IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2018, accepted February 27, 2018, date of publication March 9, 2018, date of current version April 4, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2814064

FuMAM: Fuzzy-Based Mobile Agent Migration Approach for Data Gathering in Wireless Sensor Networks

HUTHIAFA Q. QADORI[1,2], ZURIATI AHMAD ZUKARNAIN[1], (Member, IEEE), ZURINA MOHD HANAPI[1], (Member, IEEE), AND SHAMALA SUBRAMANIAM[1,3] (Member, IEEE)

[1]Department of Wireless and Communication Technology, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia
[2]University of Anbar, Ramadi 31001, Iraq
[3]Sports Academy, Universiti Putra Malaysia, Serdang 43400, Malaysia

Corresponding authors: Huthiafa Q. Qadori (Huthiafaqadori@gmail.com) and Zuriati Ahmad Zukarnain (zuriati@upm.edu.my)

This work was supported in part by Universiti Putra Malaysia and in part by the Ministry of Higher Education, University of Anbar, Iraq.

ABSTRACT A mobile agent (MA) was recently proposed to provide an alternative solution for traditional data gathering in wireless sensor networks. An MA is a software component that can migrate among network nodes by following an assigned itinerary (or path). Instead of transporting data from the nodes to the processing unit (e.g., sink) for data gathering, the MA visits each node, and thus, it performs data gathering locally. The MA has two types of itinerary planning: single-agent itinerary planning (SIP) and multi-agent itinerary planning (MIP). The MIP was introduced to address the drawbacks of the SIP in terms of task duration, energy consumption, and reliability. Despite the advantages of the MIP, determining the optimal itinerary for each MA in the MIP poses a considerable challenge. Most proposed itineraries adopt a static itinerary in which the nodes to be visited by the MA are predetermined at the sink node. The distance among nodes is the only parameter that has been used to determine the itinerary of the MA. Other parameters, such as the remaining energy and a number of neighbors, have not been considered. This omission can negatively impact MA migration and result in an unsuccessful MA round-trip, particularly when the remaining energy of the node is insufficient to transfer an MA to the next hop. In this paper, a fuzzy-based MA migration approach (FuMAM) is proposed to determine appropriate itinerary for an MA by considering three parameters: distance, remaining energy, and a number of neighbors. Simulation experiments show that the FuMAM approach improves the rate of the successful MA round-trip and network lifetime. Moreover, the proposed FuMAM approach outperforms the compared algorithms in terms of energy distribution usage among nodes.

INDEX TERMS Data gathering, mobile agent, itinerary, static itinerary, dynamic itinerary, MIP.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been recently acknowledged as a promising technology for many applications, such as military, environmental, and health. A WSN involves the deployment of hundreds or thousands of tiny sensor nodes that communicate wirelessly with one another to sense specific actions in a field of interest. The process of collecting sensory data from sensor nodes and sending it back to the processing unit (e.g., sink node) is called data gathering. In data gathering, each node sends its sensed data individually to the sink node using several hops. This process leads to data congestion, increased latency, and high energy consumption, particularly for nodes located near the sink, due to the huge amount of data flow. To mitigate these issues, a mobile agent (MA), which is a software component, was proposed in the literature [1] to provide an alternative solution for traditional data gathering in WSNs.

In MA-based data gathering, the itinerary of an MA can be planned using two approaches: single itinerary planning (SIP) and multi itinerary planning (MIP). In SIP, a single MA is dispatched from the sink and migrates to the source nodes to perform data gathering [2]–[5]. By contrast, several MAs are distributed to the network and work concurrently in MIP [6]–[8]. However, SIP approaches only perform well

in small-or medium-scale sensor networks. In large-scale networks, SIP exhibits the following drawbacks [9]:

1) Long delays because of migration to hundreds of source nodes
2) An increase in MA's packet size due to the aggregation of data from a huge number of visited source nodes
3) Low reliability when MA accumulates a considerable amount of data
4) The probability of losing the MA's packet increases when a single MA visits numerous source nodes

Despite the advantages of MIP, determining an optimal itinerary for each MA in MIP remains a challenging issue. Finding an optimal itinerary for an MA has been proven as an NP-hard problem [5]. An inappropriate MA itinerary may lead to a highly inefficient overall network performance. The determination of an MA's itinerary refers to selecting a sequence of source nodes that will be visited by the MA. In MA-based data gathering, the determination of MA's itinerary can be classified as a static, dynamic or hybrid itinerary [10], [11]. In a static itinerary, the sequence of source nodes to be visited by an MA is calculated at the sink before the MA starts its migration. In a dynamic itinerary, the sequence of source nodes to be visited by the MA is identified on the fly at each source node. Finally, in a hybrid itinerary, the source nodes to be visited are selected at the sink, but the visiting sequence is computed on the fly by the MA. However, dynamic and hybrid itineraries consume valuable node energy resources and adapt to the large size of an MA. Such conditions are caused by the MA carrying the next hop computation code at each node during the migration process. By contrast, a static itinerary consumes less energy than its dynamic and hybrid counterparts because each MA carries a predetermined itinerary that has been calculated at the sink.

The application of an MA determines which itinerary strategy should be adopted. A dynamic itinerary is more applicable for target tracking applications given that the real-time adaptation of MA itineraries is required to provide progressive accuracy [4], [12]. By contrast, a static itinerary is more suitable for data monitoring applications wherein the measurements of physical quantities (such as humidity and temperature) are periodically gathered at the sink. We focus on static itinerary in this research.

Several static itinerary approaches have been proposed. Most of these approaches consider the distance among nodes and MA's hop energy cost as the main parameters for computing an MA itinerary, whereas other parameters, such as the node's remaining energy and number of neighbors, are neglected. MA migration is affected when only distance and MA's hop energy cost are considered. This situation will result in an unsuccessful MA's round-trip, particularly when the remaining energy of the node is insufficient to transfer the MA to the next hop. Moreover, a particular node can be selected repeatedly during data gathering rounds due to its nearest distance. This circumstance will generate an imbalanced node energy usage by the MA, which will reduce network lifetime. Accordingly, this research proposes a fuzzy-based MA migration approach (FuMAM) to mitigate the aforementioned issues. The FuMAM approach determines an appropriate itinerary for an MA by considering three parameters: distance, remaining energy, and number of neighbors. The proposed FuMAM approach increases the rate of successful MA's round-trip. Furthermore, the FuMAM algorithm improves network lifetime by selecting the node with a high residual energy as the next hop for MA migration.

