The designated nodes can be selected from either RPs or SNs, which are in charge of relaying sensor data further to an RP. To choose a designated node for each SN, we consider not only the distance between the SN and RPs but also its remaining energy. The selection starts by considering the designated node for the SN closest to one of the RPs.

TABLE 5. Fuzzy rules to determine $W4$.

Rule No.	IF (RE)	AND (NNR)	THEN ($W4$)
1	H	H	M
2	H	M	H
3	H	L	VH
4	M	H	L
5	M	M	M
6	M	L	H
7	L	H	VL
8	L	M	L
9	L	L	M

Algorithm 4 Determine Final RP Path

INPUT: $STPath, TxEPath$
OUTPUT: $FinPath$

- 1: RE = list of remaining energy of all RPs
- 2: NNR = list of the numbers of SNs in communication range of each RP
- 3: $W4$ = CALL:FLS(RE, NNR) // list of energy-trend weights of all RPs
- 4: $FinPath = TxEPath$
- 5: $ratioRP = W4/\max(W4)$
- 6: **FOR** $i = 0.1; i \leq 100; i+ = 0.1$
- 7: **IF**(Distance($FinPath$) $\leq DL$)
- 8: **BREAK**
- 9: **END IF**
- 10: **FOR EACH** RP in $TxEPath$
- 11: Move data-collection location from the RP's original location in $TxEPath$ toward $STPath$ by $ratioRP$
- 12: **ADD** the new location to $FinPath$
- 13: **END FOR**
- 14: **END FOR**

Then, the process assigns one designated node to the SN that is second closest to one of the RPs. The process continues on until all SNs have a designated node.

To determine a designated node appropriate for relaying data for SN_i , we consider all candidate nodes (SN_j) within the SN_i 's communication range. Note that a candidate SN_j can be either one of the RPs or another SN that is a designated node already assigned to other SNs. In addition, it is necessary to consider the remaining energy of all nodes lying in the j^{th} candidate path from SN_i via SN_j to the corresponding RP (RP_j), denoted as $candidatePath_{i,j}$.

The node closest to SN_j is not always the SN_i 's designated node. As shown in Figure 10, FLS is adopted to determine the weights of all candidate nodes in (SN_j) based on the following factors:

- The average energy of all the nodes in $candidatePath_{i,j}$ ($AvgE_{i,j}$). (If $AvgE_{i,j}$ is already low, assigning SN_j to SN_i might cause the energy of the nodes in the j^{th} path to be depleted faster.)
- The distance from SN_i to RP_j via the candidate node SN_j ($Dist2RP_{i,j}$).
- The number of nodes lying in $candidatePath_{i,j}$ ($NNG_{i,j}$).

TABLE 6. Fuzzy rules to determine $W5$.

Rule No.	IF ($AvgE$)	AND ($Dist2RP$)	AND (NNG)	THEN ($W5$)
1	H	H	H	LL
2	H	H	M	M
3	H	H	L	LH
4	H	M	H	M
5	H	M	M	LH
6	H	M	L	H
7	H	L	H	LH
8	H	L	M	H
9	H	L	L	VH
10	M	H	H	L
11	M	H	M	LL
12	M	H	L	M
13	M	M	H	LL
14	M	M	M	M
15	M	M	L	LH
16	M	L	H	M
17	M	L	M	LH
18	M	L	L	H
19	L	H	H	VL
20	L	H	M	L
21	L	H	L	LL
22	L	M	H	L
23	L	M	M	LL
24	L	M	L	M
25	L	L	H	LL
26	L	L	M	M
27	L	L	L	LH

Table 6 shows all 27 new rules, each of which yields a $W5$ value as output based on 7 membership levels: very low (VL), low (L), little low (LL), medium (M), little high (LH), high (H), and very high (VH).

Figure 11 shows an example of selecting designated nodes. A designated node is selected for SN_1 first since it is closest to one of the RPs. Figure 11(a) shows the weights of the candidate nodes (RP_1 and RP_2 with the arrow dashed line) obtained from the FLS. Figure 11(b) shows that the designated node of SN_1 is RP_1 (shown with the arrow solid line) since its weight is higher than that of RP_2 . Figure 11(c) shows the designated-node selection for SN_3 (SN_3 is the second closest SN to one of the RPs). SN_3 's candidate nodes are SN_1 and RP_2 . Although SN_1 is closer to SN_3 , the path from SN_3 to RP_2 is shorter than that to RP_1 via SN_1 : SN_1 's weight is therefore less than that of RP_2 . Figure 11(e) shows the designated-node selection for SN_2 , where SN_2 can choose to send data via 3 candidate paths: the direct path from SN_2 to RP_1 , $candidatePath_{3,1}$ to RP_1 via SN_1 , and $candidatePath_{3,2}$ to RP_2 via SN_3 . Because the weight of SN_1 is the highest, it is assigned as the designated node for SN_2 .

2) ENERGY-AWARE DYNAMIC PATHS

Each node's remaining energy must be monitored as its power might run out prematurely. We propose a metric (T_{DP}) by

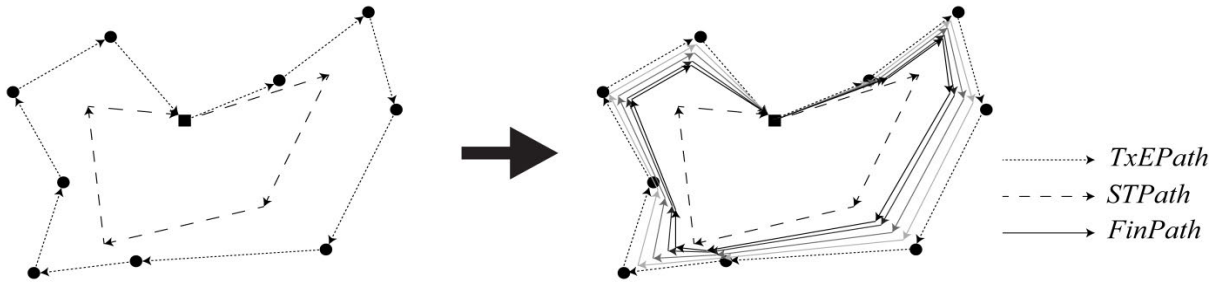


FIGURE 9. STPath is adjusted toward TxEPATH to attain a new route, FinPath.

