

Research Article

WDARS: A Weighted Data Aggregation Routing Strategy with Minimum Link Cost in Event-Driven WSNs

Omar Adil Mahdi,^{1,2} Ainuddin Wahid Abdul Wahab,¹ Mohd Yamani Idna Idris,¹
Ammar Abu Znaid,¹ Yusor Rafid Bahar Al-Mayouf,^{2,3} and Suleman Khan¹

¹Faculty of Computer Science & Information Technology, University of Malaya, 50603 Lembah Pantai, Kuala Lumpur, Malaysia

²Department of Computer Sciences, College of Education for Pure Sciences-Ibn Al-Haytham, University of Baghdad, Baghdad, Iraq

³Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Correspondence should be addressed to Omar Adil Mahdi; omar_1980117@yahoo.com
and Ainuddin Wahid Abdul Wahab; ainuddin@um.edu.my

Received 15 March 2016; Revised 9 May 2016; Accepted 25 May 2016

Academic Editor: Fei Yu

Copyright © 2016 Omar Adil Mahdi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Realizing the full potential of wireless sensor networks (WSNs) highlights many design issues, particularly the trade-offs concerning multiple conflicting improvements such as maximizing the route overlapping for efficient data aggregation and minimizing the total link cost. While the issues of data aggregation routing protocols and link cost function in a WSNs have been comprehensively considered in the literature, a trade-off improvement between these two has not yet been addressed. In this paper, a comprehensive weight for trade-off between different objectives has been employed, the so-called weighted data aggregation routing strategy (WDARS) which aims to maximize the overlap routes for efficient data aggregation and link cost issues in cluster-based WSNs simultaneously. The proposed methodology is evaluated for energy consumption, network lifetime, throughput, and packet delivery ratio and compared with the InFRA and DRINA. These protocols are cluster-based routing protocols which only aim to maximize the overlap routes for efficient data aggregation. Analysis and simulation results revealed that the WDARS delivered a longer network lifetime with more proficient and reliable performance over other methods.

1. Introduction

Wireless sensor networks (WSNs) are formed by a collaboration of sensors through data sensing, processing, and wireless communication among the sensor nodes. These networks are organized for sensing event-driven information and transmitting it to the base station for in-depth evaluation [1–3]. WSNs have delivered beneficial outcomes in several applications such as environmental monitoring, surveillance missions, health monitoring, home automation, target tracking, traffic monitoring, fire management, agriculture monitoring, industrial failure detection, and energy management [4–8]. WSNs are often deployed in the form of thousands of nodes in remote and hostile areas which are inaccessible or unsafe for humans. Therefore, the formation of autonomous and energy efficient network among the sensor nodes becomes vital to ensure prolonged network lifetime and controlled energy depletion [9–12].

Energy efficiency is directly related to effective data routing wherein cluster of nodes is formed to reduce the energy consumption and control overhead while limiting the interference among the sensor nodes [13]. Generally, the energy is consumed during data sensing, processing, and transmission. Among these activities, data transmission consumes the most energy [14]. Thus, efficient data forwarding and processing techniques must be developed to extend the network lifetime. One possible solution is by using in-network data aggregation schemes (e.g., see [15]). This approach reduces a significant number of bytes transmitted during the network operation by aggregating data at intermediate nodes and thus helps in bandwidth and energy savings. Data aggregation involves combining data from various sources so that aggregated information is received at the base station and circulation of redundant information is eliminated. By employing data aggregation, the issues of

redundancy and numbers of transmissions are reduced. For execution of common tasks, the nodes within the network must communicate with each other or through intermediate nodes [16, 17].

To develop a data aggregation scheme, three main constituents of data aggregation should be considered, which are (1) aggregation function used by the protocol, (2) data aggregation scheduling which defines the waiting period before a node aggregates and forwards the received data, and (3) routing scheme which defines the routing protocol used to send the aggregated data towards the sink by generating a network structure [18]. This paper focuses on the routing scheme of data aggregation which potentially optimizes the routing procedure by utilizing the available processing capability of the intermediate sensor nodes.

The aggregation task in our network is achieved by formation of cluster-based data aggregation in a three-level hierarchy. This reduces the processing and communication cost for randomly distributed nodes. When in-network overlapping of routes occurs inside the cluster, the member nodes of that cluster perform aggregation (aggregation via cluster members). Moreover, aggregated data are sent to sink by the cluster head node (aggregation via cluster head). If overlapping of two or more events paths occurs outside the cluster, the relaying nodes will perform the data aggregation (aggregation via relay nodes).

In the context of WSNs, efficient in-network data aggregation will require an adaptive forwarding paradigm that allows intermediate nodes to forward the data packets towards the sink node through different paths. The paths are estimated based on comprehensive weights and choosing the next hop with the maximum overlap routes to ensure the efficiency of in-network data aggregation. This forwarding paradigm is different from the classic routing which only considers the shortest path between sources and sink nodes.

In event-driven WSNs, the monitoring capability deteriorates when the over-overlapping paths of uncorrelated events perform extensive data aggregation. Hence, the improved network performance is not obtained. Inefficient data aggregation neglects the network state and causes the early energy depletion of the backbone nodes and uneven network structure due to various dead nodes. Therefore, a balance between maximizing data aggregation and energy is necessary.

In this paper, a novel strategy of weighted data aggregation routing is proposed by analyzing the existing problems. The algorithm uses hop-tree to attain maximum data aggregation. To build and update hop-tree, the local state of the nodes is considered so that adaptive behavior can be obtained for event-driven WSNs. Moreover, the proposed strategy finds the ideal point for route overlapping through the shortest paths from events to sink and optimally balanced the energy consumption. Each node chooses its next hop to forward the data in accordance with innovative triple cost functions, which are distributed, adaptive, and comprehensive weights.

The rest of this paper is organized as follows: in Section 2, the related works are discussed. In Section 3, the network model and scenario assumptions are outlined, and the proposed methodology and strategy are presented in

detail. Section 4 discusses the performance of the proposed algorithm by comparing it with other approaches. Finally, in Section 5, conclusions are drawn and possible future directions are described.

2. Related Work

2.1. Flat Network Based Aggregation. In the literature, many earlier approaches employ the flat sensor networks for data aggregation [19, 20]. For instance, many studies have used parent and child association based simpler topology for tree-based data aggregation technique [21]. The data sent by the children are aggregated by the parent node which in turn sends it to its own parent node. The key restrictions of the tree-based data aggregation methodologies are discussed as follows: (1) this technique provides a simpler approach to aggregate data but results in a high latency because the data aggregation is not performed until the packets have arrived at the parent node or grandparent node. (2) High possibility of data is not aggregated near the event of interest because any two nodes that sense the same event might have different parent nodes. This reduces proficiency of data aggregation as the data transmitted over a long path to grandparent node. (3) The tree-based data aggregation schemes require a high number of control messages to build and update the routing tree which consumes more energy. (4) The prior construction of the tree is based on the assumption that the source nodes in the network are fixed and predetermined. Hence, it fails to exhibit the flexible behavior. (5) Its main drawback is when the packet loses due to bad channel links. In this case, the entire aggregated data from the children nodes are lost.

Hierarchical tree structures are costly to maintain and prone to damage due to limited network strength. However, they are still used in designing optimal data aggregation function, energy efficient network, and proficient data aggregation at intermediate nodes. For example, Li et al. developed the data aggregation protocol using Steiner minimum tree [22]. Data centric routing approach employs shortest path tree (SPT) routing protocol [23]. This algorithm is a simple approach to construct the trees in ad hoc fashion and promotes the energy awareness in the parent nodes. When an event is detected by any node, it uses the shortest path to transmit the information towards the sink. This condition is true if the overlapping paths of data aggregation occur (i.e., opportunistic data aggregation).