The remainder of the paper is organized as follows. Section II explains related works. Section III presents the proposed FuMAM approach. Section IV describes the performance evaluation and experimental results. Section V provides the conclusion of the study and suggests a future work direction.

II. RELATED WORKS

Notably, MIP consists of two or more SIPs working concurrently to visit groups of source nodes. Each single MA's itinerary has a sequence of source nodes to be visited. The visiting order of such nodes significantly impacts overall network performance. This section reviews several static itinerary algorithms proposed for determining the appropriate itinerary for MA in WSNs. Qi and Wang [2] suggested two static itinerary approaches: local closest first (LCF) and global closest first (GCF) to determine an itinerary for MA migration. In LCF, the MA looks for the next hop node with the shortest distance from the current location. In GCF, the MA looks for the next hop node with the shortest distance to the sink. Figure 1 illustrates the difference between the LCF and GCF approaches.

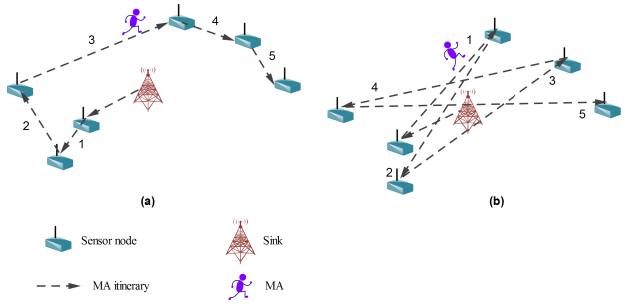

FIGURE 1. (a) LCF approach; (b) GCF approach.

An MA-based directed diffusion (MADD) was proposed in [3]. MADD is similar to LCF but differs in the sense that an MA selects the farthest node from the sink as the first source node. Although the LCF, GCF, and MADD methods are easy to implement, they are not scalable because the itinerary of an MA is determined according to the distance among source nodes. In [13], two algorithms, namely, itinerary energy minimum for first source selection (IEMF) and the itinerary energy minimum algorithm (IEMA), were introduced to achieve energy-efficient itineraries. IEMF adopts the round robin method, in which every node is tentatively selected as the first source node. Then, the LCF algorithm is applied to

the remaining source nodes. Such process generates different candidate itineraries, whereby each itinerary corresponds to an energy cost. Subsequently, an itinerary with the lowest energy cost is selected by IEMF. By contrast, IEMA is the iterative version of IEMF, such that IEMA determines the visiting order of the remaining source nodes along the first source node. Despite the advantages of the IEMF and IEMA algorithms in terms of energy efficiency, these algorithms are evidently still based on the LCF algorithm compared with LCF and GCF. LCF looks for the next MA's hop depending on the current location of the MA instead of looking for global network information. Furthermore, the LCF, GCF, IEMF, and IEMA techniques were developed with a single MA itinerary (SIP), which exhibits low performance in a large-scale network.

In a large-scale network where the MIP approach occurs, determining the optimal MA itinerary is motivated by considering the global formation of the network. In [14], a near-optimal itinerary design (NOID) algorithm was introduced to calculate the number of MAs and their itineraries. The main parameter used for calculating the cost weight is the distance between source nodes. This distance was determined using the minimum spanning tree (MST)-based NOID algorithm. The NOID algorithm uses a trade-off function for balancing, such that a source node with insufficient energy will be visited first by the MA (at the time when the accumulated data of the MA are small). Chen *et al.* [15] proposed an algorithm called BST-MIP. Network topology is modeled as a totally connected graph (TCG), such that the vertices represent the nodes in the network and the estimated hop count between each two source nodes provides the weight of an edge. The BST-MIP algorithm adopts the same method (MST) presented in the NOID algorithm. Nevertheless, the BST-MIP algorithm differs from NOID because it uses a balancing factor a during the calculation of weight in TCG. The balancing factor a is used to achieve flexible control of the trade-off between energy cost and task duration.

In [7], a genetic algorithm (GA)-based MIP was developed to determine the number of MAs and their itineraries in MIP. The concept of GA-MIP is to determine the number of MAs and their assigned source nodes using a two-level coding GA-based method. The coding represents a gene that contains a source ordering code (sequence array) and a source grouping code (group array). The sequence array contains segments, such that each segment corresponds to a number of source nodes that must be visited by an MA. The group array includes numbers. Each number in the array represents the number of source nodes of each segment in the sequence array. The two basic operations of GA (crossover and mutation) are used in each iteration, and a fitness function is applied to select better genes. Although GA-MIP exhibits good performance in terms of delay and energy consumption, it has high computation complexity because it maintains global information about the network in each iteration.

In the aforementioned MIP algorithms, the geographic information of the sensor nodes and MA migration cost are the main parameters used to determine the optimal number of MAs and their itineraries. In [16], the greatest information in the greater memory-based MIP (GIGM-MIP) algorithm was proposed. The GIGM-MIP algorithm considers geographic information to formulate the optimal number of MAs and their itineraries and the data size in each partition. In GIGM-MIP, the k-means algorithm is used to partition the network into K clusters (partitions). After partitioning the network, the GIGM-MIP algorithm calculates the data size of the source nodes in each partition. This data size will then determine the number of MAs that will be assigned to that partition, such that each partition may have more than one MA. Then, the source node that has the maximum data size among the other source nodes is assigned to the MA with the greatest free payload data. This process is repeated until all MA payload data have nearly the same size. Although this solution balances the carried data among the distributed MAs and reduces energy consumption, the itinerary of each MA still relies on the LCF algorithm, which considers the distance between the current location of an MA and the next hop node as a parameter for calculating the next MA's hop.

A new immune-inspired algorithm, named the clonal selection algorithm for MIP (CSA- MIP) [17], was proposed to calculate the number of MAs and their itineraries. CSA-MIP uses the same two-level coding method presented in GA-MIP. The difference is that CSA-MIP applies a two-stage evolutionary search procedure to achieve both global and local search capabilities, with each stage having a different mutation operator. When these search procedures are applied to the obtained solutions in CSA-MIP, a variation in the number of MAs can be obtained. Moreover, imbalance in the number of sensor nodes assigned to each MA is reduced. This condition will potentially increase the possibility of achieving improved solutions.