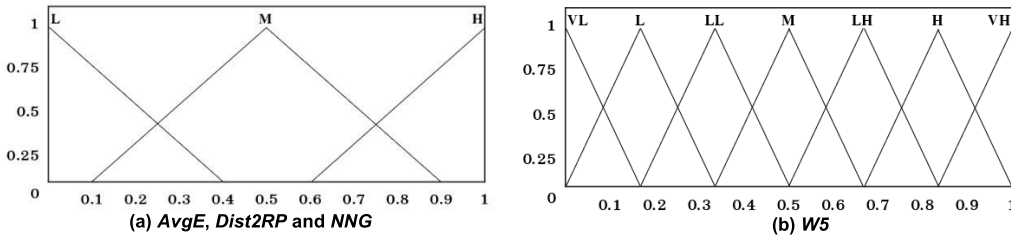


FIGURE 10. MFs of $AvgE_{i,j}$, $Dist2RP_{i,j}$, $NNG_{i,j}$, and $W5$.

comparing 1) the average remaining energy (i.e., the difference between the initial energy (E_{init}) and the average expended energy (AEC)) with 2) the least remaining energy ($\min(RE)$) among all nodes in the network, according to Equation (6).

Based on our experiments, path redesign (rerouting) should be performed when T_{DP} is greater than a threshold, indicating that the gap between the average remaining energy and $\min(RE)$ might be excessive. If the threshold is less than threshold, path redesign would be triggered more often, but there might be very little or no change in the new path. If the threshold is set more, the resulting path may significantly change from the original. However, the larger the threshold is, the greater the number of nodes that run out of power prematurely. Therefore, T_{DP} is calculated after each round of MS data collection. If T_{DP} is less than threshold, the MS travels along the same path in the next round; otherwise, the MS roams along a newly designed path.

$$T_{DP} = \left(\frac{E_{init} - AEC - \min(RE)}{E_{init}} \right) \times 100 \quad (6)$$

V. RUNNING TIME ANALYSIS & SPACE COMPLEXITY

In this section, we provide the running time analysis and space complexity of our proposed method consisting of 3 major modules: RP selection, data-collection-location determination, and designated-node selection.

A. CALCULATING THE NUMBER OF RPs AND DESIGNING THE MS TRAVELING PATH

Given that there are N SNs deployed throughout a WSN area, the process of selecting SNs as RPs consumes the most running time, $O(3N+2N^2) = O(N^2)$, consisting of the running times of the following subprocesses: obtaining

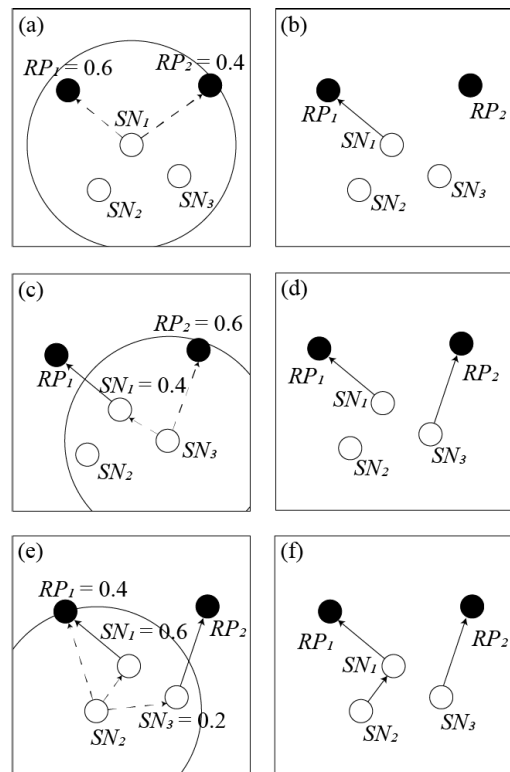


FIGURE 11. Example of selecting designated nodes to relay data for SN_i .

each SN's energy ($O(N)$) and each SN's already consumed energy, determining nodes in each SN's communication range ($O(N^2)$), calculating the distances from the mean position center to SN ($O(N)$), and the FLS calculation for ranking RPs in each traveling round ($O(N^2)$) [43].

For the MS traveling path design, TSP and RBR are used to determine the number of RPs (M) forming a path that the MS

would travel through without exceeding the DL . Specifically, the computation time for the TSP is equal to $O(M^3)$. The binary search technique reduces the computation time for determining the TSP path to $O(\log M)$. For the RBR process, the computational time for an RP to determine its neighboring RPs is $O(M^2)$. In total, TSP and RBR require $O(M^3 \log M + (M^2 + M^3) \log M) = O(M^3 \log M)$. As illustrated in the section on selecting additional RPs, $O(N)$ is required in the worst case.

B. DETERMINING THE MS'S DATA-COLLECTION LOCATIONS

To adjust the MS traveling path so that the MS does not exceed the DL , the running time is $O(M)$ at the maximum due to the usage of FLS ($O(M)$) and the 100 iterations ($O(100)$).

C. DETERMINING THE NODE TO WHICH EACH SN SENDS DATA

The worst-case running time (in the scenario that all SNs are located within each other's communication range) for choosing an RP to which each non-RP SN sends data is $O(N^2)$. In practice, the running time could be $O(N \times (N-1))$ if there is only one RP to which all the rest send data. For all the processes, our method requires $O(N^2 + M3 \log M)$ running time, while CB and WRP require $O(N^2 \log N)$ and $O(N^3)$, respectively. A comparative analysis of the running time is presented in Section VII.

D. SPACE COMPLEXITY

In regard to Algorithm 1, 10 lists and 4 integer variables require $40N + 16$ bytes. Algorithm 2 involves 2 lists and 6 integer variables, demanding an additional $8N + 24$ bytes. Algorithm 3 works similar to Algorithm 2 but requires one more list, so $12N + 24$ bytes are needed. In Algorithm 4, data are stored in 6 lists and 3 integer variables (i.e., $24N + 12$ bytes). In total, the space complexity of our method is $84N + 76$ bytes ($O(N)$). However, all the algorithms run consecutively (from Algorithm 1 to 4) and the unused spaces should be returned. Therefore, our method requires only $40N + 16$ bytes at most.

VI. EXPERIMENTAL RESULTS

In this section, we provide the experimental settings and results.

