

RESEARCH

Open Access



Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks

Mohamed El Fissaoui^{1*}, Abderrahim Beni-hssane¹ and Mostafa Saadi²

Abstract

Mobile agent (MA)-based wireless sensor networks present a good alternative to the traditional client/server paradigm. Instead of sending the data gathered by each node to the sink as in client/server, MAs migrate to the sensor nodes (SNs) to collect data, thus reducing energy consumption and bandwidth usage. For MAs, to migrate among SNs, an itinerary should be planned before the migration. Many approaches have been proposed to solve the problem of itinerary planning for MAs, but all of these approaches are based on the assumption that MAs visit all SNs. This assumption, however, is inefficient because of the increasing size of the MAs after visiting each node. Also, in case of node(s) failure, as it is often the case in WSNs, the MAs may not be able to migrate among SNs. None of the proposed approaches takes into consideration the problem of fault tolerance. In this paper, we propose multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks (MAEF) to plan itineraries for MAs. This can be achieved by grouping nodes in clusters and planning itineraries efficiently among cluster heads (CHs) only. What is more, an alternative itinerary is planned in case of node(s) failure. The simulation result clearly shows that our novel approach performs better than the existing ones.

Keywords: Wireless sensor networks, Mobile agent, Client-server, Data aggregation, Itinerary planning, Minimum spanning tree

1 Introduction

The recent technological advances in wireless communications and microelectromechanical systems (MEMS) have made it possible to develop a tiny, low-power, and low-cost sensor node (SN). Wireless sensor networks (WSNs) [1] consist of a large number of SNs densely deployed in the monitoring area with sensing, wireless communication, and computing capabilities. SNs tend to gather data from the surrounding environment and send it back to the sink.

In WSNs, the most commonly used computing paradigm is client-server (CS), where each SN communicates its collected data to the sink via a multi-hop route [2–6]. But this traditional paradigm suffers from significant drawbacks, such as lack of scalability when the network size increases. Additionally, also the amount of

data that should be processed to the sink become bigger and surpasses the capacity that the wireless link could handle given the low bandwidth of the wireless link used in WSNs. In the last few years, an efficient paradigm has emerged and been adopted by researchers as an alternative to the traditional CS paradigm, it is called mobile agent paradigm (MA) [7–11]. In this paradigm, instead of gathering data and sending it to the sink by SNs as is the case in traditional CS [11, 12], the mobile code migrates to SNs to collect data.

MA is a special type of software entity that migrates among SNs to gather data [8, 13]. In comparison with CS paradigm, MA paradigm has many features [4, 14–16] that makes it more suitable for WSNs. It is important to note that the itinerary planning for MA is the most challenging issue with this paradigm. MAs' efficiency-based data aggregation depends on the itinerary planning. An itinerary is the route that the MA follows during its migration among SNs [17]. It has been proved that planning itinerary for MA is an NP-hard problem [18, 19] and one

*Correspondence: Elfissaoui.m@ucd.ac.ma

¹Computer Science Department, Science Faculty, Chouaib Doukkali University, El Jadida, Morocco

Full list of author information is available at the end of the article

of the most challenging issues with this paradigm. Therefore, to plan itineraries for MAs, it is essential to consider the following three key elements :

- SNs grouping is a set of SNs that should be visited by each MA.
- The itinerary planning for MA refers to the order in which these SNs should be visited by the MA.
- The number of MAs is the number of MAs to be dispatched to gather data.

The itinerary planning for MA could be static or dynamic. In static itinerary planning (SIP), the itinerary of the MA is computed at the sink level, then the MA migrates among SNs based on the pre-computed itinerary presented by the sink. The dynamic itinerary planning (DIP) allows the MA to quickly decide on the fly the next destination node. The static approaches are more suitable for monitoring applications in which the collected information are gathered and communicated back to the sink. On the other hand, the dynamic approaches are often used in tracking applications. In addition, due to the nature of WSNs that are prone to failure [20], the act of planning the MAs' itinerary without taking into consideration the faulty nodes (FNs) could pose a problem in case of node(s) failure. It may prevent the MA from continuing its migration among the SNs. In the present paper, in addition to the act of planning itineraries for MAs among CHs, we propose a fault tolerance based on alternative itinerary planning in case of the failure of the first itinerary.

Taking into consideration the aforementioned issues, the main contributions of this paper are as follows:

- SN grouping in clusters
- Itinerary planning among CHs
- MA migration
- Fault tolerance based on alternative itinerary planning
- Evaluation of the proposed approach and comparing it with the other existing approaches using a Castalia simulator [9]

The rest of this paper is organized as follows: in Section 2, we briefly describe the related work. In Section 3, we introduce the network model. In Section 4, we present our proposed approach. In Section 5, we present a fault tolerance based on alternative itinerary planning. Section 6 is devoted to results and discussion. We summarize and conclude the main points discussed in this paper in Section 7.

2 Related work

There has been, in the last few years, a growing interest in the MA paradigm in WSNs. Consequently, a plethora of MA-based approaches have been proposed. The Local Closest First (LCF) and Global Closest First (GCF) are the

first two heuristics algorithms which have been proposed by Hairong et al. [10] to plan the itinerary for the MA among SNs in order to perform data collection-related tasks. In LCF, the MA migrates to the nearest node to the sink, then it searches for the closest node to its current location. In GCF, the sink dispatches the MA to the next node that is nearest to the sink and it migrates again to the nearest node to the sink instead of its current location as in LCF. In LCF algorithm, the last SNs in the MAs' itinerary are the SNs with longest distance from the sink, because LCF search for the next destination node among SNs based on its current location, not by looking at the global network distance matrix. On the other side, repetitive MA oscillations produced by GCF algorithm around the sink produce a long itinerary and poor performance [10]. Another approach-based directed diffusion has been proposed in [21]; the authors propose the mobile agent-based directed diffusion (MADD). MADD is quite similar to LCF, but instead of selecting the nearest node to the sink as the first node as is the case in LCF, MADD selects the farthest SN from the sink as the first node.

In [22], authors propose a better approach called itinerary energy minimum for first-source-selection (IEMF) algorithm, and the itinerary energy minimum algorithm (IEMA), the iterative version of IEMF. IEMF indicates the importance of choosing the first SN in the itinerary, IEMF estimates the energy costs of all alternatives of the first node and selects the nodes with the minimum energy cost. In each iteration, IEMA choose from the rest of SNs the SN with the optimal energy cost as the next destination . In [18], authors have proposed a genetic algorithm (GA) to solve the problem of the itinerary planning. The GA algorithm provides superior performance than the first proposed LCF and GCF algorithms in terms of energy cost, but it implies a time-expensive itinerary calculation, which is not tolerable for time-critical applications.