Issues in tree-based data aggregation due to correlation of sensed information have been considered in [24]. The authors demonstrated the data gathering problem as an NP-complete problem and found the ideal result to be between shortest path tree (SPT) and Traveling Salesman Problem (TSP). A hybrid scheme proposed by Park and Sivakumar [25] combines the shortest path tree and clustering in which the data are aggregated in each minimum dominating set by a head node and all head nodes are linked through a global shortest path tree.

In [26], an energy-aware spanning tree algorithm (Espan) has been proposed for data aggregation exhibiting the feature of energy awareness. This algorithm selects the source node with the highest available energy as the root while other nodes

use the residual energy and distance from the root node as metrics to select their parent node from the neighbors. However, the nodes will tend to select the neighboring nodes with the least distance to the root as parent nodes. This will cause a rapid energy depletion in the parent nodes with the least distance to the root and they will be failed sooner as compared to the other network nodes due to their frequent selection as parent nodes. To achieve a long network lifespan and eliminate the dead nodes in the network, an algorithm based on left over energy in node and distance parameters has been proposed in [27]. The node with the highest energy is selected as the parent node with a reasonable distance to the root. The energy along the path and length is used to maintain a balance between energy and distance parameters.

2.2. Cluster-Based Aggregation. Clustering is a well-established approach in hierarchical data aggregation. This method involves division of network into small sets of nodes called clusters. Within each cluster, the hierarchy is divided into a cluster head node and member nodes [28]. The data from the member nodes are collected by the cluster head. Then, the data are aggregated and forwarded to the upstream node. The clustering algorithms can be either a static or dynamic.

The static clustering is the clusters that are formed prior to network operation [29–31] and based on network parameters (e.g., the remaining energy in the nodes [32] and physical distance as in the Voronoi diagram-based method in [33]). Moreover, the reestablishments or updates of cluster do not occur adaptively. LEACH [34] and HEED [35] are two classic models of static clustering. They differ in the selection method of cluster head as follows; LEACH is formulated on the assumption that energy of all nodes is equal during the election while HEED considers the variation of energy in nodes to optimize the network lifetime.

A dynamic cluster architecture [33, 36, 37] is formed reactively within the proximity of the event sensing nodes. Once the event is located, a specific sensor node is chosen as a cluster head (ideally the node with the maximum energy or adjacent to the event) and the nodes that are one hop away are assigned as the member nodes. The main benefit of this approach is that only the participated nodes are active in the aggregation of the data. Thereby, it conserves the energy of the idle nodes.

Nakamura et al. [38] discussed the reactive algorithm of the Information Fusion Based Role Assignment (InFRA). The roles such as sink, collaborator, coordinator, and relay are assigned when any event takes place. In this protocol, clusters are formed when similar event is detected by various nodes. Then, the coordinator aggregates the data from all collaborated clusters and sends the event data towards the sink in multihop fashion. InFRA discovers the shortest path tree linking all source nodes to sink in a manner that the intra cluster data aggregation is possible. InFRA provides a role migration policy; that is, role of coordinator is transferred from one node to another so that the load of energy consumption is distributed evenly between nodes in the cluster. InFRA used intracluster and intercluster in its data aggregation schemes. A disadvantage of InFRA is that, each time a new

event is detected, the information of the event is broadcasted all over the network to notify other nodes and the paths from the available coordinators to the sink node are updated. These processes are costly and limit the network scalability.

Data Routing for In-Network Aggregation (DRINA) in WSNs [39] provides reliable and improved data aggregation. It reduces the control overhead for building routing trees and maximizes the formation of overlapping paths. The main aim of DRINA is to reduce the energy depletion and minimize the message exchanges during the network operation. However, there are few disadvantages in DRINA as follows. (1) Lacks of load balance, a heavy load in the nodes on the prior built path, will cause those nodes to expire prematurely. (2) Correlated events are ignored due to the assumption that the data from differing event areas could be aggregated adequately. (3) Sometimes the data have to be routed over the lengthier paths, which increase the total energy depletion.

In this work, we proposed a novel algorithm for in-network data aggregation which takes into account the trade-offs between routes overlapping and total link cost for data transmission. Further, the proposed algorithm exploits the local node state to construct and update the hop-tree for efficient data aggregation and efficient control of energy consumption.

3. WDARS: Weighted Data Aggregation Routing Strategy

3.1. The Network Model and Scenario Assumptions. In this study, we consider the features of sensor nodes in the simulated scenario as follows:

- (1) A 2D space has been filled with randomly deployed sensor nodes. The nodes exhibit static and homogeneous behavior in terms of storage, processing abilities, battery power, sensing, and communication capabilities.
- (2) A symmetric radio channel has been considered for modeling so that the energy needed to conduct a data transmission from sensor node S_i to sensor node S_j is equal to the energy used for the same data transmission from sensor node S_j to sensor node S_i .
- (3) The single base station is considered to be at a distant location from the sensor field. It is connected to the power supply while the sensor nodes are non-rechargeable and may die after their energy are exhausted.
- (4) Every node possesses a unique ID and forwards the data at any time during the network operation.
- (5) It is assumed that every node has the capability to calculate its remaining energy and existing buffer size (available memory to store the data before being serviced).
- (6) The network employs a dynamic cluster architecture. A cluster is formed reactively within the proximity of the sensing nodes event and terminated at the end of the event.

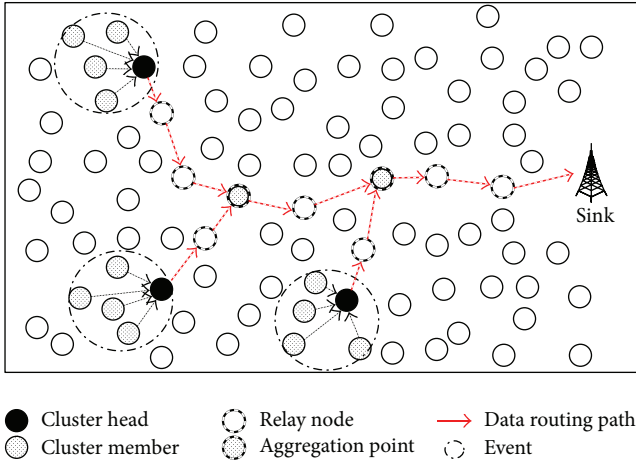


FIGURE 1: Network model diagrams for routing towards the sink node.

3.2. Proposed Model. The event-driven applications are often used in a wide network and enforced dissimilar load to the various parts of the network. This is due to the arbitrarily occurrence of the events. In order to consider such applications, the protocol should be designed to adopt ad hoc features, energy efficiency (i.e., effective management of energy resource for each node), maximum connectivity, a simple controlled processing, and transmissions.