A. EXPERIMENTAL SETTINGS

We evaluate the performance of our proposed system via simulations using MATLAB R2018b. The proposed algorithm is evaluated in three different network scenarios (i.e., areas of $220 \text{ m} \times 220 \text{ m}$ [34], $500 \text{ m} \times 500 \text{ m}$ [26] and $1000 \text{ m} \times 1000 \text{ m}$), where SNs are deployed (randomly with a uniform distribution) to demonstrate the efficiency of our proposed algorithm and determine whether it is flexible for both small and large areas. The sink node is always at (0,0) in the lower left position, as in [26]. All SNs have the same initial power (2 J) [34] and have 128 bytes of packet data to send [26].

TABLE 7. Simulation parameters.

Parameter	Value [26, 33]
Area	$220 \text{ m} \times 220 \text{ m}$, $500 \text{ m} \times 500 \text{ m}$, $1000 \text{ m} \times 1000 \text{ m}$
Number of SNs	50–1000
E_{init}	2 J
Range	30 m, 50 m
E_{elec}	50 nJ/bit
E_{amp}	10 pJ/bit/m ²
Packet size	128 bytes [26]
Speed	2 m/s
DL	275 s, 1200 s, 2400 s

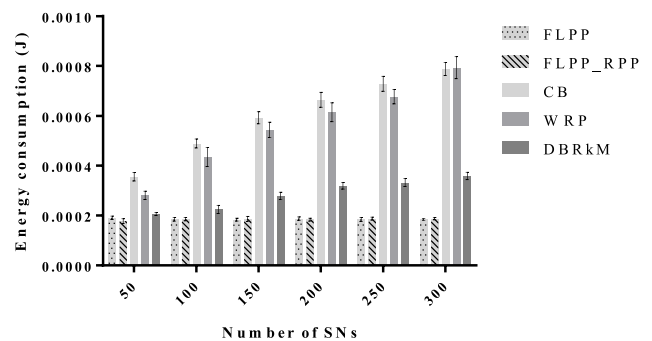


FIGURE 12. Average energy consumption.

We also assume that the MS travels at 2 m/s [34], T_{DP} threshold is 5 and has no energy constraint but is subject to the DL , i.e., 275 [34], 1200 [26] and 2400 seconds when the area size is $220 \text{ m} \times 220 \text{ m}$, $500 \text{ m} \times 500 \text{ m}$ and $1000 \text{ m} \times 1000 \text{ m}$, respectively.

We compare FLPP and FLPP_RPP with another algorithm called delay bound reduced K-means (DBRkM) [34], which selects RPs based on K-means and determines the MS traveling path based on a weight function. In addition, we also compare WRP [30] and CB [22] (operating under the DL constraint) with these three methods. All comparison results are shown in graphs (together with the corresponding 95% confidence intervals) regarding the following aspects: 1) average energy consumption of all data-collection rounds, 2) network lifetime, i.e., the duration between the time the system starts and the time the first SN's energy is depleted, 3) the MS's average traveling distance for data collection, and 4) the number of RPs. In addition, in each evaluation, there are 20 trials on average. Finally, we average the remaining results.

B. PERFORMANCE EVALUATION OF FLPP AND FLPP_RPP IN A $220 \text{ M} \times 220 \text{ M}$ AREA

We evaluated the performance of FLPP and FLPP_RPP for a small area in terms of the following three aspects.

1) AVERAGE ENERGY CONSUMPTION

Figure 12 shows the average power consumption versus the number of SNs. FLPP and FLPP_RPP always consume less power than DBRkM, WRP, and CB do because of

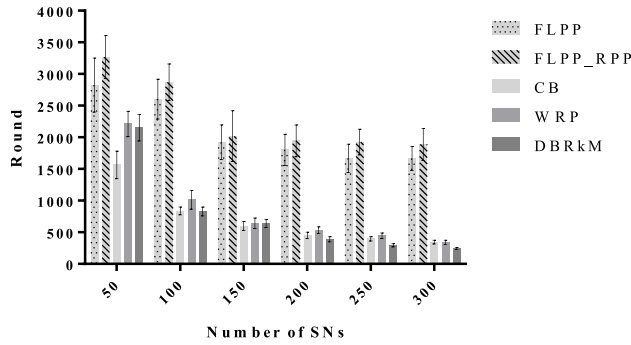


FIGURE 13. Network lifetime.

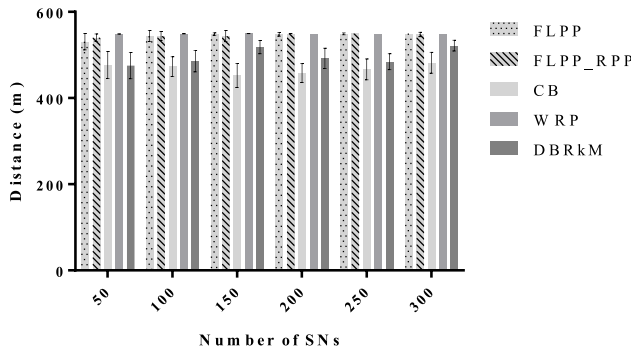


FIGURE 14. Round-trip distance (m) of the MS.

the different numbers of RPs. (FLPP spends 14%, 54% and 64% less energy than DBRkM, WRP and CB, respectively. FLPP_RPP outperforms FLPP by 0.6%.) CB randomly selects RPs, resulting in RPs at unsuitable locations. WRP uses weights to select RPs, so it outperforms CB. DBRkM yields higher performance than WRP because of its enhanced RP-selection method. FLPP and FLPP_RPP initially attempt to select as many RPs as possible, thereby increasing the likelihood that nodes transmit data directly to the MS without passing data through the RP. Shown as the I shape on the top of each bar, the 95% confidence intervals of the average energy consumptions of FLPP, FLPP_RPP, CB, WRP, and DBRkM are [0.00018, 0.00019], [0.00018, 0.00019], [0.00058, 0.00063], [0.00052, 0.0006], and [0.00027, 0.0003], respectively.