Another algorithm has been proposed in [17, 23] named single dynamic agent migration algorithm for a target tracking application. In this algorithm, a MA migrates to a SN that can get more accurate information about the target location, thus consuming less energy. The selection of the next destination node is based on cost function , which includes three components: information gain, energy consumption, and the remaining energy of nodes. Once the MA accumulates sufficient information so that the accuracy of the estimation meets the desired level, the MA will end the migration and migrate back to the sink [17, 23]. This algorithm is time-expensive and may face difficulties to migrate back to the sink without additional forwarding information.

In [24], authors propose a software agent-based directed diffusion, where the order of visiting SNs is determined at the sink node. This method takes the routing cost and

the remaining energy of a node for selecting the next node to be visited by MA. A dynamic algorithm has been proposed in [25]. In this algorithm, a cost function is used by MA to decide on the fly the next SN to visit. Authors compare their proposed algorithm just to LCF/GCF.

Another routing strategy based on MA has been proposed in [26]. To plan the itinerary for the MA, authors use a cost function and an improved colony algorithm; also, a simple scheme that turns off redundant nodes according to the measurement requirement is designed to improve the energy efficacy.

Single itinerary planning (SIP) suffers from lack of scalability. With the growing size of the network, SIP becomes inefficient and multiple itinerary planning (MIP) approaches are proposed to solve the problem.

For MIP for MAs, many approaches have been proposed. Near-optimal itinerary design (NOID) algorithm [27] adapts a method called the Esau-Williams heuristic originally designed for network design problems for the constrained minimum spanning tree (CMST) issue to the specific requirements of WSNs. In NOID, multiple itinerary planning for MAs is proposed where each MA visits a group of SNs. NOID suffers from low working speed and high computational complexity.

In [28], Chin et al. have proposed directional source grouping (DSG) algorithm to find near optimal itineraries for multiple MAs. This algorithm uses a disk around the sink by adjusting its radius and iteratively partitions a directional sector zone where the SNs are included in an itinerary. The radius of the circle is set to the maximum transmission range of a single SN. Every SN that lies in the circle is used as the first node of an itinerary. The angle size of the directional sector zone controls the size and the set of SNs that will be included in the itinerary. Another algorithm has been proposed in [29] to plan itineraries for MAs. Then the itinerary for each of the mobile agents can be planned by any SSIP algorithms.

Another data gathering system in WSNs has been proposed in [30] with mobile agents and mobile sink for agriculture application, to plan the itinerary, a dynamic itinerary planning approach DPMA is used. Energy-efficient itinerary planning for multiple mobile agent EMIP algorithm has been proposed in [31] that iteratively partitions a directional sector zone where source nodes are included in the itinerary; the length of the itinerary is controlled by the angle of the directional sector zone. Aloui et al. [32] propose a MIP solution that is based on two factors which have a direct impact on energy consumption, geographical distance, and data size. The number of required mobile agents and the SN grouping is based on these two factors. In [33], authors propose a mobile agent routing protocol called zone-based mobile agent in which a bottom-up mobile agent migration is

used and the migration of the MAs start from the center of event regions toward the sink. In [34], authors have proposed a mechanism for scheduling mobile sensors in time-sensitive WSN applications. To schedule mobile sensors, authors have applied techniques derived from ant colony optimization and genetic algorithm. The network of static sensors are derived into small regions where a mobile sensor assigned to each region.

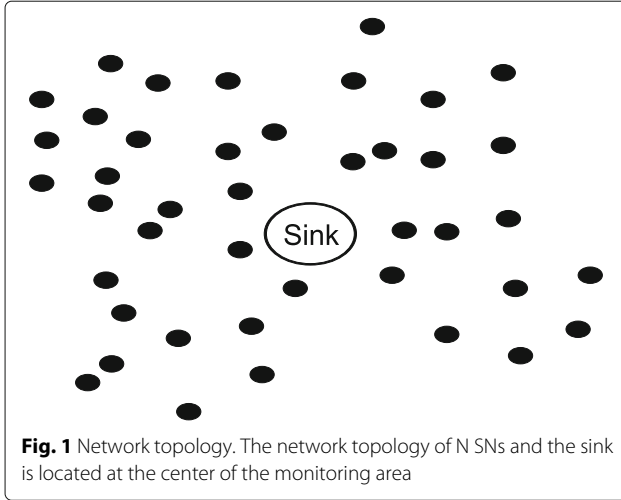
The MA paradigm has been proven to be efficient in terms of energy consumption and execution time, but the most challenging issue with this paradigm is the itinerary planning for MA. Many solutions have been proposed to solve this problem as we have surveyed in this section, but the most proposed multi-agent itinerary algorithms are either time-consuming or too complicated in practice, also these approaches are based on the assumption that MAs visit all SNs and no nodes' failure takes place in the network. Herein, we propose multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks (MAEF) to plan itineraries for MAs among CHs, and then plan alternative itineraries in case of the failure of the first one. In our proposed approach, we use a clustering method to group SNs in clusters and select a CH for each cluster for data aggregation, then plan itineraries only among CHs using a minimum spanning tree. In the existing proposed algorithms, the MA visits all SNs in the network, but in our proposed approach, the MAs visit only the CHs for data gathering. Furthermore, in case of node(s) failure, the previous proposed algorithms did not take into consideration fault tolerance problem. Therefore, a fault tolerance based on fault tolerance is proposed here.

3 Network model

As a network model adopted for this work, we use a sensor network composed of N SNs distributed randomly in 500×500 square area. The SNs and the sink have the same maximum transmission range, but in terms of battery power and computational capabilities, the sink is more powerful than the other SNs. We use the network model as shown in Fig. 1. The sink is located at the center of monitoring area. We suppose that the sink has all the required information about SNs, such as location coordinates, either by using localization algorithms [35–38] or GPS device; also, all SNs have the same amount of energy.

4 The proposed approach

In this section, we present our multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks MAEF that consists of three phases: (1) CH selection and cluster construction, (2) CH-based itinerary planning, (3) MA migration and data collection.



4.1 CHs selection and cluster construction

In this phase, the basic idea is to distribute the density impact factor of each node to other SNs in the network. Let i denote the number of SNs. Each SN will receive $i - 1$ impact factors from other SNs and one from itself. After calculating the accumulated impact factor, the SN with the highest accumulated impact factor will be selected as CH. We use the following equation to calculate the density impact factor [29]:

$$I_{jk} = e^{\frac{-(H_k^j - 2)^2}{2\sigma^2}} \quad (1)$$

where σ is a parameter used to determine how strong SN impacts each other. H_k^j is the hop count estimation between node j and node k and is calculated by the following equation:

$$H_k^j = \frac{d(k-1, k)}{R_{\max}} \quad (2)$$

R_{\max} is the maximum transmission range of each SN.

