UNIVERSITY OF BATH

Citation for published version:
Pourroostaei Ardakani, S, Padget, J & De Vos, M 2017, 'A Mobile Agent Routing Protocol for Data Aggregation in Wireless Sensor Networks', *International Journal of Wireless Information Networks*, vol. 24, no. 1, pp. 27-41. https://doi.org/10.1007/s10776-016-0327-y

DOI:
10.1007/s10776-016-0327-y

Publication date:
2017

Document Version
Peer reviewed version

Link to publication

The final publication is available at Springer via http://dx.doi.org/10.1007/s10776-016-0327-y

University of Bath

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 04. Sep. 2026

Noname manuscript No.
(will be inserted by the editor)

A Mobile Agent Routing Protocol for Data Aggregation in Wireless Sensor Networks

Saeid Pourroostaei Ardakani · Julian Padget · Marina De Vos

Received: date / Accepted: date

Abstract Mobile agent data aggregation routing forwards mobile agents in wireless sensor network to collect and aggregate data. The key objective of data aggregation routing is to maximise the number of collected data samples at the same time as minimising network resource consumption and data collection delay. This paper proposes a mobile agent routing protocol, called Zone-based Mobile agent Aggregation. This protocol utilises a bottom-up mobile agent migration scheme in which the mobile agents start their journeys from the centre of the event regions to the sink aiming to reduce the MA itinerary cost and delay and increase data aggregation routing accuracy. In addition, the proposed protocol reduces the impact of network architecture, event source distribution model and/or data heterogeneity on the performance of data aggregation routing.

Keywords wireless sensor networks · mobile agents · data aggregation · itinerary planning.

1 Introduction

A *Wireless Sensor Network* (WSN) is an infrastructure comprised of a number of computing devices that have the ability to sense and report ambient data. A WSN is deployed in an ad-hoc manner without requiring any specific infrastructure or centralised control. WSN architecture is generally classified as either distributed or hierarchical. In the former, there is no specific distribution topology and sensor nodes are randomly scattered in the sensing

S. P. Ardakani
Assistant Professor in Computer Science, Allameh Tabataba'i University, Iran
E-mail: saeed.pouroostaei@atu.ac.ir

J. Padget and M. D. Vos
Senior Lecturer in Computer Science, University of Bath, UK
E-mail: {jap and mdv}@bath.ac.uk

area, whereas in the latter, sensor nodes are organised in a number of separate groups such as clusters [1]. The sensor nodes usually are a set of static, inexpensive, small and/or tiny electronic devices that communicate through limited wireless channels. They are highly constrained resources in terms of energy, computation, communication and storage.

Data aggregation is a technique that collects data samples from different sources and then combines them using an aggregation function (e.g. Average and Maximum) to express the result in a summary form for further analysis [2]. It has the potential to reduce the number/size of transmissions and consequently decrease the network resource consumption [2]. Data aggregation needs routing to interconnect source nodes of which data samples are collected and combined. There are two schemes of data aggregation routing in WSNs: client-server and mobile agent [3], [4]. Client-server scheme allows the source nodes to transmit their data to either the sink or intermediate aggregators for data aggregation. Mobile agent data aggregation routing forwards either a single or multiple mobile agent(s) throughout a network to collect and aggregate data samples from the source nodes.

Zone-based Mobile agent Aggregation (ZMA) is a MA itinerary planning protocol that dynamically establishes optimal paths to move the MAs across the network for data aggregation. This protocol decomposes the event regions into a set of zones that are formed in a Data-Centric (DC) manner. In each zone, a set of nodes (called Zone Mobile Agent Coordinators) are selected to start the MA journeys. ZMA limits the MA migration to data regions which are formed according to the consumer interests. This results in increasing the number of captured data samples and enhancing the data aggregation accuracy. In addition, ZMA avoids random/blind MA migration and therefore reduces the journey delay and energy consumption.

In the remainder of this article, Section 2 outlines a set of MA data aggregation itinerary planning protocols to highlight their advances, features and techniques. Section 3 describes the ZMA data aggregation routing protocol. It focuses on the key techniques which are used in ZMA to resolve the existing drawbacks and enhance the performance of MA data aggregation routing. Section 4 outlines the experimental plans to test and evaluate the performance of ZMA. Section 5 tests the performance of ZMA according to five key metrics including total consumed energy, total number of captured data samples (accuracy), average end-to-end delay, MA hop count and total transmitted traffic which are usually used to test the performance of data aggregation routing protocols. The results of each parameter are measured and discussed to evaluate the performance of ZMA in comparison to two selected MA data aggregation routing protocols. Section 6 discusses the key points of the results to conclude advantages and disadvantages of the proposed protocol and then highlights the research issues which need to be addressed as future works.

2 Related Works

This section introduces a set of well-known MA itinerary planning protocols [5] have been proposed for data aggregation in WSNs. Global Centre First (GCF) and Local Closest First (LCF) are two basic MA routing protocols proposed by [6] that move a single MA into the environmental event region(s) for data aggregation. GCF routes a single MA to visit the source node which is the closest to the centre of event region through shortest paths, whereas an MA is moved in LCF to the closest source node from the current location. These protocols are comparatively simple to implement and have low computational complexity to route the MA. However, data aggregation cost and delay increases when network size and/or density rises because the single MA needs to travel through long paths to visit the source nodes. Moreover, the performance of LCF and GCF highly depends on the current location of MA and event sources. For example, the MA would be able to visit the source nodes in GCF if the centre of event region is known by the sink. Although this is not critical when the event sources are centralised, reporting the centre of random distributed event regions to the sink/MA is expensive for WSNs, especially when the deployed network is large and dense.

Itinerary Energy Minimum for First-source-selection (IEMF) and Itinerary Energy Minimum Algorithm (IEMA) proposed by [7] establish minimum cost paths for a single MA to collect and aggregate data. Similar to LCF, the objective of IEMF is to reduce MA migration cost by selecting the minimum cost (energy) link among all available ones. IEMF allocates an estimated cost value to each route that is established to an event region. According to the cost value, it selects the closest node that resides on the minimum cost link to migrate. The difference of LCF and IEMF is that LCF selects the closest node to the current location of MA, whereas IEMF considers the estimated cost value on each link to select the closest node to migrate. IEMA extends IEMF by selecting the next visiting source nodes in an iterative manner. Each available route to the source regions is allocated by a cost value that is updated iteratively when the cost value of a node is measured. Indeed, IEMA considers a number of available links to the event regions in an iterative manner to find out the route in which MA migration cost is minimised. As a result, it can be perceived that LCF and IEMF are IEMA with zero and one iteration, respectively.