2) NETWORK LIFETIME

Figure 13 shows a performance evaluation of the network lifetime (i.e., duration from the network start time until the first node runs out of energy) as the number of SNs increases. The network lifetimes of FLPP and FLPP_RPP are significantly longer than those of DBRkM, WRP, and CB because they allow more RPs to be deployed. (FLPP allows nodes to live 58%, 53% and 60% longer than DBRkM, WRP and CB do, respectively. In this aspect, FLPP_RPP outperforms FLPP by 10%.) As a result, the number of multihop transmissions in the network is smaller, lowering the overall RP energy consumption. (Note that RPs often run out of energy before the SNs do.) The communication protocol, including signaling distribution for path redesign, is not within the scope of this

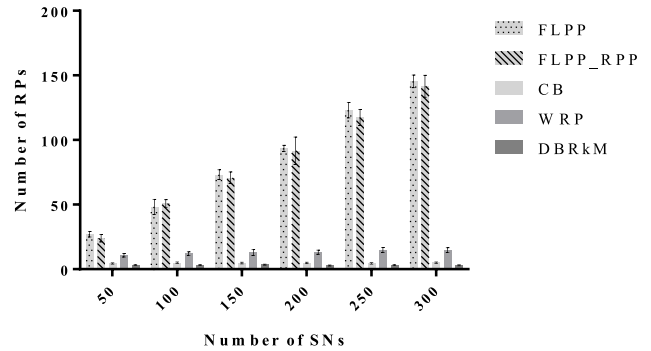


FIGURE 15. Number of RPs.

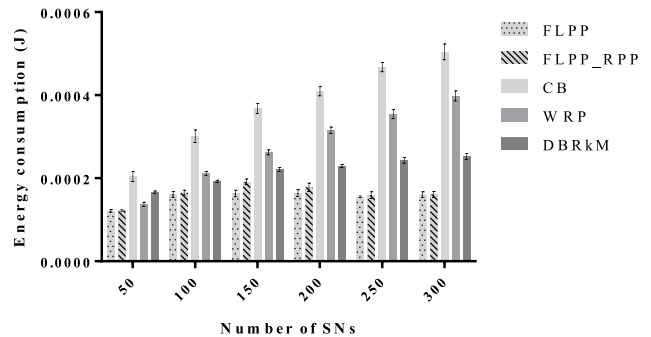


FIGURE 16. Average energy consumption.

work: we did not include the energy consumed to control signaling transmission in the experiments.

3) AVERAGE TRAVELING TIME OF THE MS

Figure 14 shows the length of the MS traveling paths as the number of SNs increases: no paths are longer than the 550 m limit ($MS\ speed \times DL = 2 \times 250 = 550$). FLPP, FLPP_RPP and WRP utilize the DL better than the others, as more time is required by the MS to traverse the paths yielded by FLPP, FLPP_RPP and WRP. This is because FLPP and FLPP_RPP use the proposed function (i.e., reducing path length from $STPath$ to $TxEPath$) to design routes, while WRP iteratively verifies that all nodes are suitable for being RPs and that the path distance does not exceed the limit.

4) THE NUMBER OF RPS

Figure 15 shows the number of RPs versus the number of SNs. FLPP and FLPP_RPP always deploy more RPs than DBRkM, WRP, and CB because FLPP and FLPP_RPP allow the MS to collect the data within the RPs' communication ranges. Consequently, the MS has more remaining energy to collect data from more RPs under the same DL limitation. (FLPP deploys larger numbers of RPs by 2.8%, 85%, 94% and 96% than those of FLPP_RPP, WRP, CB and DBRkM, respectively).

C. PERFORMANCE EVALUATION OF FLPP AND FLPP_RPP IN A 500 M × 500 M AREA

We evaluated the performance of FLPP and FLPP_RPP in a large area in three aspects, as follows.

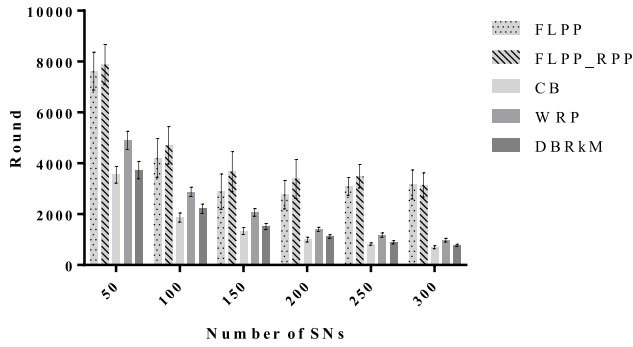


FIGURE 17. Network lifetime.

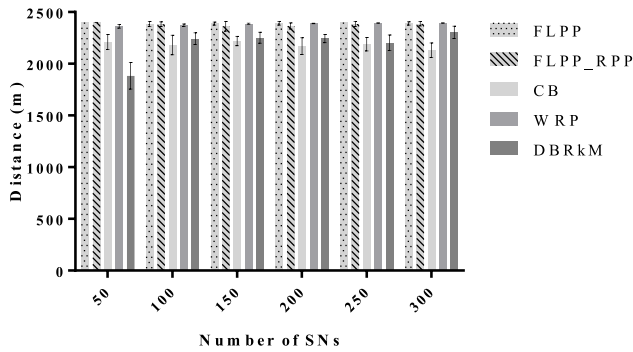


FIGURE 18. Round-trip distance (m) of the MS.

1) AVERAGE ENERGY CONSUMPTION

Figure 16 shows the average power consumption as the number of SN increases. Similar to the results in the experiments with a small area, FLPP and FLPP_RPP always consume less power than DBRkM, WRP, and CB do due to the larger DL, which contributes to the lower energy consumption in all traveling rounds. FLPP consumes 17%, 26% and 53% less energy than DBRkM, WRP and CB, respectively, and FLPP_RPP outperforms FLPP by 3%.

2) NETWORK LIFETIME

Figure 17 illustrates the network lifetime as the number of SNs increase. FLPP and FLPP_RPP have a longer overall network life than DBRkM, WRP, and CB do due to the increase in the number of RPs and the larger DL. The network based on FLPP lives 48%, 36% and 52% longer than those based on DBRkM, WRP and CB, respectively, and the network lifetime of FLPP_RPP is 13% longer than that of FLPP.

3) AVERAGE TRAVELING TIME OF THE MS

Figure 18 illustrates the path lengths the MS travels to collect data versus the number of SNs. FLPP, FLPP_RPP and WRP have higher utilization of the DL than CB and DBRkM do (i.e., the paths obtained from FLPP, FLPP_RPP and WRP are longer but do not exceed the DL).

