IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2020, accepted March 14, 2020, date of publication March 24, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983112

A Novel Distributed CDS Algorithm for Extending Lifetime of WSNs With Solar Energy Harvester Nodes for Smart Agriculture Applications

OMER GULEC[1], ELIF HAYTAOGLU[2], AND SEZAI TOKAT[2]
[1]Department of Electrical and Electronics Engineering, Pamukkale University, 20160 Denizli, Turkey
[2]Department of Computer Engineering, Pamukkale University, 20160 Denizli, Turkey

Corresponding author: Omer Gulec (omer@pau.edu.tr)

This work was supported in part by the Pamukkale University Scientific Research Projects Coordinatorship under Grant 2019FEBE015.

ABSTRACT Recent improvements in computer and software technologies affect different areas of production systems. The productivity of an agricultural process can be accepted as one of the many domains affected by these improvements. The production volume is increased not only with new designs of agricultural machines, but also by utilizing communication technologies in the production process. Even the wireless sensor technologies have started to be used for gathering some environmental features in order to optimize production parameters as well as preserving the product in determined quality levels. However, nodes which play an important part of such systems are prone to battery depletions. To alleviate this battery problem, incorporating harvester nodes into the system has been recently considered. However, only including such nodes is not sufficient for improving the lifetime of the system. In this paper, a new distributed connected dominating set algorithm on WSNs with solar energy harvester nodes for precision agriculture applications is proposed. The novel distributed connected dominating set construction with solar energy harvesting in smart agriculture applications algorithm, namely CDSSEHA, is compared with the traditional flooding methods and with an energy efficient CDS algorithm. According to the results, the proposed algorithm increased the WSNs' lifetime by up to approximately 6 times and 1.4 times compared to the traditional flooding methods and CDS based method, respectively. Furthermore, the CDS construction process constitutes only about 15% of the whole lifetime.

INDEX TERMS Distributed connected dominating sets, precision agriculture, smart agriculture applications, solar energy harvester, wireless sensor networks.

I. INTRODUCTION

With the increasing demand for food, smart agriculture and farming applications have gained importance and wide usage as the traditional methods have lost their sufficiency. Especially, the recent improvements in sensor technologies have enabled the IoT applications to be easily deployed and used for improving the quantity and quality of the agriculture production while reducing the costs. Typically, sensors are located on several locations of a farm to sense the environmental features and then the sensed information is forwarded to the base station through a wireless sensor network (WSN).

A WSN consists of many sensor nodes working cooperatively and located on a large scale of an area in a distributed manner. The collection of low powered tiny nodes equipped with sensors which gather crucial environmental variables from the nature, can transfer the sensed data to other nodes or a center.

The WSN systems have been widely utilized in medical, military, industrial, domestic, agricultural as well as environmental monitoring and control areas. Agriculture is one of the most suitable areas for deployment of WSN, since data gathering from the physical environment is a very important requirement in this area.

Precision agriculture (PA) is a modern farming management concept in which computer technologies are employed to monitor, control and optimise agricultural production processes. In PA applications, the data to be sensed is critical and its flow must be carried out continuously. To improve the quality of the production, some environmental values should be sensed and controlled frequently. These operations can be easily supported by WSNs.

There are different types of physical, chemical and biological features measured in agriculture applications such

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

as soil moisture, air humidity, soil and air temperature, light intensity, illumination, water quality, wind direction, wind speed, air flow, CO_2 and O_2 levels, gas density, radioactive contamination, pressure, pH values of water and soil, rain volume, etc. These features are utilized in smart city and smart farm solutions and in applications such as plant tracking, animal tracking and/or feeding, pollution detection, forest or farm fire detection and alert, habitat monitoring, disaster (earthquakes, hurricanes and floods) monitoring and also in many other areas mentioned in the literature [1], [2].