Most existing in-network data aggregation protocols maximize the route overlapping for efficiency of data aggregation. However, such approach may adversely affect the network stability. Therefore, we proposed a protocol to provide a trade-off between the data aggregation cost and the total link cost to solve the excessive route overlapping problem which could cause transfer of the data along longer paths and unbalance of data load on the backbone nodes. The proposed protocol builds a fully distributed cluster and efficient routing tree with the maximum energy conservation and congestion avoidance. It connects all the sensor nodes that detect the event to the sink while maximizing the data aggregation. Also, the proposed protocol optimally balanced the energy depletion paths leading to the sink from the cluster head. Figure 1 depicts the suggested approach and the roles in the routing arrangement are described as follows:

- (i) *Cluster Member (CM)*. This node is responsible for the discovery of an event and forwarding the gathered data to the cluster head.
- (ii) *Cluster Head (CH)*. The responsibility of the cluster head includes the event detection and it performs the data aggregation. Then, the gathered data are transmitted towards the sink.
- (iii) *Relay*. It is a node whose duty is to forward the received data towards its next possible hop. In some cases, relay nodes represent a data aggregation point when the data paths are overlapped on it.
- (iv) *Sink*. It is a collection of nodes or personal computers having high computational energy and processing

TABLE 1: The header of HCM.

Number	Parameter	Description
1	Node-ID	Identification of the node that transmitted/retransmitted the HCM
2	Type	Description of HCM messages
3	HtT	The distance from the node to the hop-tree (in hops)
4	HtS	The distance from the node to the sink (in hops)
5.1	ER	Energy residual of the node
5.2	AB	Available buffer memory size of the node

capability. The sink is liable to receive all the data from the cluster head and other member nodes.

The algorithm proposed in this study consists of three phases. The first phase involves establishment of a hop-tree between sensor nodes and sink. The second phase starts as soon as any event is sensed by a node. In this phase, formation of clusters and selection of cluster head take place. In the third phase establishment of routes, data aggregation, and routing process take place.

3.2.1. Phase I: Hop-Tree Building Process. The input of the initialization phase is a set of nodes which are deployed in the predetermined sensor field. Consequently, each node will identify its neighbors as possible parents within its radio frequency (broadcast) region, hop distance to reach the sink, their residual energy, and available buffer size. The initialization algorithm begins by broadcasting a Hop Configuration Message (HCM) from the sink to all the sensors in the network (Step 1, Algorithm 1). In addition to the common message fields, it contains five key parameters including Node-ID, Type, Hop-to-Tree (HtT), Hop-to-Sink (HtS), and Status (ER, AB) as explained in Table 1.

In addition to Hop-to-Tree, each node has Hop-to-Sink parameter that maintains a minimum number of hops between the node and the sink. At the beginning of the tree formation, the same values are assigned to Hop-to-Tree and Hop-to-Sink. The value of Hop-to-Tree parameter change immediately after the first event is detected. It will continuously change with the occurrence of the new events. Contrarily, the value of Hop-to-Sink remains the same in every node. However, Hop-to-Tree of any node may change due to the occurrence of following two events: (i) the member node is included in the backbone structure, which is the Hop-to-Tree of the sink node, and other nodes belonging to the backbone structure are zero. (ii) a HCM is received by the member node and gives a more accurate information about the distance.

At the beginning of the process, when the hop-tree begins to form, the value of HtT at sink node is stored as zero and infinity for other nodes, the node energy is set to actual value, and the node available buffer memory size is conceded maximum. Once the neighboring nodes of the sink receive the HCM (Step 2, Algorithm 1), a node performs the following tasks: verifies if its HtT value is greater than the

value of HtT in the HCM message (Step 3.1, Algorithm 1); this condition will guarantee that each node records the minimum number of hops to the sink. Depending on the validity of the condition, the node maintains the information of its neighbors whose HCM are received in neighbors table (Step 3.1.1, Algorithm 1). This node also updates the routing table as stated in Steps 3.1.2 and 3.1.3 in Algorithm 1 by exploiting the weights function to compute the link cost of their next hop neighbors and selects the node with the lowest cost as its next hop is depending on (4). This follows incrementing the values of HtT and HtS by one in a sensor node. The sensor node then computes its residual energy after one complete transmission and updates the ER field. Moreover, it computes the obtainable buffer size and updates the AB field and finally circulates the HCM to further neighbors, as shown in Algorithm 1 (Steps 3.1.4–3.1.11).

Otherwise, in Step 3.2, Algorithm 1, the HCM message will be dropped if the condition in Step 3.1, Algorithm 1, is false, which indicates the stored path is the shortest distance to sink.

This procedure is repeated until all the nodes in the network join the tree topology, with the sink node as the root node of the tree.

The weights for packet transmission from node x to node y are defined as follows:

$$W_1 = \left(1 - \frac{E_{\text{res}}(y)}{E_{\text{init}}(y)}\right)^2 + \left(1 - \frac{B_{\text{ava}}(y)}{B_{\text{total}}(y)}\right)^2 \quad (1)$$

$$W_2 = \frac{((\text{HtT}(y) - \text{HtT}(x)) + 1)}{\text{HtT}(y)} \quad (2)$$

$$W_3 = \frac{((\text{HtS}(y) - \text{HtS}(x)) + 1)}{\text{HtS}(y)} \quad (3)$$

$$W_f = \text{alpha} * W_1 + \text{beta} * W_2 + \text{meu} * W_3 \quad (4)$$

$$\text{alpha} + \text{beta} + \text{meu} = 1. \quad (5)$$

The weight (W_1) consists of leftover energy and average buffer size of node. In the first part of (1), when the remaining energy of node y reduces, the result approaches 1. Conversely, when the remaining energy is high, the resulting value approaches zero and the cost reduces. Furthermore, if the node energy does not change (i.e., same as the starting energy), zero cost will be obtained. Likewise, in the second part of the equation, when the buffer is spacious, the cost approaches 0 and when the buffer size has reached its maximum capacity, the cost approaches 1.

Further, W_2 and W_3 are the distance based on number of hops to calculate the next hop neighbor. These weights have the same initial values during the establishment of the tree. In (2), when the node y is one hop closer to already established path, 0 cost is obtained. If the node y is far from the earlier

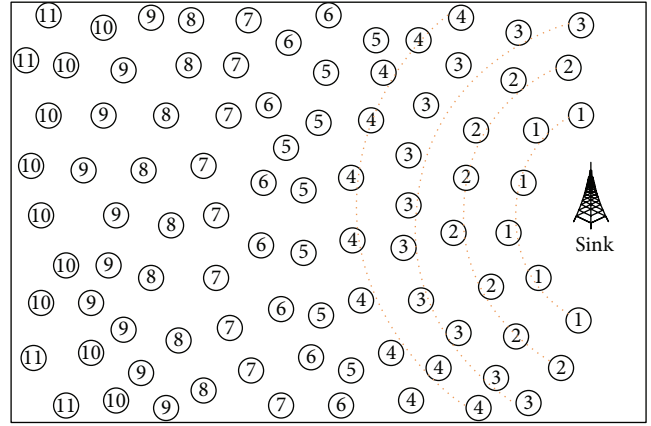


FIGURE 2: The hop-tree building process.

path, the cost is higher than 0, and if the hop distance is the same, the cost is 1. The same principle is applied to (3), but the variables are representing the distance to the sink. The weights are represented by alpha, beta, meu and their summation is equal to unity. The final weight (W_f) is obtained by combining W_1 , W_2 , and W_3 together, which represents the overall cost of the packet transmission from node x to node y .

Figure 2 shows the hop-tree building process, where the labels in the sensors indicate the Hop-to-Tree in increasing order as it moves away from the sink.

Algorithm 1. Hop-tree building process.

Step 1. The sink node broadcasts the initialization message HCM.

Step 2. “ N ” is the set of nodes in a network that receive HCM such that $x \in N$.