Near-Optimal Itinerary Design algorithm (NOID) [8] utilises multiple MAs which independently travel throughout the network to collect and aggregate data samples. It results in increasing the parallelism degree of data aggregation routing and consequently reducing delay as a number of MAs move throughout the network in parallel to aggregate data samples. The MA migrations are started from the sink through the routes that are established for the event regions. NOID allocates a cost value which considers hop count and residual level of energy to each link. This allows the MAs to select the closest node residing on the minimum energy consuming link to move. NOID also considers the amount of collected data at each node to control the MA size.

As MAs become heavier when a number of sensor nodes are continuously visited, forwarding MAs without considering their size increases the transmitted network traffic that results in increasing network resource consumption. For this reason, NOID monitors the MAs size at each node to avoid forwarding heavy MAs. It stops the migration and return the MA to the sink if its data parts become full and/or heavy. However, MA migrations to overlapped areas and capturing redundant data samples are the drawbacks of NOID. Besides, the complexity and/or overhead of managing multiple MAs in NOID depends on the network size.

Tree-Based Itinerary Design (TBID) [9] proposes a data aggregation protocol in which the MAs move through a number of spanning trees (SPTs) to collect and aggregate data samples over a zone-based network. Each tree is rooted in the single-hop neighbourhood of the sink and assigned by a MA for data aggregation. First, TBID forms a set of concentric zones around the sink. Radius of each zone is $\frac{N \times R}{2}$ which N is the zone number and R is the maximum radio range of node. Then, each node residing in the first zone starts to establish a spanning tree with the source nodes. To form the tree, source nodes which reside in the outer zones are incrementally interconnected to the inner ones using a greedy-like algorithm. The inter-zone links form the tree trunk, whereas the intra-zone links shape the tree branches. This procedure is repeated until source nodes in the last zone are reached. At the end, the MAs start their journeys from the roots to visit all source nodes that reside on the tree branches. Each MA sweeps all connected nodes to the tree in each zone and then move to visit next source nodes. The MAs return through the same infrastructure to the sink to deliver aggregated results. The drawback of TBID is that SPTs are established in a proactive manner. It increases the network resource consumption in the case of frequent network topology changes. Moreover, complexity and cost of data aggregation increases when the algorithm is implemented in large WSN as a greater number of SPTs need to be established.

3 The ZMA Protocol

The Zone-based Mobile Agent (ZMA) approach is a routing protocol which moves multiple MAs throughout the network for data aggregation. This protocol routes the MAs over a zoned network to collect and aggregate sensory data. The network model, forming the zones and ZMA path planning algorithm are discussed in next.

3.1 Network Model

The network model consists of three key components: (1) The sink node(s) is/are data consumer access-point to monitor the network performance. They have sufficient resources for data storage, communication and/or computation.

(2) The sensor nodes are responsible for measuring ambient quantities and/or forwarding the MAs. They may be homogenous or heterogeneous in terms of having variant levels/units of resources and data. It is assumed that the nodes are synchronised to manage the message passing and wireless communications [10]. (3) The event sources generate the environmental data in the network field. They may be either static or mobile. The sources are scattered in the network according to either Event-Radius (ER) and Random-Source (RS) models. The event occurs in a single point of the sensing field in the former, whereas the event sources are randomly distributed in the latter.

3.2 Forming the Zones

ZMA partitions the network into a set of concentric zones around the sink. It is started from the sink until all nodes are allocated by a zone number. The zones are constructed for three reasons: (1) limiting the routing communication to the bounded regions to reduce overhearing and network resource consumption, (2) localising the MA migrations to the zones, (3) guiding the MAs to return to the sink by moving from the outer to the inner zones. It avoids blind/random walk and/or heuristic migrations for MAs.

The zone forming phase starts when the sink broadcasts a hello message (version 1) $Hello_{v1}$. Similar to TBID [9], the messages are broadcasted within a $\frac{R}{2}$ radio range to form the zones. The messages form a set of concentric $\frac{R}{2}$ width zones around the sink. This zone size guarantees the interconnections between the nodes (with the maximum radio range R) at zone (i) with at least one node in the outer (i+1) and the inner (i-1) zones. The header of $Hello_{v1}$ message maintains a Z_{Nb} value to show the zone number. It is initiated to zero by the sink. Each node receiving the $Hello_{v1}$, increases the Z_{Nb} by one and then updates the message with the new Z_{Nb} value for the next hop. A node updates its zone number according to the minimum received Z_{Nb} value. The minimum Z_{Nb} value shows a minimum hop count path to the sink.

Each node records one of its single-hop neighbours as TS node (To the Sink) during the zone forming phase. These nodes are responsible for providing backward paths to the sink. A sender of $Hello_{v1}$ with minimum Z_{Nb} plays TS role for the receiver node at the next (outer) zone. For this reason, each node at zone i+1 keeps the ID of the last sender in zone i as TS when its zone number is updated. The nodes may also record a set of BackUp TS ($BUTS$) nodes if they receive multiple zone numbers. These nodes are used when a TS node fails or is not available. The $Hello_{v1}$ are re-broadcasted within $\frac{R}{2}$ radio range until all nodes get a zone number.

Message conflict may arise if hello messages are frequently and/or simultaneously used by a large number of nodes. To resolve this, ZMA allows the nodes to transmit the received hello messages over a uniform period time of (A, B). In other words, the received messages are re-broadcasted after a short random time and not immediately when they are received. This technique (similar to [11]) decreases the number of sensor nodes that simultaneously access to wire-

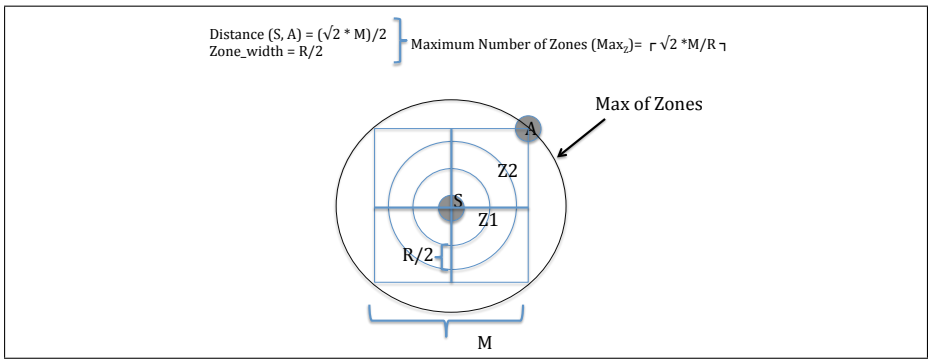


Fig. 1: The maximum number of zones in a $M \times M$ network

less channels to broadcast the messages. That is, the sensor nodes wait for a T_i which is calculated at each node using Equation 1 and then re-broadcast the hello messages for next hop. V_i is a random value which is selected from a uniform distribution of time values in the range $(A,\ B)$. The random time range is set at each sensor node in advance of network deployment.