In most of the aforementioned approaches, the only parameters used to determine MA itinerary are the distance among nodes and the link cost between each pair of nodes. Other parameters, such as the node's remaining energy and the number of neighbors of the next hop candidate, are disregarded. MA migration will be adversely affected if only the parameters of distance and energy cost are considered. This circumstance will result in an unsuccessful MA round-trip given that the nearest node may have insufficient energy to transfer an MA to the next hop. In addition, uneven energy dissipation can decrease network lifetime because a particular node can be selected repeatedly during data gathering rounds due to its nearest distance. To mitigate these issues, a fuzzy-based MA migration approach (FuMAM) is proposed in this research. The FuMAM approach determines an appropriate itinerary for an MA by considering three parameters: distance, remaining energy, and number of neighbors for each candidate node by using a fuzzy logic system (FLS). The main objective of the proposed FuMAM approach is to increase the rate of successful MA round-trip and improve network lifetime.

III. FuMAM APPROACH

This section presents the proposed FuMAM approach. The FuMAM method is based on FLS, a computational intelligent system (CIS). FLS is a decision-making algorithm that has been used to enhance network performance [18]. FLS is defined as a nonlinear input-output mapping [19]. It exhibits human intelligence reasoning behavior in handling incomplete data and unexpected situations. For designing a WSN routing protocol, FLS offers advantages in terms of transmission media characteristics and protocol performance, thereby making fuzzy representation easy and realistic. Moreover, FLS can efficiently address several issues regarding WSNs [20]. FLS receives values (parameters) from the nodes as inputs and computes them to infer output values. The computation process includes converting input values (crisp input) into a fuzzy set and processing it using rule-based FLS, where rules are expressed as a collection of IF-THEN statements that are intended to mimic human reasoning. Subsequently, the processed fuzzy set is converted back to obtain an aggregated output value (crisp output) [21].

When determining an itinerary for an MA in WSNs, numerous parameters, such as remaining energy, distance, and number of neighbors, should be considered simultaneously when selecting the next MA's migration hop. Therefore, choosing the appropriate node for the next MA's hop under these multi-parameters exerts a considerable influence on overall network performance. Here, FLS can offer an appropriate solution for this type of multi-parameter evaluation problem. That is, FLS can integrate several node selection parameters.

In the proposed FuMAM approach, FLS is used to calculate the MA's hop sequences between each two source nodes (intermediate nodes) by computing the probability of each candidate node based on their input parameters. Before the sink dispatches MAs to the network for data gathering, it must maintain the global information of all the nodes to partition the network and determine the visiting order of each MA for both source and intermediate nodes. The visiting order of source nodes is obtained by using the LCF algorithm, whereas the visiting order of intermediate nodes is calculated by using FLS. For each MA's hop among the intermediate nodes, three parameters are used as inputs to FLS for each candidate node. These parameters include remaining energy, distance to the source node, and number of neighbors of the candidate node. After FLS determines the next node for an MA, the selected node will be added to the MA's visiting order. The same process will be repeated for the next hop of MA's migration until all of the MAs' itineraries for all partitions are determined. Figure 2 shows the flowchart of the FuMAM approach.

A. FLS INPUT PARAMETERS

As mentioned earlier, three parameters were used as inputs to FLS in our proposed FuMAM approach. In this work, we limited the input parameters to three inputs to avoid the problem of fuzzy rule explosion. Increasing the number of input parameters in FLS will increase the complexity of the rule base in the proposed approach. However, a method, such as the hierarchical fuzzy system (HFS) [22], is intended to reduce the size of the rule base while maintaining adequate accuracy. The three parameters used in this research are described as follows:

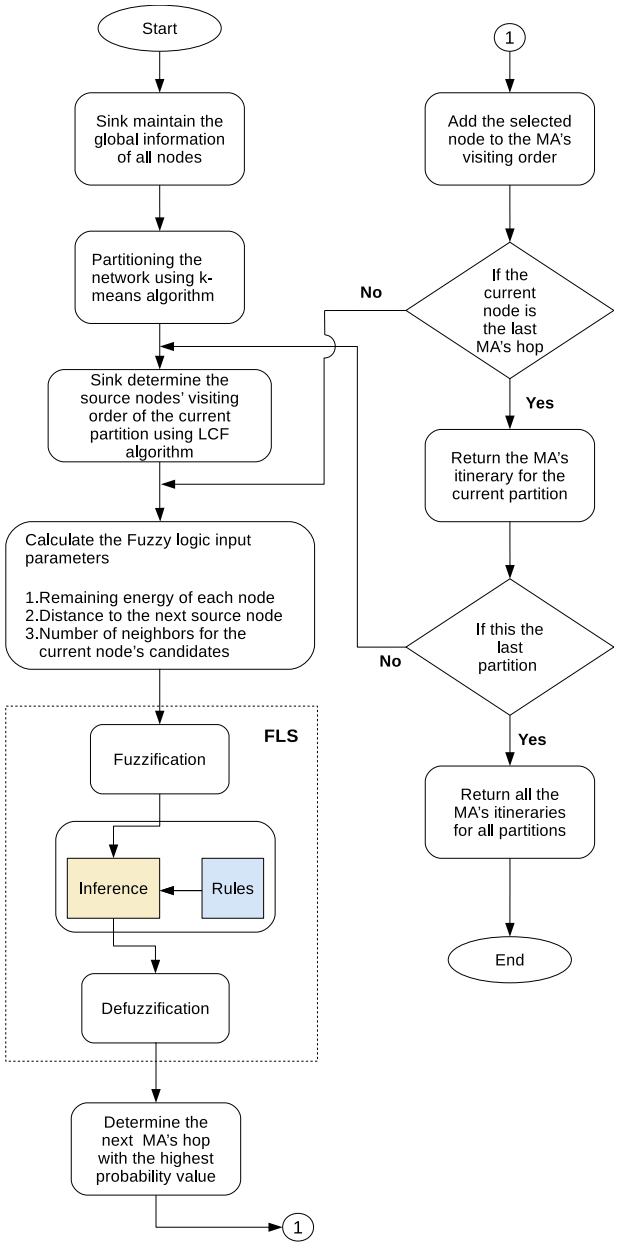

FIGURE 2. The flow chart of FuMAM approach.