4) THE NUMBER OF RPS

Figure 19 shows the number of RPs versus the number of SNs. FLPP and FLPP_RPP always deploy more RPs than DBRkM, WRP, and CB because FLPP and FLPP_RPP allow

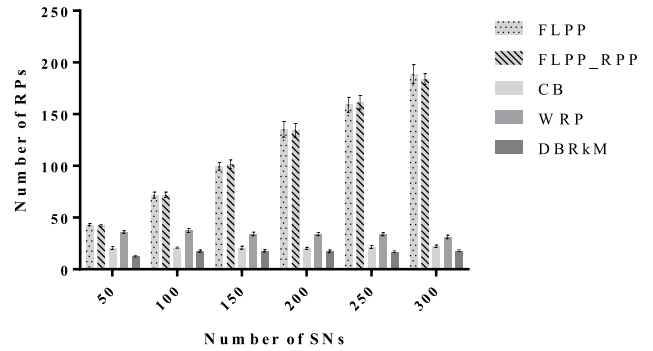


FIGURE 19. Number of RPs.

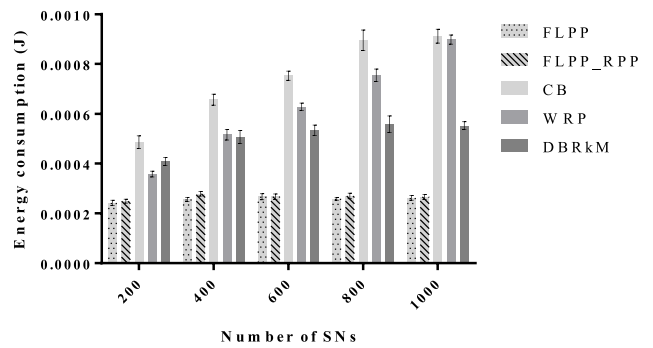


FIGURE 20. Average energy consumption.

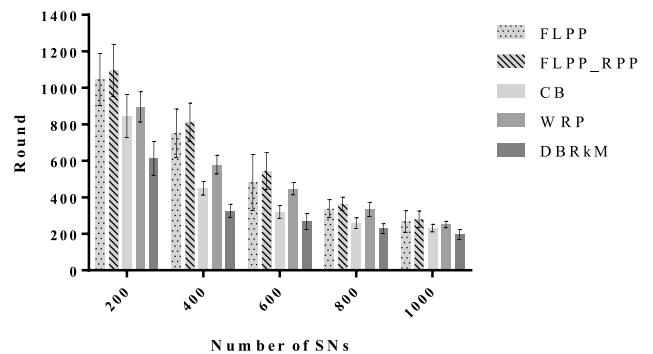


FIGURE 21. Network lifetime.

the MS to collect the data within the RPs' communication ranges. Consequently, the MS has more remaining energy to collect data from more RPs under the same DL limitation. (FLPP deploys 0.2%, 70%, 82% and 86% larger numbers of RPs than those of FLPP_RPP, WRP, CB and DBRkM, respectively).

D. PERFORMANCE EVALUATION OF FLPP AND FLPP_RPP IN A 1000 M × 1000 M AREA

We evaluated the performance of FLPP and FLPP_RPP in a very large area in three aspects, as follows.

1) AVERAGE ENERGY CONSUMPTION

Figure 20 shows the average power consumption as the number of SN increases. Similar to the results in the experiments with the small area, FLPP and FLPP_RPP always consume less power than DBRkM, WRP, and CB due to the larger

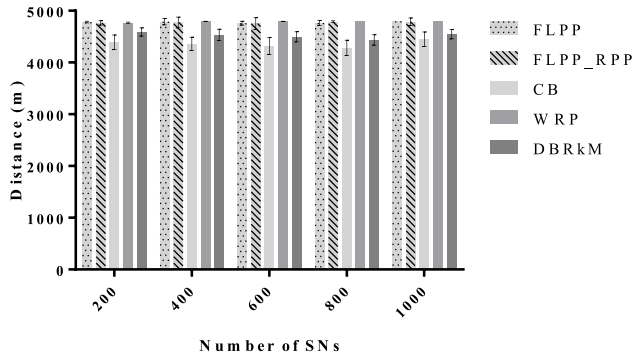


FIGURE 22. Round-trip distance (m) of the MS.

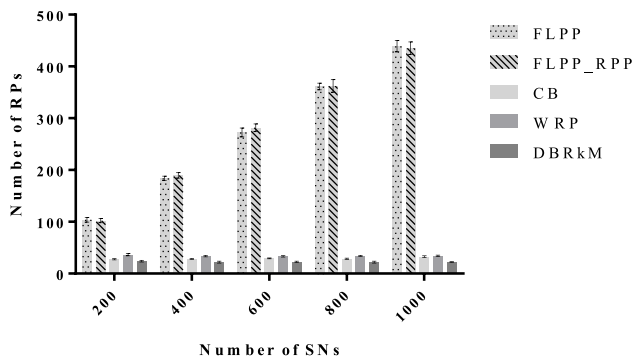


FIGURE 23. Number of RPs.

DL, which contributes to the lower energy consumption in all traveling rounds. FLPP consumes 34%, 50% and 65% less energy than DBRkM, WRP and CB, respectively, and FLPP_RPP outperforms FLPP by 1%.

2) NETWORK LIFETIME

Figure 21 illustrates the network lifetime as the number of SNs increase. FLPP and FLPP_RPP have a longer overall network life than DBRkM, WRP, and CB due to the increase in the number of RPs and the larger DL. The network based on FLPP lives 47%, 19% and 32% longer than those based on DBRkM, WRP and CB, respectively. The network lifetime of FLPP_RPP is 7% longer than that of FLPP.

3) AVERAGE TRAVELING TIME OF THE MS

Figure 22 illustrates the path lengths that the MS travels to collect data versus the number of SNs. FLPP, FLPP_RPP and WRP have higher utilization of the DL than CB and DBRkM do (i.e., the paths obtained from FLPP, FLPP_RPP and WRP are longer but do not exceed the DL).

4) THE NUMBER OF RPS

Figure 23 shows the number of RPs versus the number of SNs. FLPP and FLPP_RPP always deploy more RPs than DBRkM, WRP, and CB because FLPP and FLPP_RPP allow the MS to collect the data within the RPs' communication ranges. Consequently, the MS has more remaining energy to collect data from more RPs under the same DL limitation. (FLPP deploys 0.7%, 87%, 89% and 92% larger numbers

of RPs than those of FLPP_RPP, WRP, CB and DBRkM, respectively).