$$T_i = R_i + V_i \tag{1}$$

Hello message failure during the zone forming phase influences the performance of ZMA as the nodes without the zone numbers cannot properly forward the MAs. To resolve this, the nodes which miss or lose the hello messages ask their neighbours to get a zone number. The nodes broadcast a zone enquiry message after a time period called Zone Time (ZT). This depends on the maximum number of the created zones in the network. According to Figure 1, the maximum number of zones which are created in a $M \times M$ m^2 network is Max_Z (which equals to $\lceil \frac{\sqrt{2} \times M}{R} \rceil$). Hence, the maximum required time to finish the zone forming procedure can be calculated according to Equation 2. $start_T$ is the network start time, max (V_i) is the maximum value of the range $(A,\ B)$ which was explained earlier in Section 3.2, and Com_T is communication delay time that can be measured locally at each node [12]. After ZT, any node which has not already received a zone number broadcasts a zone enquiry message and then waits for Allowed Hello Loss [13] to receive the reply. Allowed Hello Loss determines the maximum time that a node needs to wait before assuming a message failure. According to [14], two seconds for the Allowed Hello Loss is recommended. The smallest received zone number is selected, incremented by one and set as zone number. The sender node of the respected message is recorded as a TS node.

$$ZT = start_T + Max_Z \times (Com_T + max(V_i)) \tag{2}$$

Algorithm 1: Vicinity Discovery Algorithm

```
Data: Routing Table (RT), Node ID (NID)
Node C:
if Hello_v2 message is received then
    if Z_Nb = Zone(C) then
        Distance = RSSI (link);
        RT ← (NID, Distance, D);
    else
        Discard (Hello_v2);
    end
end
```

3.3 Identifying the ZMAC Nodes

The Zone Mobile Agent Coordinators (ZMACs) are responsible for initiating the MA migrations at each zone during data aggregation routing. They are elected using a weighting function similar to Common Election Algorithm (CEA) [15]. The procedure of ZMAC selection is explained in next.

3.3.1 Vicinity Discovery

Vicinity discovery phase is performed after zone forming in ZMA. Each node discovers its local vicinity by finding available connections to any neighbour that has same type data in the same zone. They use hello messages (version 2) $Hello_{v2}$ for vicinity discovery. This message has a similar structure to $Hello_{v1}$, however its header is slightly different. The message header has an additional field named data-type that is used to establish data centric intra-zone ties. When a $Hello_{v2}$ message is received, a data centric path is recorded from the receiver to the sender if both the nodes have the same zone number. This means that a $Hello_{v2}$ message is discarded if it is received from any node with different zone number.

Each node measures Received Signal Strength Indication (RSSI) [16] value at the arrival of a $Hello_{v2}$ to estimate its distance to the sender node. RSSI is measured using the power of sent (P_T) and received (P_R) signals according to Equation 3. A receiver node would be able to measure/estimate its shortest Euclidean path to a sender node if the RSSI value is maximised. This means that RSSI value is increased when the sender node is closer as the receiving signals have greater power. According to Algorithm 1, routing tables are updated with the received $Hello_{v2}$ messages. The routing tables allow nodes to find their neighbours ID, available data type and distance (RSSI). To get more reliable results, it is assumed that a Line-Of-Sight (LOS) model [17] is used by ZMA for wireless signal propagation and there is no ambient noise affecting the wireless signals.

$$RSSI(dB) = 10\log(\frac{P_R}{P_T}) \tag{3}$$

3.3.2 Weighting Function

ZMA utilises a weighting function to find the nodes that are more eligible to become ZMAC at each zone. The weighting function (Equation 4) returns a weight value for each node according to its connectivity degree ($count_{(i,j)}$), residual energy level (E_C) and proximity to the event sources ($P_{(i,j)}$). The nodes with higher weight value ($W_{(i,j)}$) have a greater chance to become a ZMAC. In other words, a node is selected as ZMAC if it has the highest level of residual energy, the greatest data centric connectivity degree and the shortest distance to the event source in its single-hop vicinity.

$$W_{(i,j)} = (count_{(i,j)} \times P_{i,j}) \times \frac{E_C}{E_T} \tag{4}$$

To calculate the weight, $count_{(i,j)}$ and $P_{(i,j)}$ are computed at each node. First, the collected information from vicinity discovery is classified at each node based on the measured data types to rank the connectivity degrees in a Data-Centric (DC) manner. Second, $count_{(i,j)}$ and $P_{(i,j)}$ are calculated based on the classified DC links in two steps: (1) $count_{(i,j)}$: is the total value of available links for data type j at node N_i. (2) $P_{(i,j)}$: is the average distance that shows the proximity of a receiver and sender node with respect to the type of data measured. It is used to establish short, low-energy links to the source nodes. That is, the $count_{(i,j)}$ values are ordered at each node according to the type of data. Then, each node calculates its average distance to the neighbour source nodes based on the data type. Similar to [18], ZMA utilises Equation 5 to calculate the average distance to the neighbour source nodes. In the equation, $P_{(i,j)}$ represents the average RSSI value of the links which are established based on data type j at node N_i.

$$P_{(i,j)} = 10^{\left(\frac{\sum_{k=1}^{count_{(i,j)}} RSSI(i,k)}{count_{(i,j)}} \times \frac{-1}{10}\right)} \tag{5}$$

3.3.3 ZMAC Selection

The ZMACs are selected using a distributed leader selection algorithm. The nodes collect the required information for the leader selection procedure from their local vicinity and then locally select the ZMACs. According to Algorithm 2, the nodes broadcast a new hello message (version 3). Its header has two additional fields: D_i and $W_{(i,j)}$. The first is the sender data type and the second is the set of the weight values of data types that are measured from the single-hop vicinity. By this, each receiver node finds the greatest weight value for each reported data type. The node with the greatest value is selected as a ZMAC for the respective data type. Otherwise, the receiver node considers itself as the ZMAC for the zone if none of the neighbour reports a greater weight value. In the case of having the same weight values, the node with smaller ID is selected as the ZMAC. The ZMACs wait until receive the sink queries to migrate the MAs for data aggregation.