// x represents any member node

Step 3. Foreach $x \in N$

Step 3.1. If $\text{HtT}(x) > \text{HtT}(\text{received HCM})(x)$
← true then

Step 3.1.1. Insert Neighbortable (Node-ID, HtT, HtS, ER and BM);

Step 3.1.2. Node x compute the final link cost (W_f) for $NE(x)$;

// $NE(x_i) = \{x_j / d(l_i, l_j) \leq r_c \text{ and } l_j \leq l_i\}$ is the set of neighbor nodes of sensor node x_i , where l_i is the location of x_i and $d(l_i, l_j)$ is the Euclidean distance between x_i and x_j

Step 3.1.3. NextHop(x) ← ID(Bestneighbour);

//Bestneighbour = The neighbour with smallest weight

Step 3.1.4. $\text{HtT}(x) \leftarrow \text{HtT}(\text{HCM}) + 1$;

Step 3.1.5. $\text{HtS}(x) \leftarrow \text{HtT}(\text{HCM}) + 1$;

Step 3.1.6. Update the HCM;

Step 3.1.7. ID(HCM) ← ID(x);

Step 3.1.8. $\text{HtT}(\text{HCM}) \leftarrow \text{HtT}(x)$;

Step 3.1.9. $\text{HtS}(\text{HCM}) \leftarrow \text{HtS}(x)$;

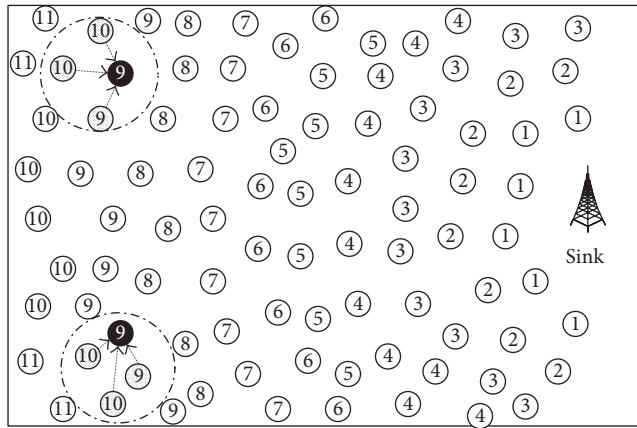


FIGURE 3: Example of cluster formation.

Step 3.1.10. Status (HCM) \leftarrow Status (x);
 //State contain ER & AB
 Step 3.1.11. Broadcast (HCM);
 Step 3.2. Else Discard HCM message;
 Step 3.3. End.

Step 4. End.

3.2.2. Phase II: Event-Driven Cluster Formation and Cluster Head Election. In phase II, a formation of dynamic cluster architecture takes place. Once the event is located, all the nodes that detect the event will be input to the cluster formation algorithm. At the end, the algorithm will choose a specific sensor node as the cluster head (ideally the one with the minimum weight cost) while nodes within the range of the cluster head will be designated as member nodes. In this process, any node that has detected the event sets its role to cluster head and calculates its weight (W_f) as stated in Steps 1.1 and 1.2 in Algorithm 2. Then, all event nodes propagate their information by Cluster Configuration Message (CCM) (Step 1.3, Algorithm 2), where CCM also constitutes quantities (Node-ID, Type, HtT, HtS, and Status). Cluster head is selected considering the final weight function (4) and minimum weight node will be assigned as the cluster head (Step 3, Algorithm 2). Finally, the decision is broadcasted to the all member nodes. The member nodes then remember their cluster head and all the events detection reports are sent directly to the cluster head. In the case of two or more concurrent nodes having the same weight, the node with the lowest ID is considered eligible, as shown in Step 3.2, Algorithm 2.

Every cluster head is liable to collect information from the member nodes and forward the aggregated data towards sink. Figure 3 shows an example of cluster formation.

Algorithm 2. Event-driven cluster formation and cluster head election.

Input. S // S is the set of nodes that detects the event.

Output. u // A node of the set S is elected as a cluster head.

Step 1. Foreach $x \in S$ do

Step 1.1. Role(x) \leftarrow Cluster head;

Step 1.2. $W_f(x) \leftarrow \alpha * W_1 + \beta * W_2 + \mu * W_3$;

//Each node calculate its weight W_f .

Step 1.3. Node x broadcast (CCM);

//Each node after create CCM broadcast it to its one hop neighbors.

Step 1.4. End.

Step 2. " N " is the set of nodes in a network that received CCM and detects the event.

Step 3. Foreach $u \in N$

Step 3.1. If $W_f(u) > W_f(\text{CCM})$ then

Step 3.1.1. Role(u) \leftarrow Member Node;

Step 3.1.2. Node u retransmits the CCM received from x ;

Step 3.2. ElseIf $W_f(u) == W_f(\text{CCM})$ && ID(u) > ID(CCM) then

Step 3.2.1. Role(u) \leftarrow Member Node;

Step 3.2.2. Node u retransmits the CCM received from x ;

Step 3.3. Else Node u discards the CCM received from x ;

Step 4. End.

3.2.3. Phase III. Route Establishment in Consideration of Nodes Weight. In phase III, a group of cluster heads which were established during phase II is considered as input and the new route to transfer the event data will be the outcome of this process. In this phase, routing tree formation is based on the saved weights in neighbor table which was created during Phase I. Each node will be well aware of all its neighbors, the possible parent nodes within its radio range, number of hops required to approach sink or former routing path, remaining energy of those nodes, and obtainable buffer memory. This implies that each node can use the information in neighbors table to send the data packet to the sink node.

In the first step of this process, the cluster head is now responsible for routing tree formation and routing packets of the new event towards the sink (Step 1, Algorithm 3). The cluster head will check if its HtT is zero and this means it is apart from the backbone of the hop-tree, so creating a fresh route as hop-trees' new backbone is not required (Step 2, Algorithm 3).

In order to acquire even energy dissipation and avoid congestion delay which is caused by data collisions, the WDARS scheme keeps track of the remaining energy level and accessible buffer memory of the nodes in the backbone, which if it gets larger than the set weight limit, the new

routing path formation is initiated, as shown in Step 3, Algorithm 3. During the reformation process, the neighbor node that relatively has less HtT and HtS, higher energy level, and available buffer memory in comparison to other candidates is chosen as the alternative next hope. Also in the reformation of routing path process, the weight-threshold level for (W_1) in every node is slightly increased if no suitable node can be found.

The cluster head starts to create a route establishment message (REM) and sends the REM to its next hop as shown in Step 4, Algorithm 3. If a route establishment message is received by the next hop node, it will retransmit and initiate the process of updating the hop-tree (Steps 5 and 6, Algorithm 3). The repetition of these steps occurs until the sink is approached or a node that took part in a formerly constructed route is discovered. The creation of routes occurs by selecting the best neighbor at every hop.

The updating of hop-tree is required so that all source nodes can be connected via shortest paths, data aggregation can be maximized, and energy dissipation can be balanced as events befall in succession. In the proposed scheme, HtT, HtS, ER, and AB values are updated at each node to fulfill these objectives. Correlation among data is of great concern in data aggregation and hence higher degree of correlation generates better results. Spatial distance between nodes determines the spatial correlation of sensed data by nodes. So, if the two events are closer, the sensed data is highly correlated whereas when the events are far apart the sensed data exhibits lower degree of correlation. Usually, efficient data aggregation could not be performed for events that are far apart. In WDARS, the purpose of using varied weights especially for HtT and HtS in route establishment is to achieve the ideal point for path overlapping for different events. When events are close, the data path will overlap as early as possible at the nearest ideal point for aggregating event's data. In contrast, when the events are far apart the data routes overlapping will occur at the point where the shortest path will be selected to reach the destination and avoid transferring the data over the long tracks.