VII. DISCUSSION

A. SIGNALING AND TASK ASSIGNMENT

The communication regarding task assignment for SNs (whether they are selected as RPs, assigned as designated nodes for other SNs, or which nodes they should send data to) is out of scope of this paper. In practice, once the WSN starts, the sink node is in charge of task assignment and path design. Control signals might be distributed by the sink node in the form of broadcasting, multicasting, or gossiping protocol.

After the MS completes each traveling round, it updates all the information of the sink node, including each SN's remaining energy (making battery recharge or replacement possible). The sink node may assign SNs new tasks and design a new path for the MS to travel to collect data while distributing the control signals. In particular, the MS transmits signals to each RP, which then distribute the signals further to their associated SNs.

In our proposed method, we also rotate SNs to be RPs to prolong the WSN's lifetime. However, the path redesign and SNs new task assignment should not be conducted every time the MS returns because they might impose more computational time and waste more energy for control signaling. For example, path rerouting can be conducted periodically (e.g., after the MS completes n rounds). In our work, based on Equation (6), we monitor the difference between the average energy and the minimum energy of all the SNs: the path is not changed until the difference is greater than a specified threshold.

In general, research works that require the MS to collect data are based on the centralized concept. All processes in our proposed method are executed by the centralized sink node (i.e., the base station), which has an unlimited power supply. The base station is required only when the MS returns from data gathering and needs to transfer the sensor data to the base station. The distributed route computation demands more computational energy from SNs, reducing the network lifetime and increasing expenses on SNs' battery replacements (especially if the SNs are deployed at the locations that are very hard to access).

Furthermore, the centralized and distributed methods can be complementary to each other. In cases where the MS is down, these events are detected by the base station, and SNs close to the base station could be triggered to perform distributed routings, as introduced by the other works [44]–[47].

B. UPON ENERGY DEPLETION

According to the experimental results, RPs generally run out of energy faster than SNs do. The RPs consume energy according to Equation (2). Specifically, the greater the number of SNs associated with an RP, the shorter the RP's longevity. We can address this issue by various means, such as supplying larger batteries to RPs or increasing the number

of RPs to help balance the load. However, the former method requires static RP locations and has a higher cost. For the second method, a greater number of RPs results in a longer MS traveling time, which is limited by the DL constraint. Consequently, we proposed a method to reduce the path length (called RBR) such that the number of RPs can be increased while the path still satisfies the DL constraint.

The energy of RPs, designated nodes, or relay nodes might be depleted before the MS returns to the sink node to have a new path redesigned. In such cases, the data of the SNs associated with the RPs or the designated nodes are lost; the maximum duration of the loss is equal to DL . The data can be collected again once the path is rerouted by the sink node.

C. DESIGN OF THE FLS

SNs are selected as RPs according to their weight, based on NNR , AEC , RE , and $D_{SN,C}$. NNR and $D_{SN,C}$ indicate whether the SN is located in the high-density area or in the center of other related SNs. RE and AEC help in rotating SNs (with remaining energy) to be RPs. The distance between a pair of RPs appears useful for distributing RPs across the WSN. However, when the distance is included in FLS, it makes our preliminary results worse, as it does not favor RPs in high-density areas.

To determine a designated node for each SN, a promising approach is to select the RP closest to the SN as the designated node. However, such selection might overload certain RPs (and shorten the RPs' lifetime), as they are surrounded by dense SNs. In addition, MST is a potential alternative for selecting designated nodes. MST can determine the shortest path for sensor-data transmission; however, the SNs close to RPs might have to forward considerable loads from other SNs to the RPs, causing premature energy exhaustion. Instead of considering only distance, we rely on FLS to calculate each candidate's weights. The weights are determined by taking into account each candidate's remaining energy, the number of SNs associated with the candidate, and the distance from the nodes to their RP. As a result, the load is distributed more widely among all candidates, reducing the energy consumption of both designated nodes and SNs.

Our fuzzy rules designed in this paper are nonadaptive. They are suitable for uniformly distributed node placements. In the situation where nodes are distributed extremely nonuniformly, the rules may not be efficient enough to obtain weights for selecting appropriate RPs. In the future, we plan to address such issues and design adaptive and more flexible fuzzy rules, by focusing on both node density and distances.

D. RUNNING TIME ANALYSIS

Let N and M be the numbers of SNs and RPs, respectively. According to Section V, the runtime of the overall method is $O(N^2 + M^3 \log M)$, while those of CB and WRP are $O(N^2 \log N)$ and $O(N^3)$, respectively. Our proposed method outperforms CB and WRP in terms of running time in the case where N is much larger than M . Specifically, our method is efficient if 1) the density of SNs is high, i.e., one RP is the

designated node of many SNs, and 2) the DL is low, causing the number of RPs (M) to be small. Although our method requires more running time in the situation of low density and high DL , the simulation results show that it takes less than 10 minutes to execute the whole process, which is satisfactory for applications that are not time sensitive.

E. MULTIPLE MSs

In this paper, we assume, without loss of generosity, that the MS has enough power to travel a round trip under the DL constraint. However, in real WSNs, there might be some cases where the MS power storage is limited; in these scenarios the administrator can decrease the DL value and our method can adjust the path accordingly. In the future, we plan to study the MS's energy consumption model by determining the correlation between the MS's energy capacity and the round-trip distance (i.e., DL) to estimate the DL value based on the MS's energy capacity. In addition, if the capacity is relatively small so that the algorithm cannot find a single cyclic path that covers the whole area, we plan to study how to design multiple cyclic paths so that the MS can travel along them and return to the base station to recharge or swap the battery before traveling on the other paths to cover the whole area.

Furthermore, it is possible to deploy multiple MSs to collect data in a shorter time period. In addition, addition MSs allow more RPs under the same DL constraint, resulting in a lower load at each RP and a longer WSN lifetime. As future work, we will address how to coordinate multiple MSs and how to have the MSs efficiently recharge the SNs energy.