Algorithm 2: ZMAC Selection Algorithm

Data: Routing Table (RT), Node ID (NID)
RT Update at Node C:
if $Hello_{v3}$ *message is received* **then**
  if Z_{Nb} = *Zone(C)* **then**
    $RT \leftarrow (NID, W(i,j), D)$;
  else
    Discard ($Hello_{v3}$);
  end
end
ZMAC Selection at Node C:
$Weight \leftarrow Max(RT.W)$;
if *Weight* >= *W(C)* **then**
  SELECT Min(NID) from RT WHERE W = Weight As ZMAC;
else
  $ZMAC \leftarrow C$;
end

The ZMACs are selected by ranking the weight values which are collected during the vicinity phase. This ranked list of nodes shows the most eligible node which may become the new ZMAC if the current one fails. The next node in the list has the greatest weight in the same zone (same data type) after the current ZMAC. In this node failure case, the current ZMAC checks the availability of a candidate node and then sends a role exchange message if it is available and has enough energy to start the MA migrations. The new node becomes the data region ZMAC as soon as receiving the role exchange message from the failing ZMAC without requiring any additional cost. However, updating the ranked list according to the network topology change is a drawback which will be discussed in the next.

3.4 Mobile Agent Routing

ZMA utilises a bottom-up MA migration scheme in which the MAs move from ZMACs to the sink. ZMA assumes that the sensor nodes are initiated by the aggregation functions in advance of network deployment. Hence, each ZMAC which receives the data collection request generates a MA with the requested aggregation function for data aggregation. The data collection requests are propagated by the sink throughout the network via direct communications. The sink adjusts its radio communication range to send a message to a particular part of the network. Using this approach, only the ZMACs stay on duty to receive data requests and other ones go to sleep to conserve energy. The ZMACs which match the sink queries update their MA codes according to the sink interests and then start to move the MAs.

The structure of MAs in ZMA consists of four components: identification, data space, code part and itinerary. The identification provides the identity information of the MA and dispatcher, the data space stores the aggregated data, the code is the aggregation function and the itinerary provides the MA

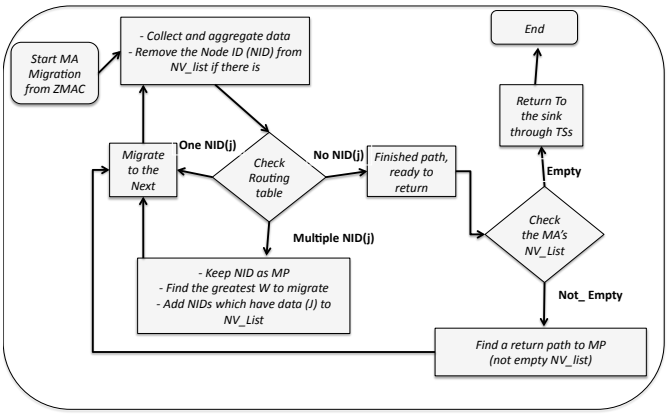


Fig. 2: MA migration chart in ZMA

routing information. The itinerary consists of four fields: next node ID, MP IDs, visited nodes list, non-visited nodes list (NV_List). The next node ID shows the address of the source node which the MA goes to next. MP IDs are the list of Meeting Point (MP) nodes referring to the nodes that may be visited in next migration rounds. The MP nodes have multiple links to the interesting source nodes. They may be visited again during next MA migrations if any of their neighbours is missed out. The visited node list consists of the node addresses which have been visited already, whereas the non-visited is the list of source nodes that are not visited yet and should be captured in next. Each non-visited list refers to an MP which has links to the non-visited nodes. Hence, each MA may return to the MP nodes (of the NV_List) which still have ties to the non-visited nodes.

The MAs collect the first data sample from their respective ZMAC and then find the next node that has the greatest weight value ($W_{(i,j)}$) to visit next. According to Figure 2, the MA considers the routing table at each node to find the next hop. In the simplest case, the MA may find just one node in the routing table that matches the sink query. The next node ID is set to the node ID (NID) and then the MA moves to collect and aggregate the data. If multiple nodes are found, the MA marks the host node as MP to return later for further migrations. Then, it selects the node which has the greatest weight to migrate to. The remaining possible nodes are stored as a list (NV_List) of the MP in the MA itinerary part. According to Figure 2, the MA migrate to the nodes and removes the IDs from the list one by one when they are being visited during the journey. The procedure is repeated until the MA visits a node that has no more links to the source nodes. In this case, the MA checks its NV list to find if there is any non-visited node. If there is, the MA returns to the MP of the list using its recorded journey to change the migration direction to visit the non-visited nodes. Otherwise, the MA prepares to return the aggregated result to the sink via TS nodes.

In the case of node failures, ZMA performs a mechanism depending on the failing node role to update the routing tables. It is assumed that the network topology is changed in ZMA if the available residual energy level falls below the required threshold to maintain the minimum connectivity between the nodes and/or keep the node alive. Unexpected node failures such as hardware damage and/or node capture attacks are addressed as future work. If the failing node is a source node, it sets its weight value to zero and then broadcasts a message to inform its neighbours. The message lets the neighbours know that the node in their vicinity failed and there is no more link/data to follow. If the node is an MP, it needs to find another node in its vicinity that has the ability to minimise the disconnections caused by the failures. This means that the new MP should have the ability to cover the maximum possible number of source nodes in the vicinity of the failure. In this case, the failing MP broadcasts a message called $Fail_{(MP)}$ to inform its neighbours of the failure. The message is attached also by a list of the source nodes that need to be covered in next MA migrations. Each node which receives the message updates its routing table with the information of failing MP and replies back then if it has available links to any of the nodes. The reply message is attached by the list of requested ties (if they are available) and the sender node weight value. The failing MP selects the node with greatest weight as it has the ability to cover broadest area (greatest number of non visited nodes) amongst all nodes that received the failures messages. The new MP begins to play the role as soon as it receives the confirmation from the failing MP node. If a ZMAC node is failing, it broadcasts a $Fail_{(ZMAC)}$ message. Each node which receives the message updates the weight value of the failing ZMAC to zero in its routing table. Then, the nodes perform Algorithm 2 to select the new ZMAC.