1) NODE'S REMAINING ENERGY

This parameter indicates the remaining energy of a node. At the outset, all nodes have the same initial energy. After starting the first data gathering task, the nodes begin to lose energy because of the MA migration process. Therefore, the remaining energy of each node can be calculated

using Equation (1).

$$RE = NodeCurrentEnergy - CEnergy \tag{1}$$

where *CEnergy* is the energy spent at a particular node to receive, process, and transmit an MA [7], [23]. The node with the maximum remaining energy is preferred as the next hop for the MA.

2) DISTANCE TO THE SOURCE NODE

This parameter refers to the distance of a candidate node to the next source node. The geographical information of all the deployed nodes in the network is assumed to be known to the sink; hence, the distance between each two nodes can be easily calculated. The candidate node with the shortest distance from the next source node is highly recommended as the next hop for the MA because the cost of MA migration is proportional to the distance among nodes [15]. The distance between each candidate node and the next source node can be calculated using Equation (2)

$$D(cand) = \sqrt{(SN_x - Cand_x) + (SN_y - Cand_y)} \tag{2}$$

where (SN_x and SN_y) represent the position of the next source node points, and ($Cand_x$ and $Cand_y$) determine the position of each candidate node.

3) NUMBER OF NODE'S NEIGHBORS

This parameter denotes the number of nodes that lies within the radius of each candidate node within the location of the current MA. The number of candidate's neighbors is used in this research to ensure that the MA always has a sufficient number of candidates for the subsequent hop decisions due to the multi-hop migration model for MA-based data gathering. In certain cases, if an MA reaches a node with a fewer number of neighbors (for example, two neighbors) and the remaining energy of the neighbors is extremely low, then the MA will select one of the neighbors as its next hop. Consequently, the MA will be dropped for selecting a node with low residual energy. Therefore, using the number of candidate's neighbors as one of the decision factors for FLS can provide an effective solution for overall performance.

B. FLS DESIGN IN FuMAM

The main objective of this research is to determine an optimal itinerary for an MA using FLS. The proposed FuMAM approach should calculate the MA's itinerary by considering three parameters (remaining energy, distance, and number of neighbors) for each MA's next hop selection. The concept of FLS is based on four main phases (Figure 3): fuzzification, rule evaluation, aggregation, and defuzzification [24]. They are used in FuMAM to calculate the probability values of each candidate node to be selected as the next MA's migration hop. The details of the operations of the four phases are as follows:

- **Fuzzification:** The crisp input values (remaining energy, distance, and number of neighbors) for each candidate node are forwarded to the fuzzification phase (Figure 3). In this phase, the input values are converted into suitable linguistic values by mapping each value into the corresponding universal set (Figure 4). For example, if X is the universal set, its then elements are denoted by x, such that the fuzzy set A in X is a set.

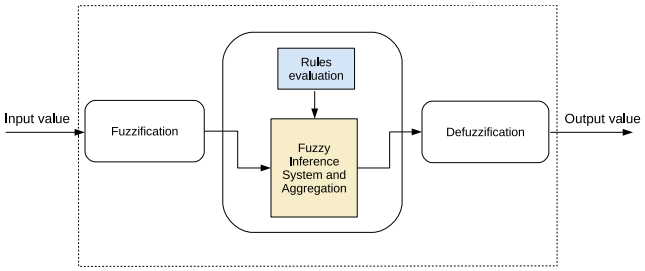

FIGURE 3. Fuzzy logic system phases.

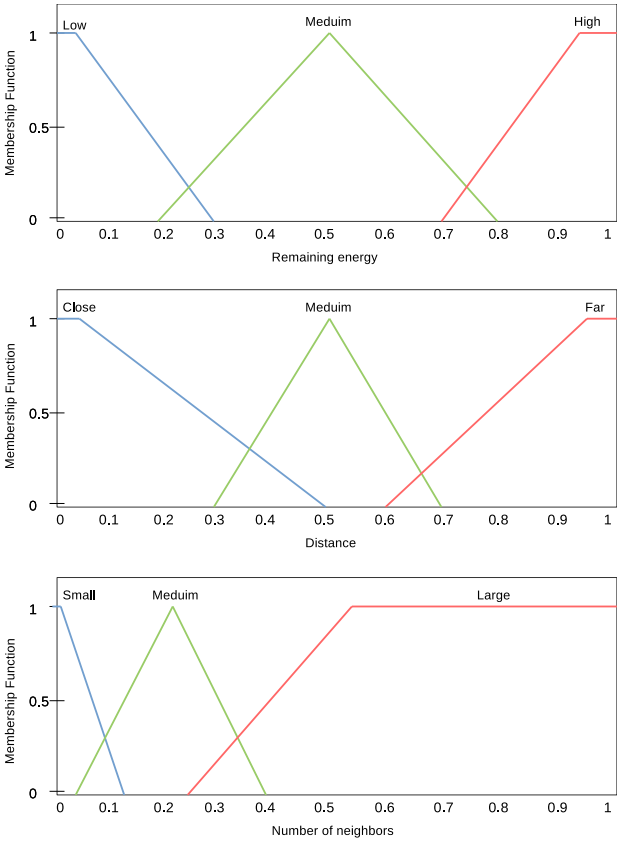

FIGURE 4. Memberships of the input parameters.

$$A = x, \mu_A(x) | x \in X \tag{3}$$

where $\mu_A(x)$ is a membership function (MF) of x in A. This function maps each element of X into a value between 0 and 1. The MF value is the intersection point of the value of the input parameters with the MF degree [25]. In this paper, in order to allow the fuzzy sets inputs (input parameters) to be applicable for any network configuration, we represents the domain of the input parameters to be a value between 0 and 1.

- **Rule evaluation:** In this phase, the membership values are interpreted by applying the IF-THEN rules, which are expressed as a collection of IF-THEN statements, to determine the new fuzzy output set. The IF-THEN rules have multiple inputs that are evaluated by the fuzzy (AND) operator. This work uses three input parameters, and each parameter is divided into three levels. Thus, we obtain 27 IF-THEN rules as shown in Table 1.

TABLE 1. Fuzzy IF-THEN rules in FuMAM approach.