VIII. CONCLUSION

This research presents a novel route-planning method. We attempt to determine efficient cycles for MSs under delay limitations while reducing the energy consumption of SNs. The method consists of 3 phases. The first step is to determine the number of RPs and create a cyclic path. In the second step, data-collection locations are specified so that the MSs can travel within the RPs' communication ranges and collect data effectively. Last, paths for relaying sensory data from SNs are calculated, allowing each SN to choose its best RP while efficiently reducing the overall energy consumption of RPs. The experiments showed that FLPP_RPP and FLPP outperform DBRkM, WRP and CB.

In the small-area experiments, FLPP consumes 14%, 54% and 64% less energy, on average, than DBRkM, WRP and CB do, respectively. Compared with FLPP, FLPP_RPP consumes 0.6% less energy on average. In terms of the network lifetime, the energy of the networks based on FLPP lasts 58%, 53% and 60% longer than that of DBRkM, WRP and CB, while FLPP_RPP outperforms FLPP by 10%. In the larger-area experiments, FLPP uses 17%, 26%, 53% and 3% less energy, on average, than DBRkM, WRP, CB and FLPP_RPP do, respectively. The average lifetime of the FLPP-based network is 48%, 36% and 52% longer than that of DBRkM, WRP and CB, respectively, and FLPP_RPP outperforms FLPP by 13%.

In the very large area experiments, on average, FLPP uses 34%, 50%, 65% and 1% less energy than do DBRkM, WRP, CB and FLPP_RPP, respectively. The average lifetime of the FLPP-based network is 47%, 19% and 32% longer than that of DBRkM, WRP and CB, respectively. Furthermore, FLPP_RPP outperforms FLPP by 17%.

Our proposed scheme is centralized in that the sink node is in charge of grouping SNs and choosing a relaying path for each SN. Although just one MS is deployed in the WSN, the proposed algorithm could be adapted to support multiple MSs. Future research, could focus on how to use MSs to charge SNs to increase the system up time.

REFERENCES

- [1] A. U. Rahman, A. Alharby, H. Hasbullah, and K. Almuzaini, "Corona based deployment strategies in wireless sensor network: A survey," *J. Neww. Comput. Appl.*, vol. 64, pp. 176–193, Apr. 2016.
- [2] J. L. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed fault tolerance," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 215–228, Feb. 2010.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.
- [4] S. M. Diamond and M. G. Ceruti, "Application of wireless sensor network to military information integration," in *Proc. 5th IEEE Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2007, pp. 317–322.
- [5] I. Bekmezci and F. Alagöz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *Int. J. Distrib. Sensor Netw.*, vol. 5, no. 6, pp. 729–747, Nov. 2009.
- [6] H. Li, C. Ao, Y. Xu, J. Tian, and K. Yamashita, "Relay node position optimization in complex environment," presented at the IEEE Wireless Commun. Netw. Conf. (WCNC), San Francisco, CA, USA, Mar. 2017.
- [7] Y. Zeng, N. Xiong, J. H. Park, and G. Zheng, "An emergency-adaptive routing scheme for wireless sensor networks for building fire hazard monitoring," *Sensors*, vol. 10, no. 6, pp. 6128–6148, Jun. 2010.
- [8] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo, "Forest fire detection system based on wireless sensor network," presented at the IEEE Conf. Ind. Electron. Appl., Xi'an, China, May 2009.
- [9] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop Wireless Sensor Netw. Appl. (WSNA)*, 2002, pp. 88–97.
- [10] G. M. E. Rahman, K. A. Wahid, and A. Dinh, "IoT enabled low power and wide range WSN platform for environment monitoring application," in *Proc. IEEE Region Symp. (TENSYP)*, Jun. 2020, pp. 908–911.
- [11] D. Shinde and N. Siddiqui, "IOT based environment change monitoring & controlling in greenhouse using WSN," presented at the Int. Conf. Inf., Commun., Eng. Technol. (ICICET), Pune, India, Aug. 2018.
- [12] M. Rao, N. Kamila, and K. V. Kumar, "Underwater wireless sensor network for tracking ships approaching harbor," presented at the Int. Conf. Signal Process., Commun., Power Embedded Syst. (SCOPE), Paralakhemundi, India, Oct. 2016.
- [13] A. Alkandari, M. Alnasheet, Y. Alabduljader, and S. M. Moein, "Water monitoring system using wireless sensor network (WSN): Case study of kuwait beaches," in *Proc. 2nd Int. Conf. Digit. Inf. Process. Commun. (ICDIPC)*, Jul. 2012, pp. 10–15.
- [14] J. Lian, K. Naik, and G. B. Agnew, "Data capacity improvement of wireless sensor networks using non-uniform sensor distribution," *Int. J. Distrib. Sens. Netw.*, vol. 2, no. 2, pp. 121–145, 2006.
- [15] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," presented at the IEEE Int. Conf. Comput. Commun., Barcelona, Spain, Apr. 2006.
- [16] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *Wireless Netw.*, vol. 14, no. 6, pp. 831–858, Dec. 2008.
- [17] S. Hashish and A. Karmouch, "Deployment-based solution for prolonging network lifetime in sensor networks," in *Proc. IFIP Conf. Wireless Sensor Actor Netw.*, in IFIP—The International Federation for Information Processing, vol. 264, 2008, pp. 85–96.
- [18] Z. Vincze, D. Vass, R. Vida, A. Vidács, and A. Telcs, "Adaptive sink mobility in event-driven multi-hop wireless sensor networks," in *Proc. 1st Int. Conf. Integr. Internet Ad Hoc Sensor Netw. (InterSense)*, vol. 138, 2006, p. 13.
- [19] J. Luo and J.-P. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 871–884, Jun. 2010.
- [20] D. Kim, B. H. Abay, R. N. Uma, W. Wu, W. Wang, and A. O. Tokuta, "Minimizing data collection latency in wireless sensor network with multiple mobile elements," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 504–512.
- [21] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 385–410, Apr. 2007.
- [22] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proc. IEEE Local Comput. Netw. Conf. (LCN)*, Oct. 2010, pp. 582–589.
- [23] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, vol. 9, 2008, pp. 231–240.
- [24] Y.-C. Wang and K.-C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1527–1540, Jul. 2019.
- [25] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in mobility-assisted wireless sensor networks," presented at the IEEE Int. Real-Time Syst. Symp. (RTSS), Tucson, AZ, USA, Dec. 2007.
- [26] A. Alsaafin, A. M. Khedr, and Z. Al Aghbari, "Distributed trajectory design for data gathering using mobile sink in wireless sensor networks," *AEU-Int. J. Electron. Commun.*, vol. 96, pp. 1–12, Nov. 2018.
- [27] B. Chazelle, "A minimum spanning tree algorithm with inverse-Ackermann type complexity," *J. ACM*, vol. 47, no. 6, pp. 1028–1047, Nov. 2000.
- [28] S. Rani and S. H. Ahmed, "Multi-hop energy efficient routing," in *Multi-hop Routing in Wireless Sensor Networks*. Singapore: Springer, 2016, pp. 15–28. [Online]. Available: <https://link.springer.com/book/10.1007/978-981-287-730-7>
- [29] J. LaRusic, A. P. Punnen, and E. Aubanel, "Experimental analysis of heuristics for the bottleneck traveling salesman problem," *J. Heuristics*, vol. 18, no. 3, pp. 473–503, Jun. 2012.
- [30] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.
- [31] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in wireless sensor networks with mobile elements," *IEEE Trans. Mobile Comput.*, vol. 7, no. 12, pp. 1430–1443, Dec. 2008.
- [32] C.-F. Cheng and C.-F. Yu, "Data gathering in wireless sensor networks: A combine-TSP-reduce approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2309–2324, Apr. 2016.
- [33] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime," *IEEE Sensors J.*, vol. 16, no. 1, pp. 137–144, Jan. 2016.
- [34] A. Kaswan, K. Nitesh, and P. K. Jana, "Energy efficient path selection for mobile sink and data gathering in wireless sensor networks," *AEU-Int. J. Electron. Commun.*, vol. 73, pp. 110–118, Mar. 2017.
- [35] C.-F. Cheng, L.-H. Li, and C.-C. Wang, "Data gathering with minimum number of relay packets in wireless sensor networks," *IEEE Sensors J.*, vol. 17, no. 21, pp. 7196–7208, Nov. 2017.
- [36] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: Energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 2, pp. 890–901, Jan. 2018.
- [37] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "ACO-based mobile sink path determination for wireless sensor networks under non-uniform data constraints," *Appl. Soft Comput.*, vol. 69, pp. 528–540, Aug. 2018.
- [38] S. A. Sert, A. Alchihabi, and A. Yazici, "A two-tier distributed fuzzy logic based protocol for efficient data aggregation in multihop wireless sensor networks," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3615–3629, Dec. 2018.
- [39] H. Q. Qadori, Z. A. Zukarnain, Z. M. Hanapi, and S. Subramaniam, "FuMAM: Fuzzy-based mobile agent migration approach for data gathering in wireless sensor networks," *IEEE Access*, vol. 6, pp. 15643–15652, Mar. 2018.