In Algorithm 1, we calculate the impact factor I_i of each SN in the network using the Eq. 1, then calculate the impact factor of each SN in relation to all nodes in the networks using the same equation. After that, we select the SN with the highest accumulated impact factor $I_{\text{accumulated}}$ as CH, then we include all SNs with a distance less than its maximum transmission range ($(d_{CH}, i) < R_{\max}$) in the cluster and take off these nodes from the remaining SNs V_n , then add the selected CHs to the group of CHs for itinerary planning in the second phase. The output of this algorithm is a group of CHs V_{CH} that will be used in Algorithm 2 to plan itineraries and the SNs in each cluster V_{group} .

Algorithm 1 CHs selection and clusters construction

```

1: Input
2:  $V_n$ 
3: for Each source node  $i$  in  $V_n$  do
4:    $I_i \leftarrow 0$ 
5: end for
6: for Each source node  $i$  in  $V_n$  do
7:   for Each source node  $j$  in  $V_n$  do
8:     Calculate  $I_{ij}$  ▷ use (1)
9:      $I_i \leftarrow I_i + I_{ij}$ 
10:  end for
11: end for
12: for Each source node  $k$  in  $V_n$  do
13:   if  $I_{\text{accumulated}} = \min I_i | k \in V_n$  then
14:     Select node  $k$  as CH
15:     for Each node  $k$  in  $V_n$  do
16:       Calculate  $(d_{CH}, i)$  ;
17:       if  $(d_{CH}, i) < R_{\max}$  then
18:          $V_{\text{left}} \leftarrow V_n - i$ 
19:          $V_{\text{group}} \leftarrow V_{\text{group}} + i$ 
20:          $V_{CH} \leftarrow V_{CH} + k$ 
21:         break;
22:       end if
23:     end for
24:   end if
25: end for
26: Output
27:  $V_{\text{group}}$ 
28:  $V_{CH}$ 

```

After calculating the impact factor of all SNs and select the SN with the highest value of the accumulated impact factor as CH, we construct the first cluster by including all SNs that lie in the range R_{\max} of CH in the cluster (Fig. 2a), thus creating the first cluster, then we repeat the same process with the rest of SNs, until there is no node left out of a cluster (Fig. 2b–d). Figure 2 shows SN grouping and CH selection phase.

At first as in Fig. 2a, we use Algorithm 1 to select the first cluster head and include all SNs in its range in the cluster, source nodes included in the cluster would not be included in any other following clusters. Secondly, as in Fig. 2b, we use the same algorithm with the lifted SNs to select the second CH and construct the second cluster. Finally, we repeat the same process in Fig. 2b–d till there is no SN left out of a cluster.

4.2 CH-based itinerary planning

In this phase as shown in Fig. 3, after constructing clusters based on node density and selecting CHs as illustrated in Fig. 2, we choose the nodes in the range of the sink as the first nodes in each branch stemming from the sink. Each

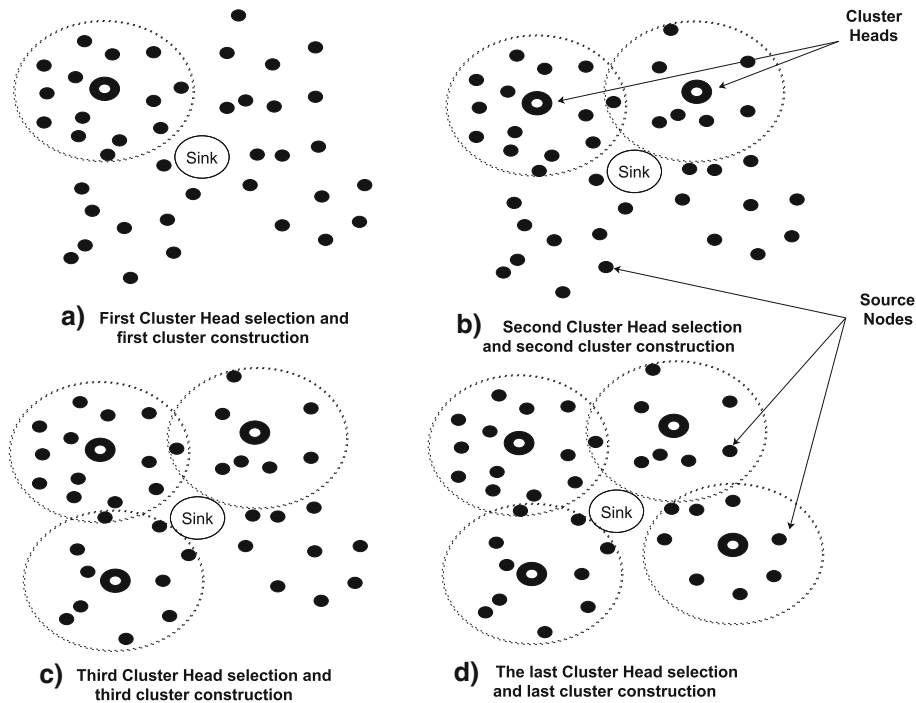


Fig. 2 Phase 1. Cluster head selection and cluster construction phase

branch present an itinerary, thus the number of itinerary will be the same as the number of the branches stemming from the sink. To generate the itinerary, we use Algorithm 2. We plan itineraries for mobile agents among CHs by using a minimum spanning tree (MST).

As we mentioned before, the sink has the coordinates of each node in the network, this way the sink can calculate the weights between CHs. To calculate weights between CHs, we use the following formula in which we use a balancing factor [39]:

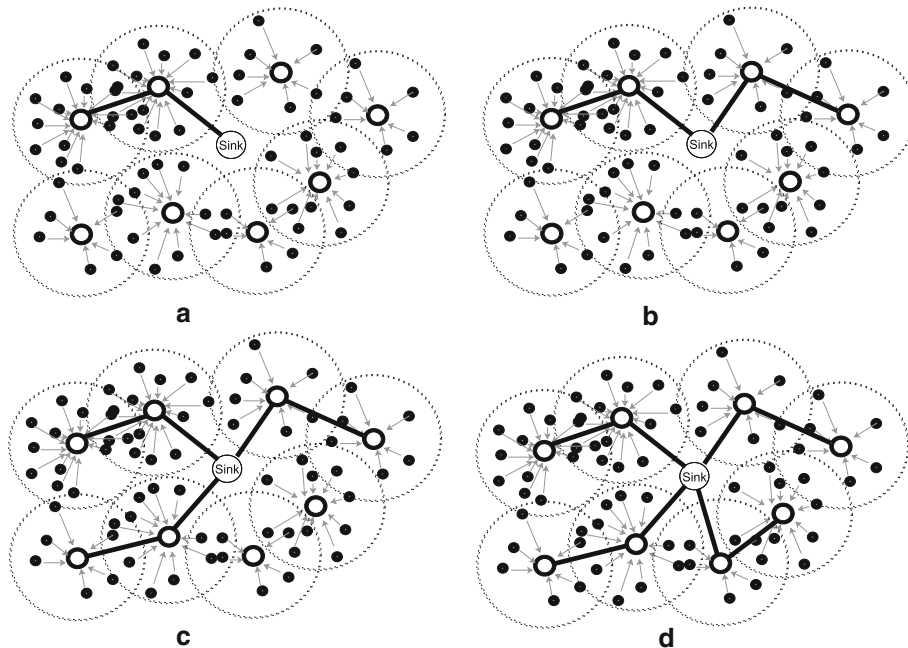


Fig. 3 Phase 2. CH-based itinerary planning phase. **a** The first itinerary planning. **b** The second itinerary planning. **c** The third itinerary planning. **d** The last itinerary planning

$$w = \alpha * H_{CHi}^{CHj} + (1 - \alpha) * (H_s^i + H_s^j) \quad (3)$$

where $0 < \alpha \leq 1$.