There are many duty cycle methods commonly used in WSN applications to balance the energy usage of nodes constituting these networks [3]–[6]. It is known that duty cycle methods can reduce energy consumption of the nodes since, in this scheme, nodes stop radio transmission and stall sensing the environmental variables in sleep mode to save energy. However, duty cycle approach may lead data losses and transfer delay, especially for multi-hop data transmission as well as complicates the discovery of neighbour process [5]. Moreover, when there exists any data to be transmitted, the delay would be increased while the receiver nodes are in sleep mode, this situation exacerbates with the increase in hop count between the source and the sink node. In this case, the sender node should wait the receiver node to become active. Thus, the data is only transferred when the nodes, on the way to sink, are active at the same time [7]. Additionally, the duty cycle should be kept larger for effective data tracking so this will cause huge energy consumption all over the network [6]. As a result, duty cycle is not the best and the unique way to reduce energy consumption where the packet delays matter.

Due to the distributed structure of WSNs, graph theory-based solutions are used to construct a communication backbone, such as independent sets (IS), dominating sets (DS) and connected dominating sets (CDS) [8]. If there are no communication backbones, the flooding method is selected as the main communication method among the nodes in WSNs. In fact, flooding-based communication is simple and fast to implement. Moreover, this kind of setup does not need any routing mechanisms in advance [9]. Despite the fact that the network flooding method is easy to maintain, CDS is more efficient in message forwarding. Thus, CDS construction is a crucial way to build backbones in ad-hoc wireless networks [10]. In a CDS, there are two types of nodes called dominators and dominatees. If a node is selected for DS or CDS, this node is called dominator which is shortly called DTOR. If a node is not a member of DS or CDS, this node is called dominatee which is abbreviated as DTEE [7].

In a WSN, finding the communication backbone with the minimum weight becomes finding the minimum weighted connected dominating set (WCDS) problem, since the node weights are related to their remaining energies [11].

After the CDS is constructed in a WSN, the communication messages will be routed through the CDS from source nodes to a destination node, namely, the sink node. Nodes first send the package to their parent nodes. Then each receiver parent forwards the package to its respective parent node until the package is delivered by the root node. Using this communication method makes the network lifetime quite longer and the path searching time shorter [12]. For an efficient routing, the CDS size should be minimized. Therefore, the minimal CDS (MCDS) problem has been investigated in many studies.

WSNs suffer from many problems which have been studied in the literature. As the nodes may be located on rough large-scale fields and are powered by low capacity batteries, WSN systems are prone to energy depletions [13]. Thus, energy consumption of the nodes is the main problem to be considered to prolong the lifetime of the network.

A sensor device supplies its energy from the batteries. If a node runs out of energy, it will disconnect from the network and this may cause connection problems in the WSN. In the worst case, the network may be separated into several sub-networks according to the position of the dead nodes. Additionally, network lifetime is a key characteristic for a WSN. The network lifetime is measured as the time duration until the first sensor in the network exhausts its battery and dies [14], [15]. As a result, the energy level is the most important problem to be considered in WSNs.

In order to alleviate the energy consumption problem, energy harvester nodes have been considered to be included into the WSNs [16]. These kind of networks are called Energy Harvest Networks (EHN). There exist many energy harvesting techniques developed in WSNs using different kinds of resources such as solar energy, thermal energy, wind and hydro flow energies.

Solar power density efficiencies can reach $15mW/cm^2$ levels in sunny days, while the other harvesting techniques stay in μ or η levels [17]. Consequently, energy harvesting from sunlight is stated as the most effective method in WSNs [18]. Technically, the solar energy harvesting method is performed by using photo-voltaic cells, the surface of which is hit by the sunlight, energizing free electrons and initiates a current flow. This operation is called as the conversion of solar energy directly into electricity [19], where no other devices are necessary to power up the sensor nodes. The other way is charging the batteries with the harvested energy and then powering the sensors with this stored energy. This is the indirect way of energy harvesting in EHNs.