Algorithm 3. Routing establishment in consideration of nodes weight.

Step 1. The leader node v of new event starts to establish routing tree;

Step 2. If $HtT(v) == 0$ then

Step 2.1. Start send data to its NextHop(v);

//There is no need to routing formation; node v is already on the backbone of the Hop-Tree.

Step 3. If $W_1(\text{NextHop}(v)) > \text{Weight-Threshold}$ then

Step 3.1. Node v find a new NextHop with less weight that satisfy the predefined limits by exploiting the neighbors table.

Step 4. The leader node v sending REM to its next hop.

Step 5. Repeat

Step 5.1. Node x is NextHop(v) that received REM.

Step 5.2. $HtT(x) \leftarrow 0$;

//node x becomes a part from new routing structure

Step 5.3. $\text{Role}(x) \leftarrow \text{Relay}$;

Step 5.4. If $W_1(\text{NextHop}(x)) > \text{Weight-Threshold}$ then

Step 5.4.1. Node x find a new NextHop with less weight that satisfy the predefined limits by exploiting the neighbors table.

Step 5.5. Node x send route establishment message REM to its NextHop(x);

Step 5.6. Node x broadcasts the HCM with value of $HtT = 0$;

Step 6. Until finds the sink node or a node belongs to the routing structure for previous event.

Step 7. End.

Figure 4(a) shows the formation of the new path and the updated hop-tree based on HtT only [39]. Their technique forces the paths from the events that occur far apart to overlap and aggregate the data from two consistent events. This results in a poor data formation, the overload in overlapping paths that causes imbalance of energy consumption in the network, and increase in the number of dead nodes. Moreover, the data transmission over the lengthier path will boost the total energy consumption. In WDARS, the state of route overlapping depends on HtT and HtS as an addition to the node status as shown in Figure 4(b). It can be observed that the route overlapping is linearly related to the distance between the two events and guaranteed the data transmission over the shortest route possible.

4. Simulation Results and Performance Evaluation

4.1. Simulation Environment. The simulation of the proposed model was conducted using MATLAB with IEEE 802.15.4 as the MAC layer protocol. This simulation was based on a wireless sensor network in event-based environment with an area of $500 \text{ m} \times 500 \text{ m}$ in which the nodes are randomly placed. If two sensors are within the range of each other, they are considered neighbors. A circular event space is considered with random position, time, and event duration. The size of the nodes varies from [100, 300] with a step size of 50 nodes while the sensor field, communication range, and event radius remain constant. As the number of nodes increases, the time taken for simulation also increases gradually. All parameters used in this simulation are listed in Table 2.

4.1.1. Description of Adopted Energy Consumption Model. The energy consumption model can be categorized into three energy consumption submodules: communication, computation, and sensing [14]. Among these activities, the communication phase consumes a significant amount of energy

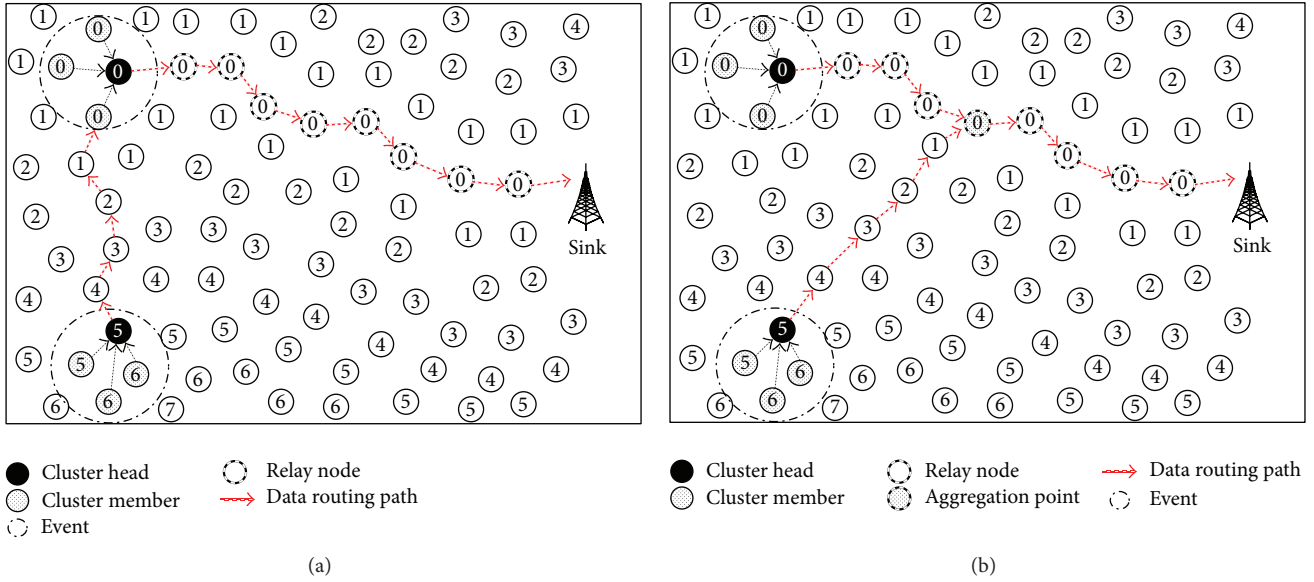


FIGURE 4: (a) and (b) The formation of the new path based on the former hop-tree when another event happens for DRINA and WDARS, respectively.

TABLE 2: Simulation parameter.

Parameter	Value
Channel	Wireless channel
Sink node	Fixed coordinates
Deployment area	500 m × 500 m
Antenna	Omni directional
Initial node energy	2 J
Number of sensor nodes	100, 150, 200, 250, 300
Number of events	3
Event radius	80 m
Communication radius	80 m
Simulation time (sec)	3000
Topology	Dynamic cluster
Bandwidth	10 Mbps
Data packet size	1024 byte
Control packet size	56 byte
Alpha	0.250
Beta	0.375
Meu	0.375

in the sensor nodes depending on the distance between source and destination node. In this simulation, we utilize the radio transceiver model as in [40]. According to the communication energy consumption model in Figure 5, the energy consumed to transmit an ℓ -bit packet over a distance d by the radio can be described in the following equation:

$$E_{Tx}(\ell, d) = \ell * E_{\text{node (Tx)}} + \ell * \epsilon_{\text{op-amp}} * d^n. \quad (6)$$

Furthermore, a sensor node is responsible for forwarding other sensor nodes' data. In this procedure, the data packets are received by the sensor nodes and their energy is not

affected by variation of distance between communication pairs. Therefore, the energy required for receiving an ℓ -bits packet over a distance d is given by the following equation:

$$E_{Rx}(\ell) = \ell * E_{\text{node (Rx)}}. \quad (7)$$

The sensing structure of each sensor node is activated and data is collected from the surroundings by consuming sensing energy denoted by E_s . It is assumed that a constant amount of energy $E_{\text{node (sensing)}}$ is dissipated to sense one bit. Hence, the overall energy dissipation for ℓ -bits is given by the following equation:

$$E_s = E_{\text{node (sensing)}} * \ell. \quad (8)$$

Finally, the sensor network has been structured into distributed clusters. The sensor nodes not only forward the data but also contribute in making key decisions regarding the network operation. Furthermore, the computation of data aggregation consumes extra energy $E_{\text{node (aggregation)}}$ but it is relatively low in comparison to the energy dissipated during communication [14]. For simplicity, we assume the energy dissipated for computation and sensing energy are the same and equal to 50 nJ/bit, while communication (sending/receiving) is assumed to be 100 nJ/bit. The following list defines the meaning of different energy terms in this work and their typical values.