To generate a balanced MST branch, we adjust α to a suitable value. H_s^i is the hop count estimation between node i and the sink s and is calculated by the Eq. (2). H_{CHi}^{CHj} is the hop count estimation between CH_i and the CH_j .

In Algorithm 2, we start from the sink, then search for the first CH to visit by using weight function in Eq. (3) that gives the minimum cost to destination; after that, we add the CH with the minimum cost to the itinerary. We select the next CH by using the same process till the last CH in the itinerary. Thus, we construct the first itinerary T_{CH} . We use the same algorithm to construct other itineraries as Fig. 3 shows.

Algorithm 2 CH-based itinerary planning

```

1: Input
2:  $T_{CH} \leftarrow Sink$ 
3:  $V \leftarrow V_{CH}$ 
4: while  $\exists(u \in T_{CH}, v \in V)$  do
5:   find  $\min w(u, v)$ 
6:    $T_{CH} \leftarrow T_{CH} + v$ 
7:    $V \leftarrow V - v$ 
8: end while
9: Output
10:  $T_{CH}$ 

```

Figure 3 shows CH-based itinerary planning phase in details. As Fig. 3a shows, we select a set of CHs that will be included in the itinerary with minimum cost, then we plan itinerary among those CHs. We repeat the same process to construct the second itinerary as Fig. 3b shows. In Fig. 3c, d, the same process is repeated to construct the other itineraries.

In the next phase, multiple mobile agents will be dispatched by the sink in parallel to collect data from CHs.

4.3 MAs migration and data collection

After organizing the network in clusters and planning the itineraries among CHs, the sink dispatches MAs to gather data from the CHs as Fig. 4a shows. At first, when MAs visit the CHs for the first time, it notifies the nodes within the range of the CHs to send the collected data to the selected CHs (Fig. 4a), then when the MAs arrive to the last CH in the itinerary as Fig. 4b shows, it starts collecting the data from the CHs in its way back to the sink. This way, MAs consume less energy and spend less time for data collection tasks.

In Algorithm 3, after planning itineraries in the previous phases, we dispatch a MA to each of itinerary in M_{CH} for data collection

Algorithm 3 MA migration

```

1:  $M_{CH} \leftarrow T_{CH}$ 
2: for Each Itinerary  $k$  in  $M_{CH}$  do
3:   Dispatch MA
4: end for

```

5 Fault tolerance based on alternative itinerary planning

Since the selected CHs are SNs that are prone to failure and malfunction, the fault tolerance property is indispensable in WSNs, especially in non-hostile and harsh environment [20]. In order to bypass node(s) failure that is always the case in WSNs, a re-clustering strategy is proposed. To achieve fault tolerance through re-clustering, we select alternative CHs and plan the itinerary among them using MST. This way, an alternative itineraries for MAs is planned in case the first ones fail.

In our proposed approach, we construct clusters based on SN density by using the impact factor, and select the nodes with the highest impact factor as CH, then plan the itinerary among CHs using a minimum spanning tree. But in case of malfunctioning CH, the MAs may not continue its migration. Therefore, we propose a fault tolerance based on alternative itinerary in case of losing the dispatched MAs due to node(s) failure. We consider the case of partial failure, where the node is able to communicate, but it cannot sense or collect any data.

We use the same impact factor as used to select primary CH_p , but instead of choosing the nodes with the highest impact factor, we choose the nodes with the second highest impact factor as secondary CHs, then plan the itinerary among them by using MST; this way, we plan the alternative itinerary for MA. Figure 5 clearly illustrates the proposed fault tolerance based on alternative itinerary.

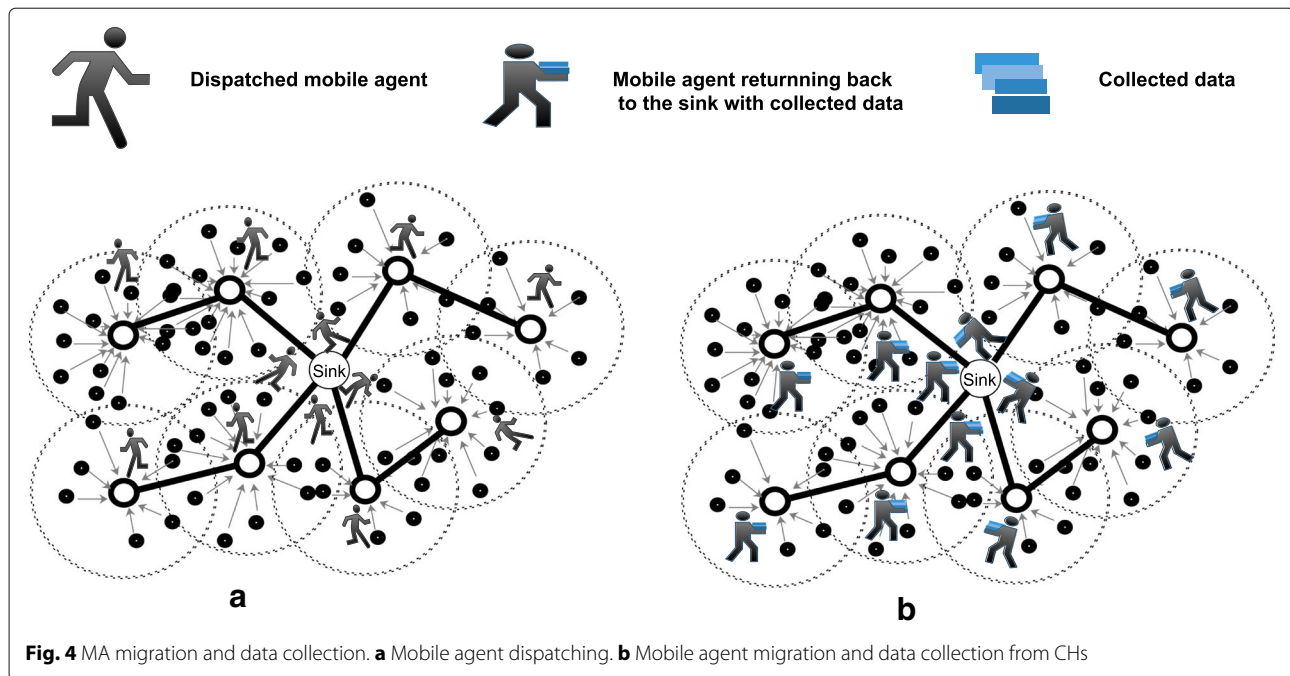
In Algorithm 4, after the selection of secondary cluster heads V_S , we use MST to plan the itineraries among those CHs by choosing the itineraries with minimum weights. The input of this algorithm is T_S that includes only the sink and group of secondary CHs V_S . The