In this paper, a novel distributed connected dominating set construction using solar energy harvester nodes in smart agriculture applications algorithm, namely CDSSEHA, which constructs CDS with solar harvester sensor nodes for agricultural areas is proposed. The communication backbone of WSN nodes deployed in all types of agriculture applications can be constituted using CDS constructed by CDSSEHA algorithm. In the proposed algorithm, the difference between the harvester nodes and the ordinary nodes, which do not show the capability of energy harvesting property, is taken into account.

The performance of the CDSSEHA is compared with the energy efficient CDS algorithm of Al-Nabhan *et. al.* [20]'s which we shortly called APRCDS to represent their CDS approach and with two traditional flooding approaches which are based on [21].

To compare our proposed algorithm with an existing approach in the literature, we have chosen APRCDS algorithm which is also used to construct energy efficient distributed CDS for WSNs. However, in that study the use of energy harvester nodes was not considered. They also compared their CDS backbone construction model with the network without CDS. Therefore, we implemented APRCDS and compared it with our proposed algorithm under the same simulation parameters given in Table 2 on networks having both harvester and ordinary nodes.

As for the flooding approaches, the first one which is called FLD uses flooding approach as in [21] on the network having only ordinary nodes. The second approach which is called FLDH uses the same flooding method including both ordinary and harvester nodes. Likewise, APRCDS is also conducted on networks having both harvester and ordinary nodes. According to the simulation results, the lifetime of WSNs in CDSSEHA is increased by approximately 6 times and 1.4 times in average compared to these traditional methods and CDS based method, respectively.

The rest of the paper is organized as follows. Related work on EHNs and CDS algorithms is summarized in Section II. Problem definitions and the proposed algorithm of the current study are given in Section III. Sample scenarios and simulation results are given in Section IV. Finally, Section V concludes the study.

II. RELATED WORK

In the literature, there exist many studies related to WSNs in different areas. It is clear that one of the major problems related to WSNs is the energy problem which causes the failure of the system. When a node runs out of energy, this leads to changes in the network topology and also affects the network's lifetime. Therefore, studies that focus on saving energy have been proposed recently [13]. In many WSN applications, it is supposed that the lifetime of a network is completed after only one node shuts down in the network [22].

In real life, WSNs are used in the agriculture area for many different purposes as mentioned before. For instance, Daskalakis *et al.* [23] presented a novel low-cost and low-power WSN application for leaf sensing using backscatter node/tag which detects water needs of the plants to prevent the water waste of the irrigation systems. Kamath *et al.* [24] designed wireless visual sensor network to monitor paddy crops using Raspberry Pi as visual sensor node and Bluetooth as a communication method. Khan *et al.* [25] presented an energy efficient water utilization and decision support system with the help of WSNs for irrigation of crops. Meda *et al.* [26] proposed a plant monitoring system which had mobile sensors moving randomly in a farm to gather data from different locations. The gathered data was analysed by the base station where the data was processed to protect the plants against any bacterial and fungal diseases via image processing techniques.

There also exist different types of WSN applications in the area of agriculture. One of them was proposed by Villarrubia *et al.* [27] in which, fuzzy logic applications were used on the WSN system for monitoring crops and controlling their irrigation. Another one was proposed by Nikolidakis *et al.* [28] where a new energy-efficient routing protocol on the WSN system using Zigbee protocol was proposed for automated control of the irrigation systems in agriculture and for monitoring the weather conditions nearby the agricultural fields. Shinghal and Srivastava [29] proposed a WSN application to improve potato crops production which monitors and analyses the requirements such as irrigation, fertilization and etc.

In addition, deploying WSNs, Hedley *et al.* [30] developed an energy-saving method of sending soil moisture data to the base station through an energy-efficient route. Khan and Kumar [31] designed an algorithm for finding low cost routing path between a sensor node and a mobile sink. They also proposed an algorithm for mobile sink travelling path for reducing energy consumption and delays of WSNs deployed for ambient crop field monitoring.