4 Experimental Plan

WSN deployment with numbers of real sensor nodes is expensive for empirical research as it may need great resource and time. For this reason, simulation is often used to test and evaluate WSN research algorithms. We have used a network simulator named OMNET++ [19] to implement and test our experiments. OMNET++ is an open-source, component-based and discrete event simulation that is used to simulate ZMA routing protocol. This simulator has a modelling framework called MiXiM [20] for mobile and/or fixed wireless networks such as wireless sensor networks. It offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols (i.e. B-Mac) [21].

The experiments measure five metrics which are usually used in the literature to evaluate the performance of MA data aggregation routing protocols [22], [23], [24]: total consumed energy, total number of captured data samples (accuracy), average end-to-end delay, MA hop count and total transmitted traffic.

1. **Total consumed energy:** represents the total amount of energy that is consumed for establishing the MA migration infrastructure, routing the MAs and network deployment and maintenance [22].
2. **Total number of captured data samples (accuracy [24]):** represents the number of data samples that are properly collected and reported to the sink. This parameter is rooted in the routing algorithm's ability to find data regions and establish reliable links to forward MAs for data aggregation and return the aggregated result to the sink.
3. **Average data collection end-to-end delay:** represents the average End-To-End delay (ETE) of MAs during the data aggregation procedure. It is measured as average time since the MAs start to collect data until they return to the sink and deliver the results [7], [25]. Average ETE influences data accuracy and freshness [10].
4. **Total MA hop count:** this is collected in order to measure the routing protocol ability to establish minimum hop count paths for MA to migrate [23], [22]. The objective is to reduce the hop count by avoiding random and/or blind MA migrations. This results in reductions to ETE and network resource consumption.
5. **Total transmitted traffic:** represents the total amount of transmitted (sent and received) network traffic during data aggregation routing procedure [26]. Network energy consumption is increased if the network traffic is increased. Moreover, increasing network traffic results in increased buffering, wireless channel access and transmission delays.

4.1 Simulation Setup

Three parameters are considered to design the simulation experiments: area size, node count and data density. These let us to observe the routing protocols behaviour, scalability and performance according to varying area size, node count, and data density. The experimental parameters are explained as below:

1. Area size: area size influences the wireless communication type (single or multi-hop) and consequently the performance of routing. The sensor nodes usually communicate in single-hop in small networks, whereas they would need to communicate in multi-hop when the network size is increased.
2. Node count (node density): it focuses on varying the number of network nodes to test the protocol scalability.
3. Data density: it is defined as the number of desirable source nodes in the network. This parameter allows us to observe the ability of MA routing protocol to find and capture interesting data samples when the proportion of desirable source nodes is varied in the network.

First, the network is deployed with three different area sizes in a two-dimensional field: small (200×200 m^2), medium (400×400 m^2) and large (800×800 m^2). This allows observation of the protocol's behaviour and performance in big, medium and small networks.

Table 1: The setup simulation parameters

Parameters	**Simulation Time**	**Repetition**	**Network initialisation time**
Range	3600s	300	50s
Parameters	**Environmental noise**	**Node distribution model**	**Sink location**
Range	enabled	random	a single sink in the centre
Parameters	**Node battery capacity**	**MAC protocool**	**Node radio range**
Range	99999 mAh (3.3 V)	B-MAC	75 meters
Parameters	**Node count**	**Area size**	**data density**
Range	small (16, 32, 64) medium (64, 128, 256) large (256, 512, 1024)	small (200×200 m^2) medium (400×400 m^2) large (800×800 m^2)	25%, 50%, 75% and 100%

To test protocol scalability, a varying node count is considered for each area size. Deploying networks with a variable node count lets us observe the protocol's behaviour, scalability and performance in sparse and dense networks. A minimum required number of nodes ($Count_N$) to deploy a wireless network is calculated based on equation[1] 6 [27]. N is the number of nodes, R is the maximum radio range, O is the overlapping area between nodes radio range, and M and K are the dimensions of the network field. Accordingly, each network is set up with a minimum number of nodes that is required to provide a connected network in the area. Then, the node count is increased with respect to the density which is calculated using the Equation 7 [28]. This means that first the protocols are tested over a small network (200×200 m^2) that is deployed with node count of 16, 32 and 64. The same experiments are then performed in medium (400×400 m^2) and large (800×800 m^2) areas with – in order to produce the similar levels of node density – 64, 128, 256 and 256, 512, 1024 nodes, respectively.

$$Count_N = \left\lceil \frac{0.5 \times (M \times K)}{(R - (0.5 \times O))^2} \right\rceil \tag{6}$$

Last, each experiment features one of four proportions of source nodes which have interesting data samples to report. Each node count in each area size is allocated with four different data densities (25%, 50%, 75% and 100%) that need to be collected/reported. This would result in evaluating the performance of routing protocols to find, collect and aggregate the random scattered data samples in the network. The setup simulation parameters are shown in Table 1

$$Density = \frac{N}{M \times K} \tag{7}$$

[1] To find out the number of uniformly positioned (grid) nodes to fully cover a 2D area, factor 0.3125 should change to 0.5 in this equation. The original equation (with factor 0.3125) does not consider the uncovered area which is formed among each four sensor nodes that are placed in a 2×2 grid. Owing to this, factor 0.5 should be used as one node is required to fill the uncovered area for each four nodes.

5 Results and Discussions

This section evaluates the performance of ZMA, NOID [8] and TBID [9] according to the routing performance parameters that were chosen.

5.1 Total Energy Consumption

ZMA reduces the energy consumption as compared to NOID and TBID when network node count increases. In addition, it outperforms the benchmark protocols when data density increases, especially in dense networks. This stems from two key reasons:

1. **Limiting communications:** ZMA limits the routing communications into the data regions. This means that the nodes communicate to each other if they belong to the same zone and/or have data samples which match the MA requirements. Otherwise, the nodes leave the communication and go to sleep to save energy.
2. **DC MA migration:** the MAs are routed through data centric paths to the nodes that have the greatest connectivity to the interesting source nodes. In addition, ZMA localises the MA migrations into the data regions to avoid blind and/or random migrations. This means that MAs move at each data region only if an interesting data sample is waiting to be collected. This results in reduction of MA migration hop count and consequently energy consumption.