Algorithm 4 Fault tolerance based on alternative itinerary planning

```

1: Input
2:  $T_S \leftarrow Sink$ 
3:  $V \leftarrow V_S$ 
4: while  $\exists(u \in T_S, v \in V)$  do
5:   find  $\min w(u, v)$ 
6:    $T_S \leftarrow T_S + v$ 
7:    $V \leftarrow V - v$ 
8: end while
9: Output
10:  $T_S$ 

```



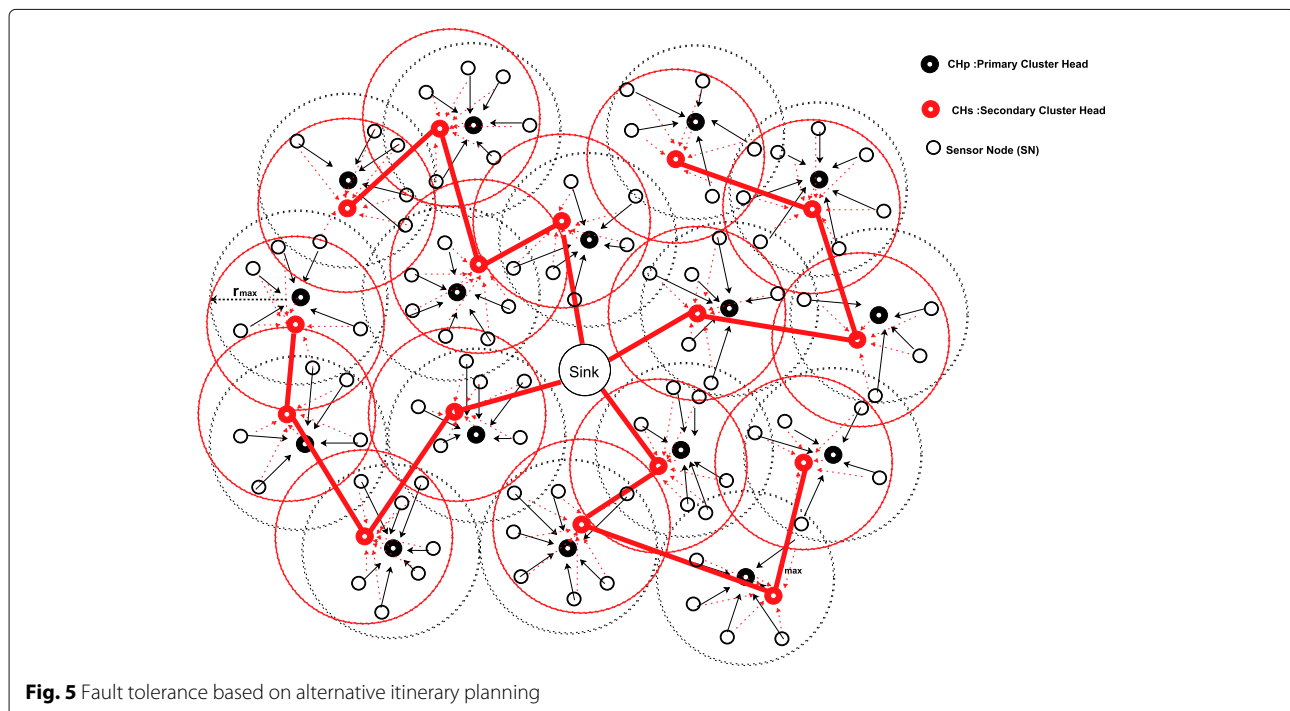
output of the algorithm is the itinerary planned among secondary CHs T_S .

6 Results and discussion

We implemented our proposed approach and compared it with the other existing ones using a Castalia simulator [9]. Castalia is a simulator for WSNs, body area network BANs, and generally, networks of low-power embedded devices. It is based on the OMNeT++ platform and

used by researchers and developers to test protocols in realistic wireless channel and radio models, with a realistic node behavior especially relating to the access of the radio. Castalia can also be used to evaluate different platform characteristics for specific applications, since it is highly parametric, and can simulate a wide range of platforms.

SNs were randomly deployed in square monitoring area of 500×500 m, and varied from 200 to 800 nodes and



the sink is located at the center of monitoring area; also, SNs had the same transmission range and battery power, except for the sink, that has more computation capabilities and battery power.

The rest of the simulation parameters are shown in Table 1.

We use four metrics to evaluate our novel proposed approach:

- Overall energy consumption is the energy consumed by all SN and MA execution.
- Execution time is the required time for MAs to visit all CHs and returning back to the sink.
- Total traveled distance is the total traveled distance by all MAs.
- Dispatched MAs is the number of dispatched MAs to gather data from CHs.

Figure 6 illustrates the comparison of overall energy consumption per round of GCF, IEMF, MADD, DSG, NOID, our proposed approach MAEF, and its alternative A-MAEF with the increasing size of network from 200 to 800 SN. As shown in Fig. 6, the overall energy consumption per round increases as the number of SNs increase. As the figure shows, our proposed protocol consumes less energy than all the other protocols with the increasing size of the network. This is due to the strategy of visiting only CH nodes to gather data already sent by SNs and by using MST to plan itineraries among CHs that are close to each other. IEMF and MADD protocols consume almost the same amount of the energy, with better performance of IEMF. DSG and NOID consume less energy than GCF, IEMF, and MADD; this is due to the multiple MA usage, but with better performance of DSG. For our proposed protocol, MAEF and its alternative A-MAEF consume almost the same amount of energy, which is the lowest amount of energy consumption than all other protocols in all scenarios.