Energy consumption model parameters are presented as follows.

Term and meaning are given:

$E_{\text{node (Tx)}}$: energy consumed in transmitter electronics per bit (taken to be 100 nJ/bit);

$E_{\text{node (Rx)}}$: energy consumed in receiver electronics per bit (taken to be 100 nJ/bit);

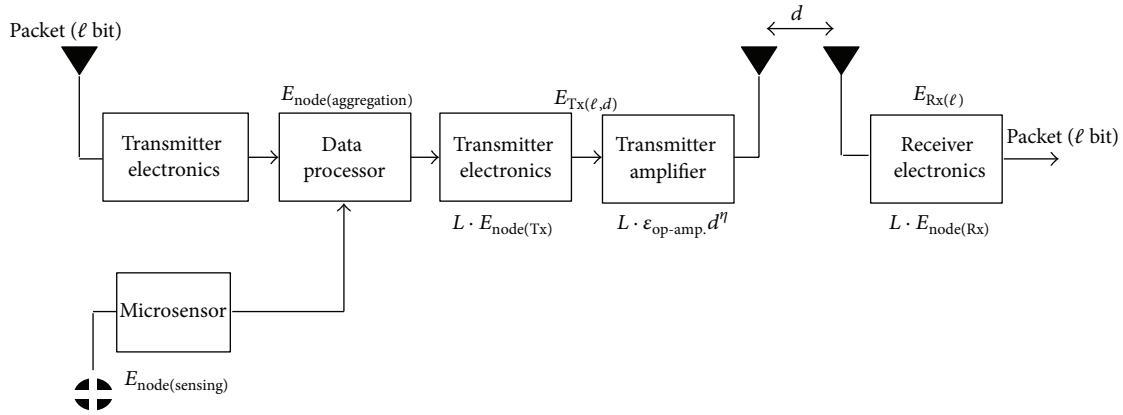


FIGURE 5: Energy consumption model.

$E_{\text{node (sensing)}}$: energy consumed in sensing one bit (taken to be 50 nJ/bit);

$E_{\text{node (agggregation)}}$: energy consumed for aggregating one bit (taken to be 50 nJ/bit/signal);

$\epsilon_{\text{op-amp}}$: energy consumed in the transmitter amplifier for either a free-space channel or a multipath fading channel (taken to be 10 pJ/bit/m² and 0.0013 pJ/bit/m⁴, resp.);

d : distance that the data has to travel;

l : data packet size in bits (taken to be 8192 bit);

η : path loss exponent that ranges between 2 and 6.

4.2. Performance Evaluation. The performance of the proposed WDARS has been evaluated with various network test cases and compared against DRINA and InFRA protocols. Figure 6 illustrates the energy consumption of the nodes for various network densities between WDARS, DRINA, and InFRA protocols. The InFRA recorded the highest energy consumption followed by DRINA and WDARS. This is due to the fact that the InFRA sends more control packets throughout the network during the event detection to inform the nodes and update the routes from existing coordinators to the sink node. This process results in a high communication cost in InFRA. In DRINA, lengthier paths are formed due to unwanted overlapping of distant and uncorrelated events. Therefore, a significant energy is wasted by forwarding the data through the longer paths. On the contrary, the suggested methodology of WDARS consumes less energy by taking into consideration the residual energy of the nodes to stabilize the energy consumption among the nodes. Moreover, the weights of HtT and HtS play an important role in energy saving through the selection of the proper overlapping points. Consequently, a balanced trade-off between the data aggregation and link cost is achieved. The average of the total energy consumption for all tested protocols is depicted in Figure 7. The proposed WDARS recorded the lowest energy consumption with 11.65% compared to the DRINA (21.75%) and the InFRA (35.71%).

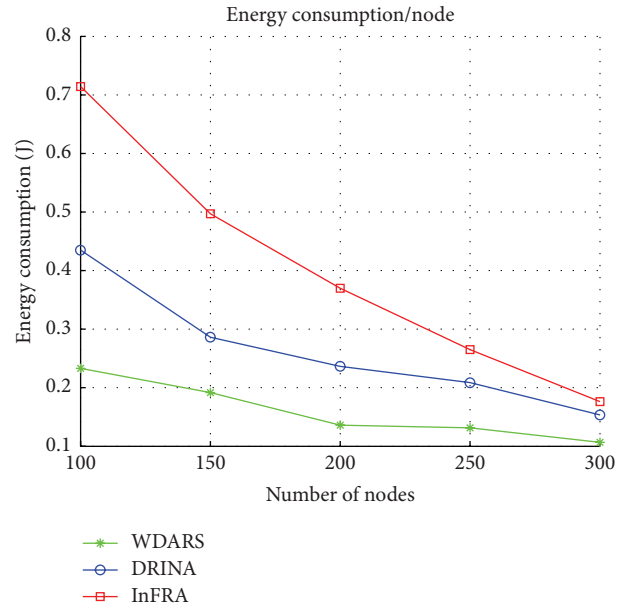


FIGURE 6: Comparison of average energy consumption for different node density.

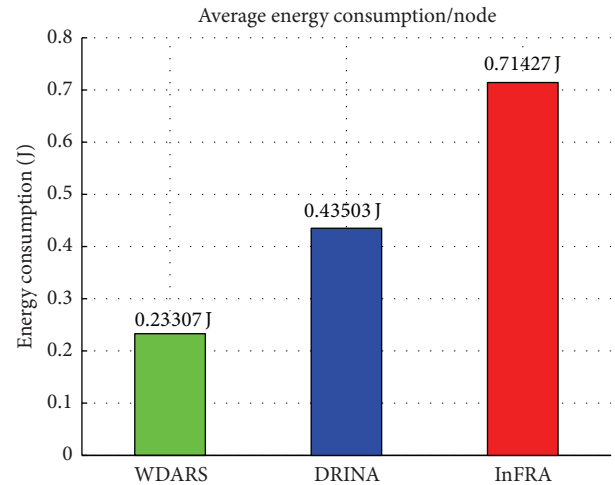


FIGURE 7: Comparison of total energy consumption.

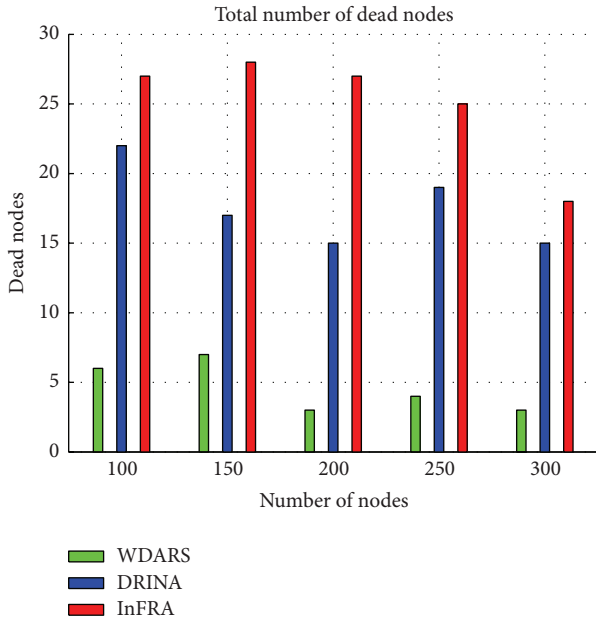


FIGURE 8: Comparison of the dead nodes number.