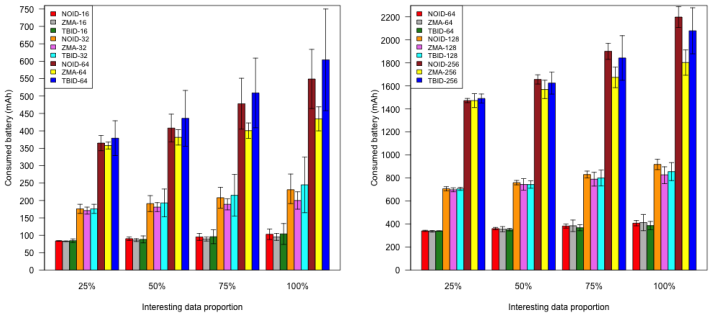
According to Figure 3, ZMA has a better performance in terms of energy conservation when it is used in a small area such as 200×200 m^2. This is because the cost of zone forming is reduced when the deployed network is small. However, energy efficiency of ZMA is reduced as compared to NOID and TBID when the area size increases. This is because of ZMA's capability to find and capture a greater number of desirable data samples in the network.

5.2 Total Number of Captured Data Samples (Accuracy)

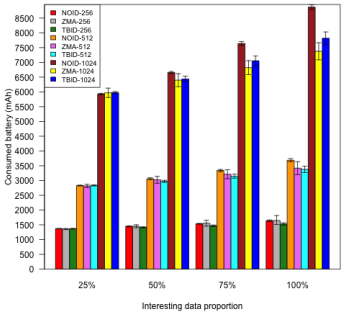
ZMA outperforms both the benchmark protocols in terms of accuracy. This means that the MAs in ZMA have the ability to find source nodes and deliver captured data samples to the sink in either sparse or dense networks. According to Figure 4, TBID and NOID are not efficient to find and capture desirable data samples when the network is sparse or data density is low. The accuracy of the benchmark protocols is highly dependent on the node count and/or data density in the network. The reason is that the MAs are not informed by the intermediate nodes about the location of source nodes at which to gather data samples.

The accuracy of ZMA is better than TBID and NOID as the area increases. This is for three reasons:

Fig. 3: Energy consumption of MA routing protocols.



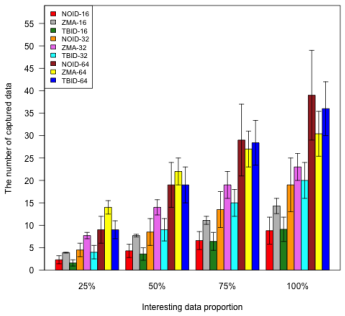
(a) small area (200×200)m^2 (b) medium area (400×400)m^2



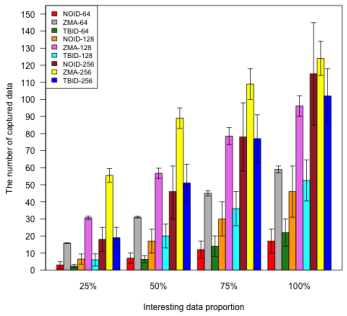
(c) large area (800×800)m^2

1. **Forming data regions:** ZMA has the ability to discover and form the event regions for the MAs to migrate. The MA migration areas are formed in a DC manner by the sensor nodes that have interesting data to report. It would result in interconnecting the source nodes for the MAs to move and visit. Hence, the MAs would be able to visit a number of source nodes which are connected through single or multi-hop DC links if one of them is visited.
2. **Bottom-up MA migration:** ZMA utilises a bottom-up scheme for MA migration. This means that the MA migrations are started from the centre of the event regions (ZMACs) that are surrounded by the desirable source nodes. Each MA migration is an opportunity to capture one new data sample as it starts from a ZMAC node that is close to the centre of an event region and has short links to desirable source nodes.
3. **Maintaining the list of non-visited source nodes:** ZMA records the address of visited and non-visited nodes to avoid looping and to visit the missed source nodes. Using the list, the MAs may return to MP nodes

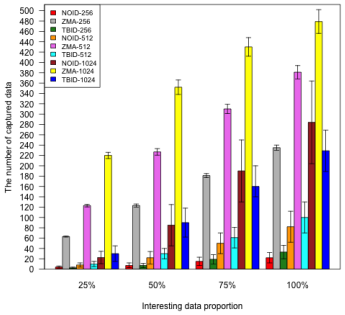
Fig. 4: Accuracy of MA routing protocols.



(a) small area (200×200)m^2



(b) medium area (400×400)m^2



(c) large area (800×800)m^2

which have links to the non-visited source nodes if any source node is missed to visit.

According to Figure 4a, it is observed that the accuracy of TBID and NOID is better than ZMA in a dense network with a high number of source nodes in a small area. In this case, a smaller number of the event regions is formed in ZMA, resulting in a small number of MAs for data collection. On the other hand, a greater number of MAs is generated in NOID and TBID because of an increased number of source nodes which are able to communicate directly with the sink or its single-hop neighbours in a small area. Increasing the number of MAs and/or the node count (which leads to increased interconnectivity between the source nodes) increases the probability of finding and capturing desirable data and consequently improves the accuracy in TBID and NOID over ZMA in a dense network in a small area.

5.3 Average Data Collection End-to-End Delay

ZMA reduces the average ETE as compared to NOID and TBID, especially when the node count increases. These are three reasons for the reduction of end-to-end delay in ZMA:

1. **Avoiding blind and/or random MA migrations:** ZMA moves the MAs via the paths that are established according to a weighting function focusing on DC connectivity degree and distance to the event sources. By this, the MAs avoid unnecessary, blind and/or random migrations to find the location of event regions. This results in decreased path hop count and ETE.
2. **Increasing parallelism degree:** ZMA reduces ETE compared to TBID due to a higher degree of data aggregation parallelism from using a greater number of MAs at data regions. The MAs are initialised at ZMACs and move in parallel throughout the network to collect and aggregate data. This reduces ETE in ZMA.
3. **Hybrid routing:** ZMA utilises a hybrid routing scheme in which MAs move via proactively created links at each data region and then reactively establish routes to the sink. Owing to this, ETE is reduced using ZMA as compared to when only reactive routing is used.

According to Figure 5, ZMA has increased average ETE when the network is large and sparse. This is because ZMA has the ability to collect and aggregate a greater number of data samples comparing to the benchmark protocols (especially TBID) in sparse networks (see Figure 4).

5.4 Total MA Hop Count

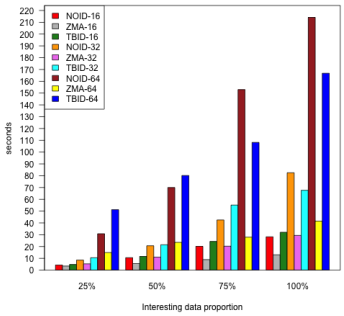
ZMA reduces the MA hop count as compared to the benchmark protocols when the node count increases. This is because of the ability of ZMA to avoid blind/random migrations and establish shortest paths to forward the MAs.