Figure 7 shows the execution time of MAs to visit all CHs including the time to return back to sink. It compares

Table 1 Simulation parameter

Parameter	Value
Number of nodes	[200, 800]
Network transfer rate	250 kbps
Monitoring field size	500 × 500
Node energy	18720 J
Energy consumed by MA execution	5 nJ
MA processing delay	50 ms
MA instantiation delay	10 ms
Collected data size at each node	200 bytes
MA code size	1024 bytes

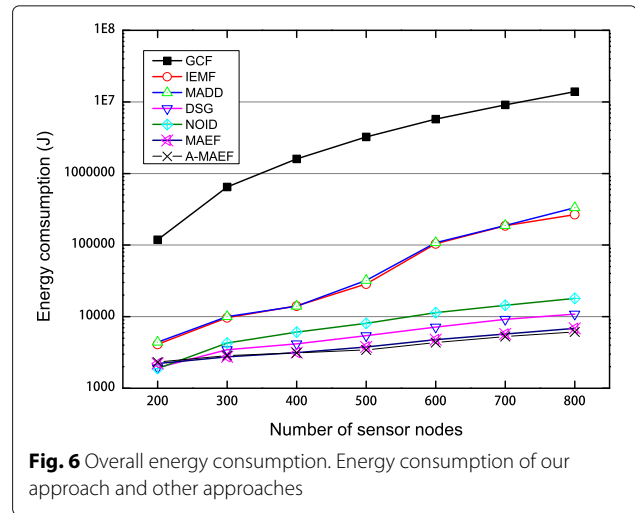


Fig. 6 Overall energy consumption. Energy consumption of our approach and other approaches

our proposed approach MAEF and A-MAEF to the other existing protocols (GCF, MADD, IEMF, DSG, NOID). As it can be observed, the execution time of all protocols increases as the number of SNs increases. GCF has the highest execution time than all other protocols, because of its poor strategy to migrate among SNs nearest to the sink. IEMF and MADD have almost the same execution time with less execution time of IEMF. For DSG, it has less time than all other single MA protocols (GCF, MADD, IEMF), because of multiple MA usage. The DSG perform better than all other protocols with a network of 200 to 400 nodes, but our proposed protocols MAEF/A-MAEF start to perform better than DSG with the increasing size of the network. Also the alternative A-MAEF has almost the same performance, since it spends less time for visiting CHs only and by using optimal itineraries.

Figure 8 compares the total itinerary length of our proposed protocols MAEF/A-MAEF with the other existing

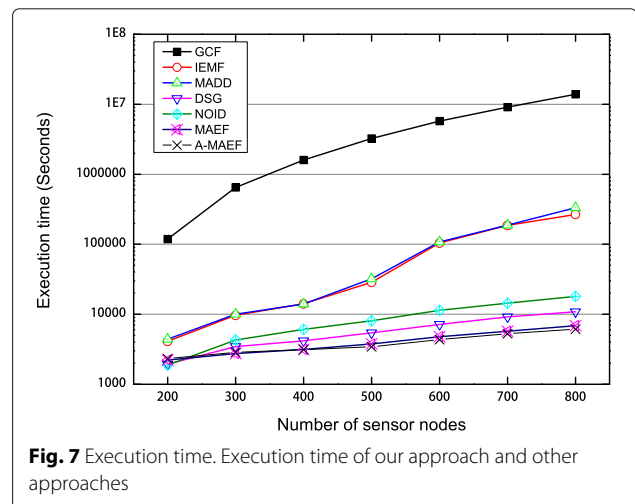
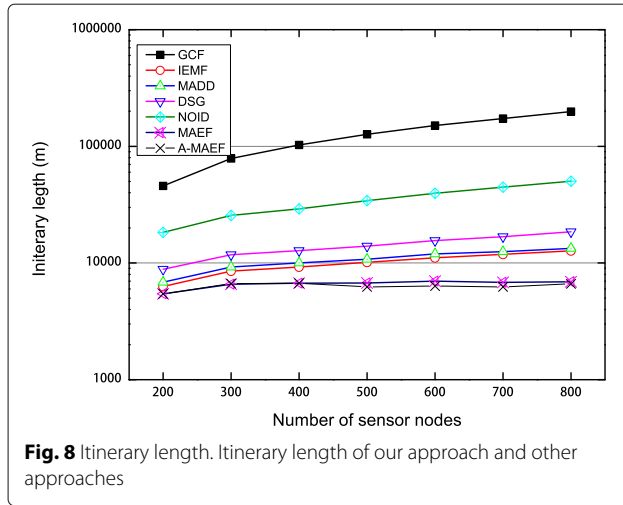
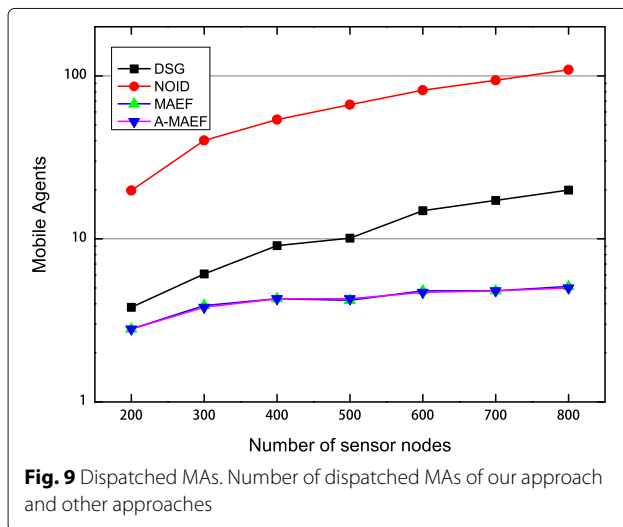


Fig. 7 Execution time. Execution time of our approach and other approaches



protocols IEMF, MADD, GCF, DSG, and NOID. Our proposed protocols have the shortest itinerary length than all the other protocols. This is due to the strategy of planning itineraries just among CHs and not all the SNs and by using MST to plan itineraries for MAs. GCF has the longest itinerary length because of its poor strategy of choosing nearest SN as the next destination and planning itinerary among all SNs. For DSG and NOID, the length of the itinerary is longer than MADD and IEMF. On the other side, MADD and IEMF have almost the same itinerary length.

Finally, Fig. 9 shows the number of dispatched MAs for multiple itinerary planning protocols; as it can be observed, NOID dispatch the highest number of MAs (more than 100 MAs) followed by DSG protocol (20 MAs). On the other side, our protocols (MAEF and A-MAEF) manage to dispatch the minimum number of MAs with better performance.



As it can be observed from the result, our approach perform better than all the other proposed protocols. It consumes less energy and it needs less execution time, also it has the shortest itinerary length. Also, our fault tolerance strategy to use an alternative path in case of failure of the first one makes sure that an alternative path is planned and ready to use in case of the failure of the first one.

7 Conclusions

In this paper, we have presented multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks MAEF. Instead of visiting all SNs, we select just some CHs based on node density, then we dispatch the MAs to collect data from the selected CHs. By using our novel proposed approach, the network lifetime and energy consumption improved significantly. Additionally, the execution time takes less than previous proposed approaches. The itinerary length of our proposed approach is the shortest among all previous proposed ones; this is due to the strategy of visiting only CHs and not all SNs. In case of the failure of the first itinerary due to node failure which is often the case in WSNs, alternative paths are calculated and ready to use by the MAs. Our findings have showed that our proposed approach perform better than the other previously proposed approaches.