Remaining energy	Distance to Source node	Number of neighbors	Probability
Low	Far	Small	V-Low
Low	Far	Medium	V-Low
Low	Far	Large	V-Low
Low	Medium	Small	V-Low
Low	Medium	Medium	V-Low
Low	Medium	Large	Low
Low	Close	Small	V-Low
Low	Close	Medium	Low
Low	Close	Large	L-Low
Medium	Far	Small	V-Low
Medium	Far	Medium	Low
Medium	Far	Large	Low
Medium	Medium	Small	V-Medium
Medium	Medium	Medium	Medium
Medium	Medium	Large	L-Medium
Medium	Close	Small	V-Medium
Medium	Close	Medium	Medium
Medium	Close	Large	L-Medium
High	Far	Small	V-Low
High	Far	Medium	Low
High	Far	Large	L-Low
High	Medium	Small	L-high
High	Medium	Medium	High
High	Medium	Large	High
High	Close	Small	V-High
High	Close	Medium	V-High
High	Close	Large	V-High

- **Fuzzy Inference System and Aggregation:** This phase is used to form inferences, such that the rules are combined to achieve an aggregated fuzzy output. We use the most common fuzzy inference technique, called the Mamdani method [26], due to its simplicity in calculating the output value of each candidate node.
- **Defuzzification:** The defuzzification phase is applied to achieve the crisp output value after the aggregation of the results obtained from each rule. A defuzzification method called center of area (CoA) [27] is used to obtain the crisp output value, which can be computed using Equation (4).

$$Output = \frac{\int \mu(x) \times x dx}{\int \mu(x) dx} \tag{4}$$

where $\mu(x)$ is the aggregated MF of the fuzzy set, and x is the output variable. To acquire flexible output results, the linguistic variables of the probability value are divided into nine levels as shown in Figure 5. For a candidate node, a high output value increases its probability of being selected as the next MA's migration hop.

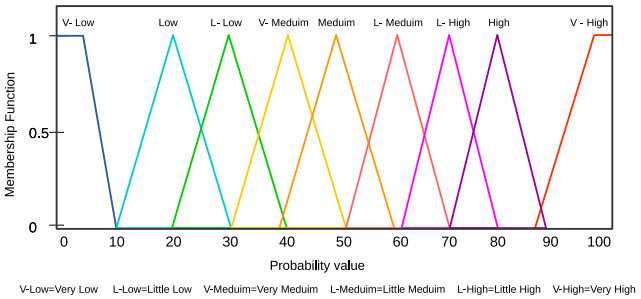

FIGURE 5. Membership of the probability value.

IV. PERFORMANCE EVALUATION

As stated in Section 2, the MIP approach consists of two or more SIPs working concurrently to visit groups of source nodes. Most of the proposed MIP approaches adopt the proposed SIP algorithms (LCF and GCF) for the itinerary of each individual MA. In this section, the FuMAM approach is evaluated relative to GIGM-MIP [16], the well-known CL-MIP approach [6] and our previous work (spawn multi-mobile agent itinerary planning (SMIP) approach [23]). We test the GIGM approach with two MA's itinerary algorithms (LCF and GCF) to compare their performance with the proposed FuMAM. For the other compared approaches, the LCF algorithm is adopted to determine an MA's itinerary. Simulations are performed using MATLAB R2014a. The itineraries of MAs are statically predetermined at the sink node before the MAs are dispatched to the network. In FuMAM, the LCF algorithm is adopted to determine the order of the visited source nodes, whereas the visiting order of the intermediate nodes between each two source nodes is determined by FLS. The sink node is positioned at the center of the network and is the starting and ending points of each MA's itinerary. The same network model in [6], [7], [16], and [28] is adopted, and the energy consumption model in [7] and [13] is used. The nodes are static, dense, and uniformly deployed in a large-scale network (800 nodes) to validate the scaling of FuMAM. All nodes have the same initial energy, and each node has at least one neighbor node with a transmission range of 60 m. Table 2 shows the simulation parameters used in the experiments.

TABLE 2. Simulation parameters.

Network Parameters	Value
Size of network	1000 m × 500 m
Number of deployed nodes	800
Node initial energy	2 J
Number of source nodes	10 - 80
Transmission range	60 m
Raw data size	2048 bits
MA Parameters	**Value**
MA processing code	1024 bits
MA accessing delay	10 ms
Raw data reduction ratio	0.8
Aggregation ratio	0.9
Data processing rate	50 Mbps

All the compared approaches are evaluated using five performance metrics, namely, MA's round-trip rate, network

lifetime, task energy consumption, task duration, and energy distribution usage. For each performance metric, a random number of source nodes from 10 to 80 is selected to study their impact on the performance of FuMAM. We plot the average of 100 simulation runs for each data point in each presented figure.

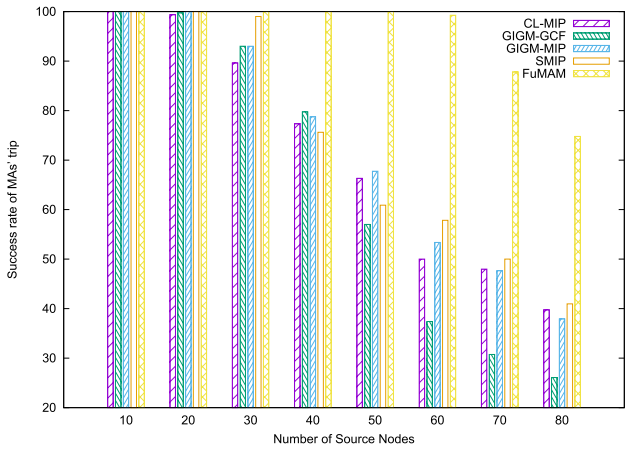

FIGURE 6. Success rate of MA's round-trip.

Figure 6 shows the impact of the varying number of source nodes on the success rate of MAs' round-trip (S_{rate}). The success rate of MAs' round-trip represents the ratio of the number of MAs that are successfully received at the sink after migration to the total number of MAs dispatched by the sink [29]. This metric can be calculated using Equation (5).

$$S_{rate} = \frac{N_{receivedMAs}}{N_{dispatchedMAs}} \times 100 \tag{5}$$

where $N_{receivedMAs}$ represents the number of MAs that are received at the sink after migration. $N_{dispatchedMAs}$ represents the total number of the MAs that are dispatched by the sink before migration. As shown in Figure 6, FuMAM outperforms the compared schemes in terms of the success rate of MAs' round-trip. In this scenario, we increase the number of source nodes to 80 to ascertain the effectiveness of the proposed approach. FuMAM enhances the success rate of an MA's round-trip by 26.84%, 22.92%, 29.76%, and 22.19% compared with CL-MIP, GIGM-MIP, GIGM-GCF, and SMIP, respectively. These outcomes are obtained because FuMAM considers the remaining energy of candidate nodes. That is, the node with the highest residual energy is selected as the next MA's migration hop. By contrast, the compared approaches have lower success rates for an MA's round-trip because distance is the only parameter they consider in constructing the MA itinerary. Consequently, some MAs fail to complete their round-trip because certain nodes with low residual energy are selected. Figure 7 illustrates network lifetime with the impact of varying the number of source nodes. Network lifetime is defined in this work as the duration when the first sensor node in the network dies because of the depletion of its energy [30].