Beside plant or soil monitoring in agricultural applications, WSN systems are also efficient for animal tracking and behaviour detection systems. For instance, with the help of WSNs, Radoi *et al.* [32] proposed a tracking and monitoring system for Retuerta horses where body-worn mobile sensors were attached to the horses in order to collect and then to transfer data to the base station. Luque *et al.* [33] proposed an environmental Wireless Acoustic Sensor Network architecture based on MPEG-7 standart which is tested on anuran spices that live in Spanish natural parks. Loreti *et al.* [34] designed a wireless sensor node and energy harvesting architecture of WSN for tracking and monitoring the pink iguanas of the Galápagos which have recently been discovered in Galápagos Islands. Nath and Azharuddin [35] proposed a new model for WSN application and developed routing algorithm for the model to protect rhinos against poachers.

As previously mentioned, the flooding is mainly selected for the communication between the nodes in WSN systems. There are numerous flooding-based studies in the literature. Cheng *et al.* [36] proposed a dynamics reliable flooding method in low duty cycle WSNs. The study is based on dynamically switching the decisions when a transporting failure exists in a network. Thus, the constructed flooding tree is transformed depending on packet reception results. Cheng *et al.* [37] also presented construction of an energy-efficient flooding tree for minimum delay on duty cycle in unreliable wireless links in WSN systems.

Duty cycle methods are also used commonly in aforementioned WSN applications. However, these methods are not very energy efficient in large networks and cause time delays. To alleviate this problem, Liu *et al.* [7] proposed a CDS algorithm in addition with an appropriate duty cycle control method which reduces the energy consumption and data transmission delay. Similarly, W. Shi *et al.* [6] proposed a novel model by adding duty cycle in CDS for energy efficiency and fast data transmitting. However, in these studies, energy harvester nodes have not been considered.

Like flooding, CDS is more recently considered as a communication method in WSN systems. In the literature, many methods have been proposed to construct CDS to prolong the lifetime of WSNs without any harvester nodes. For instance, Luo *et al.* [38] proposed a novel DS-based distributed algorithm to find minimal CDS (MCDS) on the network. Mohanty *et al.* [39] proposed a novel MCDS using two-hop distances which reduces the CDS size. Qi and Yang [40] developed a MCDS-based control system to provide high efficient connectivity of direct communications in the flying ad-hoc networks. Pino *et al.* [41] introduces multiple local search algorithms which maximize the lifetime of the WSN using dominating sets.

Many energy harvesting technologies are also used in agriculture applications. Kwon *et al.* [42] constructed solar energy harvest networks (EHN) with a new prediction technique. Hou and Gao [43] constructed a WSN system with solar harvester nodes. In their work, humidity and temperature sensors are used and power consumption is minimized. Gutiérrez *et al.* [44] proposed an irrigation method using WSN nodes powered with solar panels to optimise water use in crops. The network consists of temperature and soil moisture sensors placed in plant roots and a gateway. López-Lapeña and Pallas-Areny [45] developed a novel solar energy radiation measurement method with decreasing power consumption in WSNs having solar energy harvester nodes in agriculture. Zou *et al.* [46] extended the lifetime of WSN with energy harvesting nodes using shadow detection. In their study, nodes can optimise their power production and consumption according to the power source. Sharma *et al.* [47] proposed a solution to extend the lifetime of WSNs with solar energy harvesting deployed for smart agriculture monitoring. In their study, they used duty cycling which may not be the best way for communication aforementioned, to decrease energy consumption. Hence, the network lifetime is increased at 25% duty cycle.