Nomenclature

R_{\max}	Maximum transmission range of SN
n	Total number of SNs
V_n	The set of all SNs
T	The set of CHs in normal itinerary
V_{CH}	The set of CHs
V_{left}	The set of remaining SNs
V_{group}	The set of SNs in the same cluster
V_{CH}	The set of CHs
T_{CH}	The normal itinerary of MA among CHs
V_S	The set of alternative CHs
T_S	The alternative itinerary of MA among CHs
M_{CH}	The set of planned itineraries
E_r	The remaining energy of SN
I_i	The impact factor of SN i
I_{ij}	The impact factor between SNs i and j
$I_{\text{accumulated}}$	The accumulated impact factor of SN
$(d_{CH, i})$	The distance between CH and SN i

Abbreviations

CH: Cluster head; CS: Client/server; MA: Mobile agent; MST: Minimum spanning tree; SN: Sensor node

Funding

The author declares that there is no funding for this work.

Availability of data and materials

The datasets supporting the conclusions of this article are available in the github repository <https://github.com/kikosto/castalia-Result>.

Authors' contributions

The individual contributions of each authors to the manuscript are the same. All authors read and approved the final manuscript.

Authors' information

Mohamed EL FISSAOUI is a PhD candidate in wireless networks at the Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco. He received his bachelor's degree in computer science from the University Moulay Ismail and his Master's Degree in computer science from Chouaib Doukkali University, El Jadida, Morocco.

Abderrahim BENI-HSSANE is a researcher and a professor (HDR) at the Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco, since September 1994. He got his bachelor's degree in Applied Mathematics and his Doctorate in computer sciences, respectively, in 1992 and 1997 from Mohamed V University, Rabat, Morocco. His research interests focus on performance evaluation in wireless networks, VANET, security in cloud computing, and big data.

Mostafa SAADI received the B.Sc. degree in Computer Sciences at the University Hassan 2nd, Faculty of Sciences Ain-Chook, Casablanca, Morocco, in 2003, and a M.Sc. degree in Mathematical and Computer engineering at the University Chouaib Doukkali, Faculty of Sciences, El Jadida (FSJ), Morocco, in 2009. He has been working as a professor of Computer Sciences in high school between 2003 and 2013, in Sidi Rahal Beach, Morocco. He received his Ph.D in evaluation, analysis, and simulation of wireless sensor networks at FSJ, Morocco (2009–2013). Actually, he worked as a research assistant professor at Mathematics and Computer Science Department at the National School of Applied Sciences at Khouribga (ENSA-K), Morocco. His research interests focus on performance evaluation in wireless networks.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Computer Science Department, Science Faculty, Chouaib Doukkali University, El Jadida, Morocco. ²Mathematics and Computer Science Department, National School of Applied Sciences, Hassan 1st University, Khourigba, Morocco.

Received: 14 September 2017 Accepted: 12 April 2018

Published online: 02 May 2018

References

- IF Akyildiz, W Su, Y Sankarasubramaniam, E Cayirci, Wireless sensor networks: A survey. *Comput. Netw.* **38**(4), 393–422 (2002)
- S Rani, SH Ahmed, *Multi-hop Routing in Wireless Sensor Networks—An Overview Taxonomy and Research Challenges*. (Springer, Singapore, 2016). <https://link.springer.com/book/10.1007%2F978-981-287-730-7>
- JN Al-Karaki, AE Kamal, Routing techniques in wireless sensor networks: a survey. *IEEE Wirel. Commun.* **11**(6), 6–28 (2004). <https://doi.org/10.1109/MWC.2004.1368893>
- S González-Valenzuela, M Chen, VCM Leung, Chapter 4 - applications of mobile agents in wireless networks and mobile computing. *Advances in Computers*, vol. 82, 113–63 (2011). <https://doi.org/10.1016/B978-0-12-385512-1.00004-9>. <http://www.sciencedirect.com/science/article/pii/B9780123855121000049>
- Y Yuan, M Chen, T Kwon, A novel cluster-based cooperative mimo scheme for multi-hop wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2006**(1), 072493 (2006). <https://doi.org/10.1155/WCN/2006/72493>
- I Vukasinovic, Z Babovic, G Rakocevic, in *2012 IEEE International Conference on Industrial Technology*. A survey on the use of mobile agents in wireless sensor networks, (2012), pp. 271–7. <https://doi.org/10.1109/ICIT.2012.6209950>
- K Moizumi, G Cybenko, The traveling agent problem. *Math. Control Signals Syst.* **14**(3), 213–32 (2001). <https://doi.org/10.1007/PL00009883>
- B Khaleghi, A Khamis, FO Karray, SN Razavi, Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion.* **14**(1), 28–44 (2013)
- D Padiaditakis, Y Tselishchev, A Boulis, in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques. SIMUTools '10*. Performance and scalability evaluation of the castalia wireless sensor network simulator (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, 2010), pp. 53–1536. <https://doi.org/10.4108/ICST.SIMUTOOLS2010.8727>. <http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8727>
- H Qi, F Wang, Optimal Itinerary Analysis for Mobile Agents in Ad Hoc Wireless Sensor Networks (2001)
- A Beni Hssane, ML Hasnaoui, M Saadi, S Benkirane, M Laghdir, Equitable leach-e protocol for heterogeneous wireless sensor networks. *Stud. Comput. Intell.* **315**, 171–6 (2010). <https://doi.org/10.1007/978-3-642-15211-518>. cited By 3
- M EL Fissaoui, S Benkirane, A Beni-Hssane, M Saadi, Scalability aware energy consumption and dissipation models for wireless sensor networks. *Int. J. Electr. Comput. Eng.* **7**(1), 424–31 (2017). <https://doi.org/10.11591/ijece.v7i1.pp424-431>. cited By 1
- S Franklin, A Graesser, in *Intelligent Agents III*, ed. by JP Mueller, MJ Wooldridge, and NR Jennings. Is it an agent, or just a program?: A taxonomy for autonomous agents (Springer, Berlin, 1997), pp. 21–36
- DB Lange, M Oshima, Seven good reasons for mobile agents. *Commun. ACM.* **42**(3), 88–89 (1999). <https://doi.org/10.1145/295685.298136>
- D Chess, C Harrison, A Kershenbaum, in *Mobile Object Systems Towards the Programmable Internet: Second International Workshop, MOS'96 Linz, Austria, July 8-9, 1996 Selected Presentations and Invited Papers*, ed. by J Vitek, C Tschudin. Mobile agents: Are they a good idea? (Springer, Berlin, 1997), pp. 25–45
- A Fuggetta, GP Picco, G Vigna, Understanding code mobility. *IEEE Trans. Softw. Eng.* **24**(5), 342–61 (1998). <https://doi.org/10.1109/32.685258>
- Y Xu, H Qi, Mobile agent migration modeling and design for target tracking in wireless sensor networks. *Ad Hoc Netw.* **6**(1), 1–16 (2008). <https://doi.org/10.1016/j.adhoc.2006.07.004>
- Q Wu, NSV Rao, J Barhen, SS Iyengar, VK Vaishnavi, H Qi, K Chakrabarty, S Member, S Member, On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Trans. Knowl. Data Eng.* **16**, 740–53 (2004)
- G Chen, S Wu, J Zhou, AKH Tung, Automatic itinerary planning for traveling services. *IEEE Trans. Knowl. Data Eng.* **26**(3), 514–27 (2014). <https://doi.org/10.1109/TKDE.2013.46>
- S Chouikhi, IE Korbi, Y Ghamri-Doudane, LA Saidane, A survey on fault tolerance in small and large scale wireless sensor networks. *Comput. Commun.* **69**, 22–37 (2015). <https://doi.org/10.1016/j.comcom.2015.05.007>
- M Chen, T Kwon, Y Yuan, Y Choi, VCM Leung, Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP J. Appl. Signal Process.* **2007**(1), 219 (2007). <https://doi.org/10.1155/2007/36871>
- M Chen, LT Yang, T Kwon, L Zhou, M Jo, Itinerary planning for energy-efficient agent communications in wireless sensor networks. *IEEE Trans. Veh. Technol.* **60**(7), 3290–9 (2011). <https://doi.org/10.1109/TVT.2011.2134116>
- Y Xu, H Qi, Dynamic mobile agent migration in wireless sensor networks. *Int. J. Ad Hoc Ubiquitous Comput.* **2**(1/2), 73–82 (2007). <https://doi.org/10.1504/IAHUC.2007.011605>
- E Shakshuki, H Malik, MK Denko, Software agent-based directed diffusion in wireless sensor network. *Telecommun. Syst.* **38**(3), 161–174 (2008). <https://doi.org/10.1007/s11235-008-9102-4>
- D Lohani, S Varma, Energy efficient data aggregation in mobile agent based wireless sensor network. *Wirel. Pers. Commun.* **89**(4), 1165–76 (2016). <https://doi.org/10.1007/s11277-016-3310-0>
- Y Jing, Z Heng, W Xiao, Z Yuzhi, Y Ping, L Xuemei, S Yuefang, L Zetao, in *2017 29th Chinese Control And Decision Conference (CCDC)*. An energy-efficient routing strategy based on mobile agent for wireless sensor network, (2017), pp. 2632–6. <https://doi.org/10.1109/CCDC.2017.7978959>
- A Mpitziopoulos, C Konstantopoulos, G Pantziou, Deriving efficient mobile agent routes in wireless sensor networks with noid algorithm (2007). <https://doi.org/10.1109/PIMRC.2007.4394337>
- M Chen, S Gonzalez-Valenzuela, VCM Leung, in *Information and Communication Technology Convergence (ICTC), 2010 International Conference On*. Directional source grouping for multi-agent itinerary