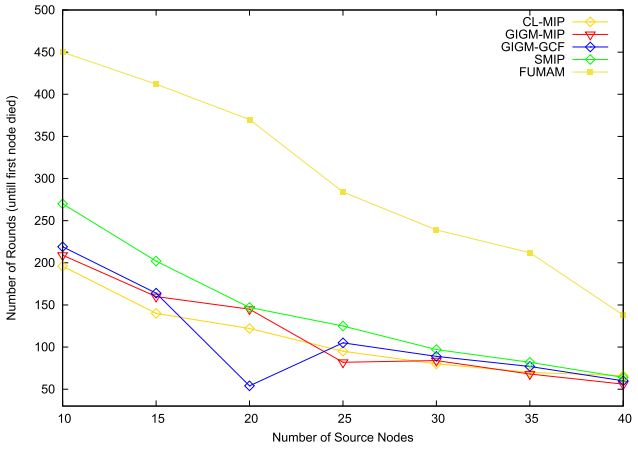

FIGURE 7. Network lifetime.

As shown in Figure 7, network lifetime decreases as the number of source nodes increases. This trend occurs because an increase in the number of MA hops generates a lengthy MA itinerary and results in high energy consumption. The network lifetime of FuMAM is reduced at a slow rate compared with the rate of other approaches. Moreover, the network lifetime of FuMAM significantly increases by approximately 76.86%, 61.8%, 63.51%, and 53.11% relative to CL-MIP, GIGM-MIP, GIGM-GCF, and SMIP, respectively. Such improvement is achieved due to the adoption of FLS during the determination of an MA's itinerary. In this manner, the node with the lower remaining energy among its neighbors will not be selected as the next hop for MA migration. This method can also be used to achieve an improved distribution of energy usage among nodes.

To evaluate the performance of FuMAM in terms of energy distribution usage among nodes, we examine its energy-balancing feature and observe its impact on network performance. In this evaluation, 100 data gathering rounds are simulated with 80 randomly selected source nodes from the entire network. Before the start of the first round, each node begins with an initial energy of 2 J. Then, we evaluate the residual energy of each node at the end of the simulated rounds. The cumulative distribution function plot of the remaining energy distribution is shown in Figure 8. This figure demonstrates that 82% of the deployed nodes in the network has a remaining energy of less than or equal to 1 J with FuMAM, whereas for the other algorithms (SMIP, CL-MIP, GIGM-GCF, and GIGM-LCF), 42%, 27%, 25%, and 22%, of the deployed nodes in the network has a residual energy of less than or equal to 1 J, respectively. These outcomes indicate that FuMAM outperforms the compared algorithms in terms of distribution of energy usage among nodes. Moreover, the plot also shows that the maximum remaining energy in the network is 1.4 with FuMAM, whereas for the other algorithms, the maximum remaining energy in the network is approximately 2 J. This finding is attributed to the involvement of more nodes in forwarding MAs in

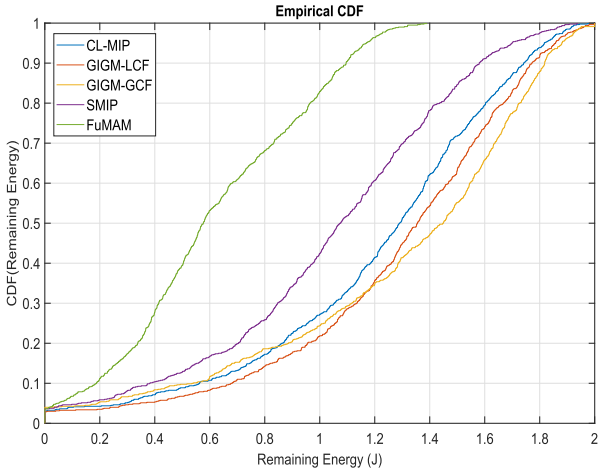

FIGURE 8. Energy consumption distribution (80 source nodes).

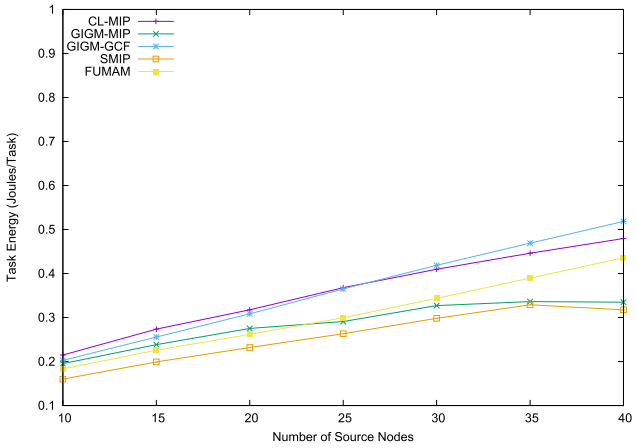

FIGURE 9. Task energy consumption.

FuMAM, which indicates the energy-balancing feature of this algorithm. These results prove that some of the nodes in the network are not involved in forwarding MAs in the other algorithms. In summary, FuMAM effectively increases network lifetime because it always consider the maximum remaining energy of candidates when determining an MA's itinerary. Figure 9 shows the task energy consumption with the impact of varying source nodes. Task energy consumption is the total energy spent for transmitting, receiving, and exchanging control messages to complete the data gathering process from all the source nodes. Such consumption is the accumulated migration energy cost for all the MAs' itineraries [31]. Therefore, the total energy cost of all distributed MAs' itineraries can be calculated using Equation (6).

$$Ctotal = \sum_{t=1}^{|I|} IC^t \tag{6}$$

where IC^t is the energy cost of itinerary I^t covered by an MA, and IC^t can be simplified into Equation (7)

$$IC^t = \sum_{j=1}^{|I^t|} (jdf + pc)c_{i,j} \tag{7}$$

where $|I^t|$ represents the number of sensor nodes in itinerary I^t visited by an MA, j is the visited sensor node, jdf refers to the data aggregated by an MA at sensor node j, f is the aggregation ratio, pc is the initial MA's size (processing code plus MA packet header), and $c_{i,j}$ is the energy consumed by an MA when migrating from sensor nodes i to j. Notably, j can act as a source node (which has data to be collected by the MA) or as an intermediate node.