Despite the fact that the energy harvesting methods or CDS constructing algorithms were used separately in WSN systems, to the best of our knowledge, in the literature there exists no CDS construction algorithm, in which both solar energy harvester nodes and agricultural fields are taken into account. This is an important gap to be considered for smart agricultural applications. Extending the lifetime of WSNs with solar energy harvesting is not sufficient on its own without any communication backbone. Thus, in this study, CDS construction algorithms are proposed as the communication backbone. Likewise, constructing CDS backbone is not adequate for energy saving of the network. To this end, solar energy harvester nodes are placed in the network to support CDS.

To extend the lifetime of the network, choosing all the nodes as harvester nodes may not be possible due to the costs and maintenance processes. Hence, this study shows that using ordinary and solar energy harvester nodes that collaboratively operate in CDS construction process for agricultural areas prolongs the lifetime of the WSN systems.

III. THE PROPOSED ALGORITHM

In the following subsections, a new distributed CDS algorithm developed for agriculture applications based on WSNs including solar energy harvester nodes and ordinary nodes, CDSSEHA, is proposed. The sub-algorithms of CDSSEHA for energy harvester and ordinary nodes are given in Section III-A. In Section III-B, the connection method for the selected DTORs, which involves finding partial trees and choosing some DTEE nodes as DTORs to connect these trees as in [11], is elaborated.

In typical agriculture applications, nodes sense the required environmental features and forward the gathered information to a base station. Thus, after the communication backbone has been constructed using CDSSEHA, to obtain the performance results of the proposed algorithm, sample application given is conducted on the network.

A. DTOR SELECTION ALGORITHM

The WSN system that is composed of sensor nodes and the communication links between them, can be seen as an undirected connected graph $G(V, E)$ where V represents the sensor node set and E represents the edges between the pairs of sensors that have a direct communication. The sensor nodes in the proposed WSN system are divided into two types: energy harvester and ordinary ones. Algorithms employed on these nodes are changed depending on their types.

Before starting the algorithm, every node in WSN boots in IDLE state and knows its one hop neighbour. Once the algorithm is started, every node sends the remaining energy of its batteries to its neighbours.

Each harvester node becomes DTOR directly to be a part of the connected dominating set (CDS) depending on its energy harvesting capacity. In order to construct the CDS, the harvester nodes run Algorithm 1. Once a harvester node becomes DTOR, it broadcasts HARV_STATE announcement to its neighbours. After this stage, the harvester nodes wait for the decisions of the other nodes about becoming DTOR or DTEE.

Whenever an IDLE neighbour node receives HARV_STATE announcement, it becomes DTEE as stated in Algorithm 2, sets the harvester neighbour to inactive state and, thus, it will not join the next rounds. When a node collects HARV_STATE messages from all of its harvester neighbours, it sets the list of its active neighbours, Γ_{act}, which does not include any harvester neighbour. Active neighbours list is a list of nodes, the owner of which uses it for their potential rivals on being DTOR for the next round. If a neighbour is not IDLE and has no any IDLE neighbours, either, it means that it has nothing else to do for the CDS construction algorithm and will be deleted from Γ_{act} like the harvester nodes. The list is updated on each round until there is no active neighbours left.

Algorithm 1 Dominating Set Algorithm for Energy Harvester Nodes

data:
$\Gamma_m \leftarrow$ neighbour list of node m
$e_m \leftarrow$ remaining energy of node m
$state_m \leftarrow$ *State of node m (IDLE, DTOR or DTEE)*
$harvesterFlag_m \leftarrow$ *true if the node is harvester*
$finished \leftarrow$ *true if the algorithm terminates*
$doneCount \leftarrow$ *has the value of the number of nodes that are ready to start the next round*
initially:
$state_m \leftarrow IDLE$
$harvesterFlag_m \leftarrow true$
$finished \leftarrow false$
$doneCount \leftarrow 0$
upon the algorithm started:
$state_m \leftarrow DTOR$
broadcast **HARV_STATE** msg to Γ_m
broadcast **DONE** msg to Γ_m
upon receiving $DONE_n$ msg from $node_n$
$doneCount \leftarrow doneCount + 1$
if $doneCount = |\Gamma_m|$ **then**
$finished \leftarrow checkPhaseEnd()$
end if

Upon receiving the HARV_STATE message, an ordinary node sends its STATE message to all of its active neighbours. If a node receives a STATE message from a neighbour n, it updates the state of node n.