According to Figure 6, the MA hop count in ZMA is increased in comparison to the benchmark protocols, when the network is sparse. This is because ZMA can find and capture a greater number of source nodes compared to the benchmark protocols (Figure 4).

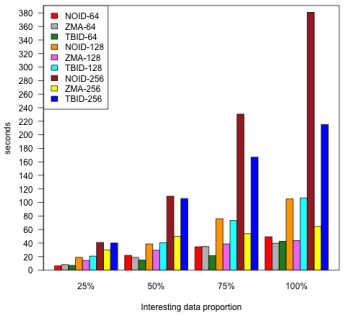
5.5 Total Transmitted Traffic

As Figure 7 shows, ZMA reduces the total transmitted traffic as compared to the benchmark protocols when the node count increases. The reason is that ZMA localises the network transmissions into the network zone and/or data regions. For example, control packets are transmitted (in multicast) between the nodes which reside in the same zone instead of any node which resides in the radio range of the sender node (broadcast).

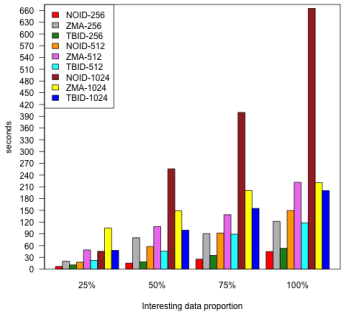
Fig. 5: End-to-end delay of MA routing protocols.



(a) small area (200×200)m^2



(b) medium area (400×400)m^2

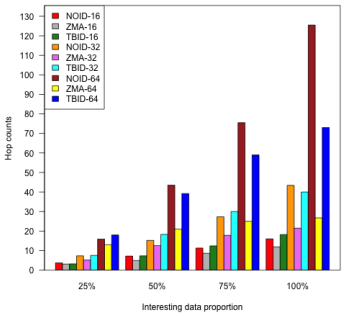


(c) large area (800×800)m^2

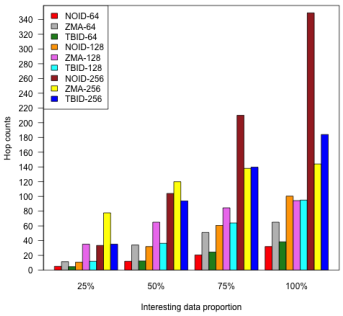
6 Conclusion and Future Works

ZMA performs well compared to NOID and TBID in terms of energy, accuracy and delay especially when the area size and the node count increase. It forwards the MAs to find, capture and aggregate desirable data samples from the source nodes which may be scattered in ER or RS model. ZMA has an overall satisfactory performance and satisfies its objectives for the following reasons:(1) Reduces routing overhearing: ZMA localises the MA routing communications into restricted data regions which are dynamically formed in a DC manner. This allows the sensor nodes to collect the required routing information locally (in multi-cast or unicast) to forward the MAs at each data region. This reduces the communication overhearing. (2) DC MA routing: ZMA avoids blind/random MA migrations and establishes only the paths which guide the MAs to the desirable source nodes. The paths are established in a data centric form and the MAs move through if a desirable source node needs to be visited. (3) Bottom-up MA migration: the MA journeys start to collect and aggregate

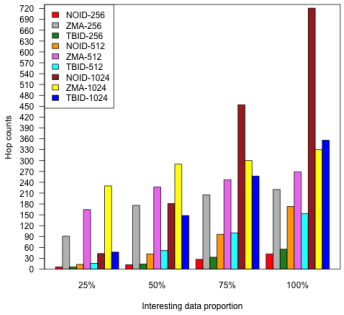
Fig. 6: Hop counts of MA routing protocols.



(a) small area (200×200)m^2



(b) medium area (400×400)m^2

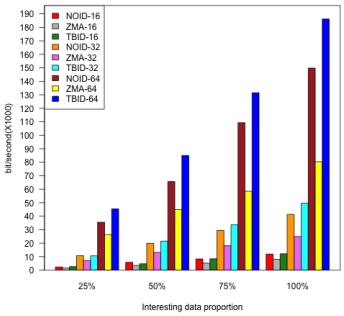


(c) large area (800×800)m^2

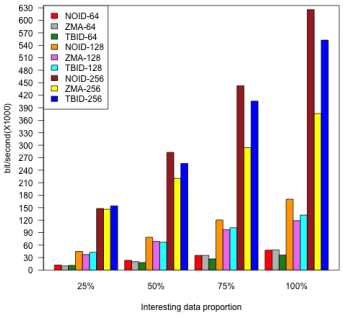
data samples from the ZMACs residing in the centre of event regions. ZMACs have the maximum connectivity degree with the desirable source nodes at each data region. (4) Forms data regions: ZMA forms a set of data regions by interconnecting the source nodes which have interesting data according to the sink queries. This limits the MA route search domain to the nodes which match the sink interests and are interconnected through DC links at each region.

In future, the correlations between energy, accuracy and delay need to be investigated. These correlations can form a triangle which influences the performance of data aggregation routing. (1) Consumed energy can be (positively) correlated with end-to-end delay as energy saving may result in increasing ETE. (2) Energy consumption may be increased if accuracy is increased. This is because of the increased number of data samples which are forwarded to the sink using MAs. (3) The performance of data aggregation routing protocols would not be efficiently evaluated if only ETE is considered. ETE is measured according to the received time of MAs at the sink. Hence, a routing protocol may have a lower ETE if only a few number of data samples (using

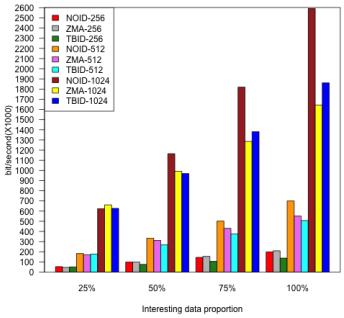
Fig. 7: Transmitted network traffic of MA routing protocols.



(a) small area (200×200)m^2



(b) medium area (400×400)m^2



(c) large area (800×800)m^2

MAs) are delivered to the sink. For this reason, the performance of data aggregation routing protocols needs to be evaluated according to the correlation between ETE and accuracy in which the protocols ability to minimise ETE while maximising accuracy is examined.