- planning in wireless sensor networks, (2010), pp. 207–212. <https://doi.org/10.1109/ICTC.2010.5674660>
29. M Chen, S Gonzalez, Y Zhang, VCM Leung, *Multi-Agent Itinerary Planning for Wireless Sensor Networks*. (N Bartolini, S Nikolettseas, P Sinha, V Cardellini, A Mahanti, eds.) (Springer, Berlin, 2009), pp. 584–97. <https://doi.org/10.1007/978-3-642-10625-5-37>. https://doi.org/10.1007/978-3-642-10625-5_37
 30. M Dong, K Ota, LT Yang, S Chang, H Zhu, Z Zhou, Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks. *Comput. Netw.* **74 Part B**, 58–70 (2014). <https://doi.org/10.1016/j.comnet.2014.06.019>. Special Issue on Mobile Computing for Content/Service-Oriented Networking Architecture
 31. J Wang, Y Zhang, Z Cheng, X Zhu, Empip: energy-efficient itinerary planning for multiple mobile agents in wireless sensor network. *Telecommun. Syst.* **62**(1) (2016). <https://doi.org/10.1007/s11235-015-9985-9>
 32. I Aloui, O Kazar, L Kahloul, A Aissaoui, s. servigne, in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies. BDAW'16*. A new “data size” based algorithm for itinerary planning among mobile agents in wireless sensor networks (ACM, New York, 2016), pp. 36–1369. <https://doi.org/10.1145/3010089.3010121>. <http://doi.acm.org/10.1145/3010089.3010121>
 33. S Pourroostaei Ardakani, J Padget, M De Vos, A mobile agent routing protocol for data aggregation in wireless sensor networks. *Int. J. Wireless Inf. Netw.* **24**(6), 27 (2017). <https://doi.org/10.1007/s10776-016-0327-y>
 34. Y Zhang, Z Zhou, D Zhao, M Barhamgi, T Rahman, Graph-based mechanism for scheduling mobile sensors in time-sensitive wsns applications. *IEEE Access.* **5**, 1559–69 (2017). <https://doi.org/10.1109/ACCESS.2017.2667687>
 35. G Mao, B Fidan, BDO Anderson, Wireless sensor network localization techniques. *Comput. Netw.* **51**(10), 2529–53 (2007). <https://doi.org/10.1016/j.comnet.2006.11.018>
 36. M Khelifi, S Moussaoui, S Silmi, I Benyahia, Localisation algorithms for wireless sensor networks: A review. *Int. J. Sen. Netw.* **19**(2), 114–29 (2015). <https://doi.org/10.1504/IJSNET.2015.071632>
 37. Z Fang, Z Zhao, X Cui, D Geng, L Du, C Pang, Localization in wireless sensor networks with known coordinate database. *EURASIP J. Wirel. Commun. Netw.* **2010**(1), 901283 (2010). <https://doi.org/10.1155/2010/901283>
 38. A Beni Hssane, ML Hasnaoui, S Benkirane, D El Ouadghiri, M Laghdir, Householder algorithm applied to localization for wireless sensor networks. *Int. J. Mob. Comput. Multimedia Commun.* **4**(1), 18–30 (2012). <https://doi.org/10.4018/jmcmc.2012010102>. cited By 0
 39. M Chen, W Cai, S Gonzalez, VCM Leung, in *Ad Hoc Networks*, ed. by J Zheng, D Simplot-Ryl, and VCM Leung. *Balanced Itinerary Planning for Multiple Mobile Agents in Wireless Sensor Networks* (Springer, Berlin, 2010), pp. 416–28

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)