Next, the rates of dead nodes for each algorithm during the network operation were evaluated and compared as shown in Figure 8. Again, the WDARS had the lowest number of the dead nodes followed by the DRINA and InFRA in ascending order. This can be explained as follows: InFRA ignores the node status in decision to forward the nodes. Only the shortest path tree is considered to link the source and sink node and enable the intracluster data aggregation. Consequently, the nodes which are located at the shortest path are extensively utilized during the data transmission. This causes excessive energy consumption by those nodes and in turn generates a lot of dead nodes in the network. In DRINA, the maximization of the route overlapping is irrespective to the actual status of the network nodes and the spatial correlation between the events. Such approach has placed an immense load on the overlapping area and a very low aggregation ratio. As a result, the energy of the nodes in these areas is rapidly depleted and contributed to the early death of the backbone nodes. Contrarily, WDARS minimized the dead nodes by setting the weights rationally. Furthermore, the scenario of low hop distance and extensive energy depletion were successfully avoided in WDARS. Thus, the energy consumption of the nodes in the network is well balanced.

The network lifetime is highly depending on the routing protocol and influence by two factors: energy consumed over time and the initial node energy. If the initial energy capacity of a node is E and the energy consumed by each node is e_i , then the time up to which the node can work is given as ($t_i = E/e_i$). All such arrays of t_i provide us with the information about network lifetime. As depicted in Figure 9, the network lifetime of all algorithms increased as the node density increased. The lifetime of the WDARS exceeded other algorithms at all node densities. When the node density was minimum, the difference of lifetime between WDARS and DRINA and InFRA was 8.89% and 9.44%, respectively. At

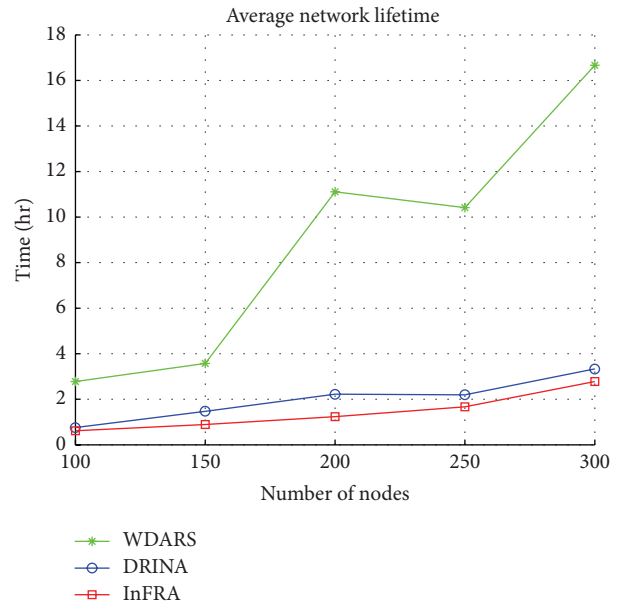


FIGURE 9: Comparison of network lifetime.

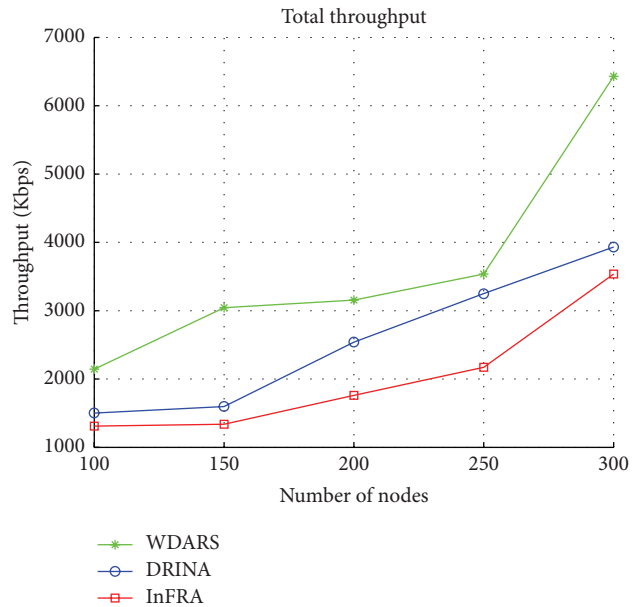


FIGURE 10: Comparison of throughput.

node size of 300, the WDARS scored 91.11% which was higher as compared to DRINA and InFRA with 75.55% and 77.78%, respectively. Moreover, the approach of the energy balance adopted by the WDARS appears to be beneficial with the increase of network density.

Figure 10 shows the throughput of WDARS, DRINA, and InFRA in different network density. It can be observed that the total throughput is increased with the increasing of node density. Overall, the WDARS achieved 8.22% and 13.07% more throughput at the lowest density network and 39.31% and 41.90% at the highest density network as compared with DRINA and InFRA, respectively. Such performance of

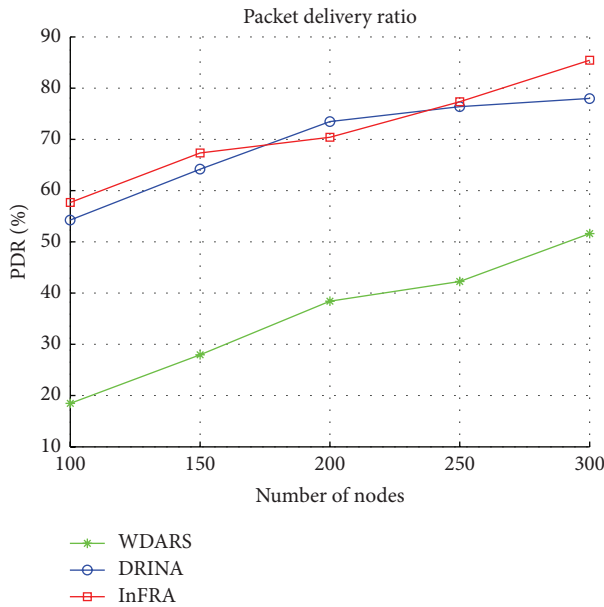


FIGURE 11: Comparison of packet delivery ratio.

WDARS is contributed by its protocol that tracks the level of the available buffer memory in the backbone and effectively avoids the congestion delay caused by data collisions.

Another evaluation metric that quantifies the quality of routing tree built by the algorithms is measured by the packet delivery ratio (PDR). A low PDR implies a greater aggregation rate of the built tree. In Figure 11, the PDR for all algorithms is increased as the network density increased. At the lowest network density, the WDARS has low PDR with 36.10% and 39.80% whereas at highest density it has a 26.50% and 33.10% as compared to DRINA and InFRA, respectively. The WDARS selects the ideal point for routes overlapping which results in fewer relay nodes in the routing tree as compared to DRINA and InFRA for varying network density. The perfect aggregation results were achieved via minimal Steiner tree [23].