Figure 9 illustrates that when the number of source nodes is less than 25, the proposed FuMAM approach consumes nearly the same amount of energy as that in the GIGM-MIP algorithm because the nodes still have sufficient remaining energy. As the number of source nodes increases, FuMAM begins to consume more energy relative to the other approaches. This increase in energy consumption is attributed to FuMAM constructing a large number of hops during the determination of an MA's itinerary. This condition occurs because FuMAM selects the candidate node with the maximum remaining energy and a large number of neighbors as the next MA's hop. Although FuMAM consumes more energy than the other approaches when the number of source nodes increases, it performs well when the number of source nodes is small.

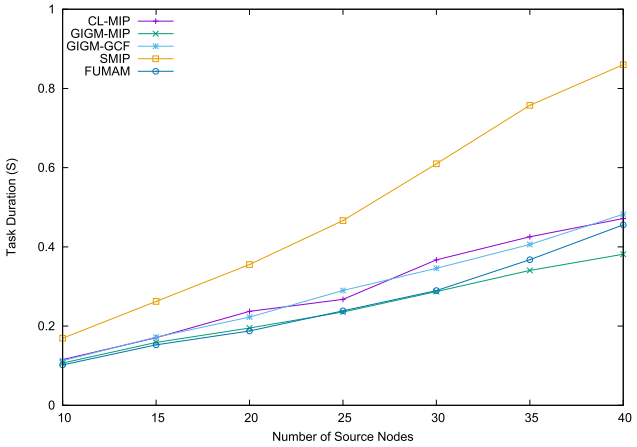

FIGURE 10. Task duration.

Task duration in MA-based data gathering is represented by the round-trip time for one particular data gathering task. In the SIP approach, where only one MA migrates to the network, task duration indicates the average delay from the time when the MA is dispatched to the time it returns to the sink. In MIP, where multiple MAs migrate to the network in parallel, task duration is the delay of the last MA that returns to the sink. Figure 10 illustrates the task duration of FuMAM with the impact of source nodes. Compared with the other approaches, FuMAM has a shorter task duration when the number of source nodes is less than 25. This finding is attributed to FuMAM considering the distance parameter as one of the FLS inputs when calculating the itinerary of an MA. Furthermore, FuMAM selects the node with the shortest distance for the next MA's hop. When the number of source nodes increases, FuMAM begins to consume more time than

GIGM-MIP because FuMAM considers the residual energy of the nodes along with distance. If the distance of a candidate node is short and its residual energy is low, FuMAM selects another candidate node with sufficient residual energy even if its distance is far. Therefore, FuMAM builds a long MA itinerary when the remaining energy of the candidate node is low, which can lead to an increase in the number of MA hops, and consequently, in task duration.

V. CONCLUSION

Determining an appropriate itinerary for each MA in MIP is considered a critical issue because it directly impacts the overall performance of data gathering in WSNs. In a static itinerary, most of the well-known algorithms consider the distance between nodes as the main parameter for determining MA itinerary. Other parameters, such as remaining energy and number of neighbors, are disregarded. This situation can affect MA migration and result in an unsuccessful MA's round-trip, which consequently decreases network lifetime. In this work, the FuMAM approach is proposed to determine an appropriate MA itinerary. FuMAM considers three parameters: distance, remaining energy, and number of neighbors. Extensive simulation experiments are conducted to evaluate the performance of FuMAM. The simulation results show that FuMAM outperforms the compared MIP approaches in terms of the success rate of MA's round-trip. In addition, FuMAM improves network lifetime relative to the other MIP approaches. As a future work, other input parameters to FLS can be considered to evaluate the performance of FuMAM. The remaining energy of the candidate's neighbors can be added as an input to FLS in the proposed FuMAM. This inclusion can enhance energy balancing among nodes for each computed itinerary. However, an increase in the number of input parameters will increase the complexity of the rule base. Therefore, a method such as HFS can be used to reduce the size of the rule base.

REFERENCES

[1] H. Qi, Y. Xu, and X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1172–1183, Aug. 2003.

[2] H. Qi and F. Wang, "Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks," *Proc. IEEE*, vol. 18, pp. 147–153, Jul. 2001.

[3] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V. C. M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *EURASIP J. Appl. Signal Process.*, vol. 2007, p. 036871, Dec. 2007.

[4] Q. Wu *et al.*, "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 6, pp. 740–753, Jun. 2004.

[5] M. Chen, V. Leung, S. Mao, T. Kwon, and M. Li, "Energy-efficient itinerary planning for mobile agents in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2009, pp. 1–5.

[6] M. Chen, S. Gonzalez, Y. Zhang, and V. C. Leung, "Multi-agent itinerary planning for wireless sensor networks," in *Proc. Int. Conf. Heterogeneous Netw. Quality, Rel., Secur. Robustness*, 2009, pp. 584–597.

[7] W. Cai, M. Chen, T. Hara, L. Shu, and T. Kwon, "A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks," *Mobile Netw. Appl.*, vol. 16, no. 6, pp. 782–793, 2011.

[8] M. Bendjima and M. Feham, "Multi mobile agent itinerary for wireless sensor networks," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 1, no. 1, pp. 1–6, 2012.

[9] H. Q. Qadori, Z. A. Zulkarnain, Z. M. Hanapi, and S. Subramaniam, "Multi-mobile agent itinerary planning algorithms for data gathering in wireless sensor networks: A review paper," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 1, pp. 1–13, 2017.

[10] M. Chen, S. Gonzalez, and V. C. M. Leung, "Applications and design issues for mobile agents in wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 20–26, Dec. 2007.

[11] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Mobile agent middleware for autonomic data fusion in wireless sensor networks," in *Autonomic Computing and Networking*. Boston, MA, USA: Springer, 2009, pp. 57–81.

[12] P. K. Biswas, H. Qi, and Y. Xu, "Mobile-agent-based collaborative sensor fusion," *Inf. Fusion*, vol. 9, no. 3, pp. 399–411, 2008.