The performance of ZMA needs to be extended by considering unexpected node failures during the data aggregation procedure. As nodes fail suddenly, there is no way to inform the neighbour nodes in advance of the failure. Besides, wireless sensor nodes usually utilise a connection-less model of communication to transmit the network packets. Hence, a sender node never knows about a failure in its neighbourhood as no acknowledgement message is supposed to be received. Occasional routing infrastructure reconstruction can be a potential solution to deal with unexpected node failures. The sink asks the sensor node to reconstruct the routing infrastructure (i.e. data regions) at a set of specific periods.

Further research to remove the existing limitations of ZMA may be subjected to different results and contributions. ZMA assumptions such as LOS

model of signal propagation, noise-free environment and synchronised nodes are feasible for empirical scenarios. However, ZMA needs to be extended in order to fit real applications in which NLOS signal propagation model is used and the network is deployed in urban areas. The structure of ZMA might be slightly modified to address these changes.

References

1. X. Zhu and W. Zhang, "A mobile agent-based clustering data fusion algorithm in wsn," *International Journal of Electrical and Computer Engineering*, vol. 5(5), pp. 277–280, 2010.
2. E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *Wireless Communications*, vol. 14(2), pp. 70–87, 2007.
3. Y. Xu and H. Qi, "Mobile agent migration modeling and design for target tracking in wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 1, pp. 1–16, 2008.
4. P. K. Biswas, H. Qi, and Y. Xu, "Mobile-agent-based collaborative sensor fusion," *Information Fusion*, vol. 9, no. 3, pp. 399–411, 2008.
5. I. E. Venetis, G. Pantziou, D. Gavalas, and C. Konstantopoulos, "Benchmarking mobile agent itinerary planning algorithms for data aggregation on wsns," *The Sixth International Conf on Ubiquitous and Future Networks (ICUF), Shanghai, China, 8-11 July,*, pp. 105–110, 2014.
6. H. Qi and F. Wang, "Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks," *In the proceedings of the13th International Conference on Wireless Communication, Calgary, Alberta, Canada, July 9-11*, vol. 1, no. 1, pp. 147–153, 2001.
7. M. Chen, V. Leung, S. Mao, T. Kwon, and M. Li, "Energy-efficient itinerary planning for mobile agents in wireless sensor networks," *IEEE International Conference on Communications (ICC), Dresden, Germany, June 14-18*, vol. 1, no. 1, pp. 1–5, 2009.
8. D. Gavalas, A. Mpitziopoulos, G. Pantziou, and C. Konstantopoulos, "An approach for near-optimal distributed data fusion in wireless sensor networks," *Wireless Networks*, vol. 16, pp. 1407–1425, 2010.
9. C. Konstantopoulos, A. Mpitziopoulos, D. Gavalas, and G. Pantziou, "Effective determination of mobile agent itineraries for data aggregation on sensor networks," *IEEE transactions on knowledge and data engineering*, vol. 22, no. 12, pp. 1679–1693, 2010.
10. I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," *IEEE International Conference on Communications (ICC), Paris, France, June, 20-24*, vol. 6, pp. 3640–3645, 2004.
11. F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," *The 10th International Conference on Computer Communications and Networks, Scottsdale, Arizona, USA, October 15-17*, vol. 5, no. 2, pp. 304–309, 2001.
12. S. P. Ardakani, J. Padget, and M. D. Vos, "Hrts: A hierarchical reactive time synchronization protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 129, pp. 47–62, 2014.
13. C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA '99), New Orleans, Louisiana, USA, February 25-26*, vol. 1, no. 1, pp. 90–100, 1999.
14. I. D. Chakeres and E. M. Belding-Royer, "The utility of hello messages for determining link connectivity," *5th International Symposium on Wireless Personal Multimedia Communications (WPMC), Honolulu, Hawaii, October 27-30*, vol. 2, pp. 504–508, 2002.
15. S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, and D. Towsley, "Leader election algorithms for wireless ad hoc networks," *DARPA Information Survivability Conference and Exposition*, vol. 1, pp. 261–272, 2003.
16. J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, "Distance measurement model based on rssi in wsn," *Wireless Sensor Network*, vol. 2(8), pp. 606–611, 2010.

17. P. Uthansakul, M. E. Bialkowski, S. Durrani, K. Bialkowski, and A. Postula, "Effect of line of sight propagation on capacity of an indoor mimo system," *IEEE Antennas and Propagation Society International Symposium 2005, 3-8 July, Washington, DC*, pp. 707–710, 2005.
18. X. Shen, Z. Wang, P. Jiang, R. Lin, and Y. Sun, "Connectivity and rssi based localization scheme for wireless sensor networks," *International Conference on Intelligent Computing (ICIC'05), Hefei, China, August 23-26*, vol. 1, no. 2, pp. 578–587, 2005.
19. OMNET++, "Omnet++ simulator," 2012, http://www.omnetpp.org/, Retrieved (March 2016).
20. A. Viklund, "Mixim code," 2013, http://mixim.sourceforge.net/index.html, Retrieved (December, 2015).
21. A. Kopke, M. Swigulski, K. Wessel, D. Willkomm, P. K. Haneveld, T. Parker, O. Visser, H. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision," *the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (Simutools '08), Marseille, France, March 3-7*, vol. 1, no. 1, pp. 71–78, 2008.
22. Q. Wu, N. S. Rao, J. Barhen, S. S. Iyengar, V. K. Vaishnavi, H. Qi, and K. Chakrabarty, "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Transactions on Knowledge & Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.
23. M. Chen, S. Gonzalez, Y. Zhang, and V. C. Leung, "Multi-agent itinerary planning for sensor networks," *Quality of Service in Heterogeneous Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 22, pp. 584–597, 2009.
24. A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: an energyaccuracy trade-off," *Ad Hoc Networks*, vol. 1, no. 1, pp. 317–331, 2003.
25. M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V. C. M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 219, 2007.
26. S. S. A. Basurra, "Collision guided routing for ad-hoc mobile wireless networks," Ph.D. dissertation, Department of Computer Science, University of Bath, October 2012.
27. S. A. R. Zaidi, M. Hafeez, S. A. Khayam, D. Mclernon, M. Ghogho, and K. Kim, "On minimum cost coverage in wireless sensor networks," *The 43rd Annual Conference on Information Sciences and Systems (CISS 09), Johns Hopkins University, Baltimore, MD, March 18-20*, vol. 2, no. 3, pp. 213–218, 2009.
28. M. A. Youssef, A. Youssef, and M. F. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Transactions on parallel and distributed systems*, vol. 20, no. 12, pp. 1844–1856, December 2009.