5. Conclusion and Future Recommendations

This paper presented a multiobjective clustered-based routing protocol by utilizing the comprehensive weights. The proposed WDARS, successfully maximized the route overlapping via ideal aggregation point while simultaneously ensuring the data transmission across a lightweight route in WSNs. The performance of WDARS was compared with well-known DRINA and InFRA algorithms in terms of scalability, aggregation rate, data delivery rate, energy consumption, and network lifetime. For a set of different test instances, the simulation results showed that WDARS-based clustering protocol outperformed the DRINA and InFRA. Also, our WDARS attained a better performance in the network lifetime by providing a route structure with lower energy consumption for in-network communication. Furthermore, WDARS demonstrated an acceptable level in balancing the energy load among all nodes. This reflects that the proposed algorithm

can provide considering aggregation cost. Also, a trade-off between network load and network lifetime was established while considering residual node energy and obtainable buffer size. Other improvements in route structures exhibiting delay and fault tolerance abilities will be investigated in future work.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

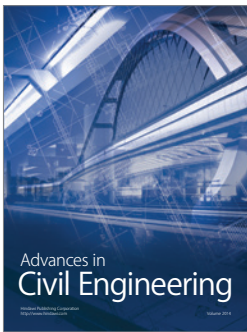
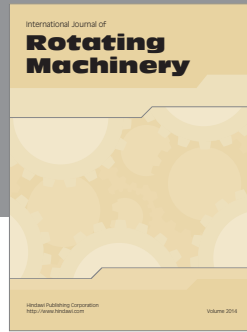
Acknowledgments

This research is supported by Research Grant RP036(A,B,C)-15AET and HIR MoE Grant UM.C/625/1/HIR/MoE/FCSIT/17 from the Ministry of Education Malaysia.

References

- [1] K. Matrouk and B. Landfeldt, "RETT-gen: a globally efficient routing protocol for wireless sensor networks by equalising sensor energy and avoiding energy holes," *Ad Hoc Networks*, vol. 7, no. 3, pp. 514–536, 2009.
- [2] M. H. Anisi, G. Abdul-Salaam, M. Y. I. Idris, A. W. A. Wahab, and I. Ahmedy, "Energy harvesting and battery power based routing in wireless sensor networks," *Wireless Networks*, 2015.
- [3] H. Modares, R. Salleh, and A. Moravejsharieh, "Overview of security issues in wireless sensor networks," in *Proceedings of the 3rd International Conference on Computational Intelligence, Modelling and Simulation (CIMSIM '11)*, pp. 308–311, September 2011.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [5] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.
- [6] M. Y. I. Idris, A. M. A. Abu Znaid, A. W. A. Wahab, L. K. Qabajeh, and O. A. Mahdi, "Low communication cost (LCC) scheme for localizing mobile wireless sensor networks," *Wireless Networks*, 2016.
- [7] M. Ahmed, M. R. Sattari, M. K. Nasir et al., "Vehicle adhoc sensor network framework to provide green communication for urban operation rescue," *Lecture Notes on Information Theory*, vol. 1, no. 2, pp. 77–82, 2013.
- [8] S. Khan, A. Gani, A. W. Wahab, M. Shiraz, and I. Ahmad, "Network forensics: review, taxonomy, and open challenges," *Journal of Network and Computer Applications*, vol. 66, pp. 214–235, 2016.
- [9] S. Park, "Performance evaluation of data aggregation schemes in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, vol. 1, pp. 413–418, Las Vegas, Nev, USA, April 2006.
- [10] M. H. Anisi, A. H. Abdullah, and S. A. Razak, "Energy-efficient and reliable data delivery in wireless sensor networks," *Wireless Networks*, vol. 19, no. 4, pp. 495–505, 2013.
- [11] S. A. A. Shah, M. Shiraz, M. K. Nasir, and R. B. M. Noor, "Unicast routing protocols for urban vehicular networks: review, taxonomy, and open research issues," *Journal of Zhejiang University: Science C*, vol. 15, no. 7, pp. 489–513, 2014.

- [12] G. Abdul-Salaam, A. H. Abdullah, M. H. Anisi, A. Gani, and A. Alelaiwi, "A comparative analysis of energy conservation approaches in hybrid wireless sensor networks data collection protocols," *Telecommunication Systems*, vol. 61, no. 1, pp. 159–179, 2016.
- [13] S. Halawani and A. W. Khan, "Sensors lifetime enhancement techniques in wireless sensor networks—a survey," *Journal of Computing*, vol. 2, no. 5, pp. 34–47, 2010.
- [14] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [15] M. Gasser, "In-Network Aggregation Techniques in Wireless Sensor Networks," *Internet Economics VIII*, p. 7, 2014.
- [16] B. Abid, T. T. Nguyen, and H. Seba, "New data aggregation approach for time-constrained wireless sensor networks," *The Journal of Supercomputing*, vol. 71, no. 5, pp. 1678–1693, 2015.
- [17] C. Chen, K. Lee, J.-S. Park, and S. J. Baek, "Minimum cost data aggregation for wireless sensor networks computing functions of sensed data," *Journal of Sensors*, vol. 2015, Article ID 506909, 17 pages, 2015.
- [18] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregation scheduling algorithms in wireless sensor networks: solutions and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1339–1368, 2014.
- [19] L. Chitnis, A. Dobra, and S. Ranka, "Aggregation methods for large-scale sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 2, article 9, 2008.
- [20] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.
- [21] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, article 20, 2009.
- [22] J. Li, A. Deshpande, and S. Khuller, "On computing compression trees for data collection in wireless sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, Calif, USA, March 2010.
- [23] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems Workshops*, pp. 575–578, Vienna, Austria, 2002.
- [24] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 41–54, 2006.
- [25] S.-J. Park and R. Sivakumar, "Energy efficient correlated data aggregation for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, pp. 13–27, 2008.
- [26] M. Lee and V. W. Wong, "An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks," in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '05)*, IEEE, August 2005.
- [27] Z. Eskandari, M. H. Yaghmaee, and A. Mohajerzadeh, "Energy efficient spanning tree for data aggregation in wireless sensor networks," in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN '08)*, pp. 1–5, IEEE, St. Thomas, Virgin Islands, USA, August 2008.
- [28] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.
- [29] R. Ghosh and S. Basagni, "Napping backbones: energy efficient topology control for wireless sensor networks," in *Proceedings of the IEEE Radio and Wireless Symposium*, pp. 611–614, January 2006.
- [30] Y. Chen and S. H. Son, "A fault tolerant topology control in wireless sensor networks," in *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications*, pp. 269–276, January 2005.
- [31] M. Busse, T. Haenselmann, and W. Effelsberg, "TECA: a topology and energy control algorithm for wireless sensor networks," in *Proceedings of the 9th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '06)*, pp. 317–321, October 2006.
- [32] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [33] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.
- [34] W. B. Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks*, Massachusetts Institute of Technology, 2000.
- [35] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [36] S. Park, *Performance analysis of data aggregation schemes for wireless sensor networks static-cluster aggregation and dynamic-cluster aggregation section [Ph.D. dissertation]*, North Carolina State University, Raleigh, NC, USA, 2006.
- [37] J.-Y. Cheng, S.-J. Ruan, R.-G. Cheng, and T.-T. Hsu, "PADCP: power-aware dynamic clustering protocol for wireless sensor network," in *Proceedings of the IFIP International Conference on Wireless and Optical Communications Networks*, 6 pages, IEEE, Bangalore, India, April 2006.
- [38] E. Nakamura, H. de Oliveira, L. Pontello, and A. Loureiro, "On demand role assignment for event-detection in sensor networks," in *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pp. 941–947, Cagliari, Italy, 2006.
- [39] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. de Oliveira, R. B. de Araujo, and A. A. Loureiro, "DRINA: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 676–689, 2013.
- [40] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, IEEE, Maui, Hawaii, USA, January 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