[13] M. Chen, L. T. Yang, T. Kwon, L. Zhou, and M. Jo, "Itinerary planning for energy-efficient agent communications in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3290–3299, Sep. 2011.

[14] D. Gavalas, A. Mpitziopoulos, G. Pantziou, and C. Konstantopoulos, "An approach for near-optimal distributed data fusion in wireless sensor networks," *Wireless Netw.*, vol. 16, no. 5, pp. 1407–1425, 2010.

[15] M. Chen, W. Cai, S. Gonzalez, and V. C. M. Leung, "Balanced itinerary planning for multiple mobile agents in wireless sensor networks," in *Ad Hoc Networks*. Berlin, Germany: Springer, 2010, pp. 416–428.

[16] I. Aloui, O. Kazar, L. Kahloul, and S. Servigne, "A new itinerary planning approach among multiple mobile agents in wireless sensor networks (WSN) to reduce energy consumption," *Int. J. Commun. Netw. Inf. Secur.*, vol. 7, no. 2, pp. 116–122, 2015.

[17] Y.-C. Chou and M. Nakajima, "A clonal selection algorithm for energy-efficient mobile agent itinerary planning in wireless sensor networks," in *Mobile Networks and Applications*. New York, NY, USA: Springer, 2017, pp. 1–14.

[18] Y. Zhang, J. Wang, D. Han, H. Wu, and R. Zhou, "Fuzzy-logic based distributed energy-efficient clustering algorithm for wireless sensor networks," *Sensors*, vol. 17, no. 7, p. 1554, 2017.

[19] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.

[20] A. K. Mishra, R. Kumar, V. Kumar, and J. Singh, "A grid-based approach to prolong lifetime of WSNs using fuzzy logic," in *Advances in Computational Intelligence*. Singapore: Springer, 2017, pp. 11–22.

[21] I. A. Umar, Z. M. Hanapi, A. Sali, and Z. A. Zulkarnain, "FuGeF: A resource bound secure forwarding protocol for wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 943, 2016.

[22] J. Zajaczkowski and B. Verma, "Selection and impact of different topologies in multi-layered hierarchical fuzzy systems," *Appl. Intell.*, vol. 36, no. 3, pp. 564–584, 2012.

[23] H. Q. Qadori, Z. A. Zulkarnain, Z. M. Hanapi, and S. Subramaniam, "A spawn mobile agent itinerary planning approach for energy-efficient data gathering in wireless sensor networks," *Sensors*, vol. 17, no. 6, p. 1280, 2017.

[24] M. S. Gharajeh and S. Khanmohammadi, "Static three-dimensional fuzzy routing based on the receiving probability in wireless sensor networks," *Computers*, vol. 2, no. 4, pp. 152–176, 2013.

[25] R. Mhemed, N. Aslam, W. Phillips, and F. Comeau, "An energy efficient fuzzy logic cluster formation protocol in wireless sensor networks," *Proc. Comput. Sci.*, vol. 10, pp. 255–262, Jan. 2012.

[26] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Studies*, vol. 7, pp. 1–13, Jan. 1975.

[27] N. Nokhanji, Z. M. Hanapi, S. Subramaniam, and M. A. Mohamed, "An energy aware distributed clustering algorithm using fuzzy logic for wireless sensor networks with non-uniform node distribution," *Wireless Pers. Commun.*, vol. 84, no. 1, pp. 395–419, 2015.

[28] J. Wang, Y. Zhang, Z. Cheng, and X. Zhu, "EMIP: Energy-efficient itinerary planning for multiple mobile agents in wireless sensor network," *Telecommun. Syst.*, vol. 62, no. 1, pp. 93–100, 2016.

[29] G. P. Gupta, M. Misra, and K. Garg, "Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 41, pp. 300–311, May 2014.

[30] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 828–854, 2nd Quart., 2017.

[31] X. Wang, M. Chen, T. Kwon, and H. C. Chao, "Multiple mobile agents' itinerary planning in wireless sensor networks: Survey and evaluation," *IET Commun.*, vol. 5, no. 12, pp. 1769–1776, 2011.

HUTHIAFA Q. QADORI received the B.S. degree in computer science from the Faculty of Computer Sciences and Information Technology, University of Anbar, Iraq, in 2006, and the M.Sc. degree in information technology from University Tenaga National, Malaysia, in 2012. He is currently pursuing the Ph.D. degree with the Department of Wireless and Communication Networks, Universiti Putra Malaysia. His research interests include computer networks, wireless sensor networks, routing algorithms, and Internet of Things.

ZURIATI AHMAD ZUKARNAIN received the bachelor's degree in physics and education and the M.Sc. degree from Universiti Putra Malaysia (UPM) in 1997 and 2000, respectively, and the Ph.D. degree in quantum computing and communication from the University of Bradford, U.K., in 2005. She was appointed as the Head of the Department Communication Technology and Networks from 2006 to 2011. She has been an Academic Staff with the Faculty of Computer Science and Information Technology, UPM, since 2001, where she taught several courses for undergraduate students, such as data communication and networks, distributed system, mobile and wireless, network security, computer architecture, and assembly language. For postgraduate students, she taught few courses, such as advanced distributed and research method. Her areas of specialization are as computer networks, distributed system, mobile and wireless, network security, quantum computing, and quantum cryptography. She was the Head of the Section of High Performance Computing, Institute of Mathematical Research, UPM, from 2012 to 2015. She is a member of the IEEE Computer Society.

ZURINA MOHD HANAPI received the B.Sc. degree in computer and electronic system engineering from Strathclyde University in 1999, the M.Sc. degree in computer and communication systems engineering from Universiti Putra Malaysia (UPM) in 2004, and the Ph.D. degree in electrical electronic and system engineering from Universiti kebangsaan Malaysia in 2011. She is currently an Associate Professor with the Faculty of Computer Science and information Technology, UPM, where she is also a Lecturer. She is the leader of some research projects. She has published many conference and journal papers. Her research interests include routing, wireless sensor network, wireless communication, distributed computing, network security, cryptography, and intelligent systems. She has received an Excellence Teaching Awards in 2005, 2006, and 2012, the Silver Medal in 2004, and the Bronze Medal in 2012.

SHAMALA SUBRAMANIAM received the B.S. degree in computer science and the M.S. and Ph.D. degrees from Universiti Putra Malaysia in 1996, 1999, and 2002, respectively. Her research interests include computer networks, simulation and modeling, scheduling, and real-time system.