- [40] P. K. Donta, B. S. P. Rao, T. Amgoth, C. S. R. Annavarapu, and S. Swain, "Data collection and path determination strategies for mobile sink in 3D WSNs," *IEEE Sensors J.*, vol. 20, no. 4, pp. 2224–2233, Feb. 2020.
- [41] F. Chen, N. Wang, R. German, and F. Dressler, "Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications," *Wireless Commun. Mobile Comput.*, vol. 10, no. 5, pp. 609–621, May 2010.
- [42] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
- [43] E. P. Ephzibah, "Time complexity analysis of genetic-fuzzy system for disease diagnosis," *Adv. Comput., Int. J.*, vol. 2, no. 4, pp. 23–31, Jul. 2011.
- [44] S. Chattopadhyay, N. Sett, S. Nandi, and S. Chakraborty, "FLIPPER: Fault-tolerant distributed network management and control," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 421–427.
- [45] S. Mitra, A. Das, and S. Mazumder, "Comparative study of fault recovery techniques in wireless sensor network," in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Dec. 2016, pp. 130–133.
- [46] V. K. Menaria, S. C. Jain, N. Raju, R. Kumari, A. Nayyar, and E. Hosain, "NLFFT: A novel fault tolerance model using artificial intelligence to improve performance in wireless sensor networks," *IEEE Access*, vol. 8, pp. 149231–149254, 2020.
- [47] X. Di, Z. Zhang, C. Ao, J. Tian, K. Ozaki, Y. Wen, and H. Fujita, "Energy-efficiency of load-balancing routing for wireless convergecast networks: Centralized versus distributed implementation," in *Proc. 11th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 343–348.



CHATCHAI PUNRIBOON received the B.S. and M.S. degrees in computer science from Khon Kaen University, Thailand, in 2012 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science, Faculty of Science. His research interests include image processing, vehicle routing, wireless sensor networks, and the Internet of Things.



CHAKCHAI SO-IN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Washington University, St. Louis, MO, USA, in 2010. He was an Intern with CNAP-NTU (SG), Cisco Systems, WiMAX Forums, and Bell Labs, USA. He is currently a Professor with the Department of Computer Science, Khon Kaen University. His research interests include the Internet of Things (IoT), mobile computing, wireless/sensor networks, cellular networks, signal processing, computer networking, cybersecurity, and applied artificial intelligence. He is also a Senior Member of ACM. He has served as an Editor for IEEE ACCESS, PLOS ONE, PeerJ, and ECTI-CIT and as a Committee Member for many conferences/journals, such as Globecom, ICC, VTC, WCNC, ICNP, ICNC, PIMRC, IEEE TRANSACTIONS, IEEE LETTER, *IEEE Magazines*, *Computer Networks*, and *Communications*. He has authored over 100 international publications and ten books, including some in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), *IEEE Magazines*, and *Computer Network/Network Security Labs*.



PHET AIMTONGKHAM received the B.S., M.S., and Ph.D. degrees in information technology from the Department of Computer Science, Khon Kaen University, Thailand, in 2013, 2017, and 2020, respectively. He is currently a Lecturer and a Researcher with the Department of Computer Science, Faculty of Science, Khon Kaen University. His research interests include computer networking, the Internet of Things (IoT), multimedia networks, and machine learning.



NUTTHANON LEELATHAKUL received the B.Eng. degree (Hons.) in electrical engineering from Kasetsart University, Bangkok, Thailand, in 1999, the M.Eng. degree in electrical and computer engineering from Cornell University, in 2004, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, in 2010. He has been a Faculty Member with the Faculty of Informatics, Burapha University, Chon Buri, Thailand, since 2010. His current research interests include computer networking, security in the IoT, deep learning, and data analytics.

...