Whenever a node completes receiving the STATE messages from all of its neighbours for the first round, it sends DONE messages to its neighbours. DONE variable is set to true when the node's state is not IDLE and none of its neighbours are IDLE, either. Otherwise, if the node does not finish yet, DONE variable is set to false and it is ready for the next rounds. A node notices the termination of a round after receiving DONE messages from all of its active neighbours. The round number is also incremented.

If a node does not finish yet and receives DONE messages from its active neighbours, it updates its Γ_{act} list and calculates its new WEIGHT for the next round. The new WEIGHT of the ordinary nodes is calculated by summing the remaining energy and the number of harvester nodes. Since the level of the remaining energy is much higher than the value of the number of harvester nodes, the values are scaled first and then the summation operation is performed.

After calculating the new WEIGHT values, all active nodes broadcast the value to their active neighbours. If an active node receives the new WEIGHT announcement messages from all of its active neighbours, it checks if its WEIGHT is the maximum among its active neighbours and then it sets its AMIMAX variable with the result. If its weight is the maximum, it becomes DTOR. Whether it becomes DTOR or not, it broadcasts its AMIMAX value to its active neighbours.

Algorithm 2 Dominating Set Algorithm for Ordinary Nodes

data:
$\Gamma_m \leftarrow$ neighbour list of node m
$\Gamma_{act} \leftarrow$ active neighbour list of node m
$\Gamma_{harv} \leftarrow$ harvester neighbour list of node m
$e_m \leftarrow$ remaining energy of node m
$\omega_m \leftarrow$ weight of node m
$state_m \leftarrow$ *State of node m (IDLE, DTOR or DTEE)*
$doneCount \leftarrow$ *has the value of the number of nodes that are ready to start the second round*
$n{-}>newRound \leftarrow false, \quad \forall n \mid n \in \Gamma_m \setminus \Gamma_{harv}$
$harvesterFlag_m \leftarrow false$
$finished \leftarrow$ *true if the algorithm terminates*
initially:
$state_m \leftarrow IDLE$
$harvesterFlag_m \leftarrow false$
$\omega_m \leftarrow battery\_level, \ \Gamma_{act} \leftarrow \Gamma_m$
$doneCount \leftarrow 0, AMIMAX \leftarrow false$
$newStateCount \leftarrow 0, amIMaxCount \leftarrow 0$
$finished \leftarrow false$
$DONE \leftarrow false$
upon receiving STATE msgs from node n**,** $\forall n\ an \in \Gamma_{act}$ **and** $harvesterFlag_m =$ **false**
update state of the nodes
A:
if $\neg$ checkanyIDLE(Γ_m) $\wedge$ $state_m \neq$ IDLE **then**
DONE $\leftarrow$ true
end if
broadcast DONE to Γ_m
upon receiving HARV_STATE msgs from node n**,**$\forall n \in \Gamma_{harv}$ **:**
$state_m \leftarrow$ DTEE, $\Gamma_{act} \leftarrow \Gamma_{act} \setminus n$, $\forall n \in \Gamma_{harv}$
broadcast STATE msgs to Γ_{act}
run **A**
upon receiving WEIGHT$_n$ **msgs from node** n
$weights \leftarrow weights \cup (n, w_n)$
if $|weights| = \Gamma_{act}$ **then**
find node z having max weight
$weights \leftarrow \emptyset$
if $z = m$ **then** $AMIMAX \leftarrow true$
end if
send AMIMAX to Γ_{act}
end if

If a node receives an AMIMAX message possessing a true value from any of its neighbours, this means that the sender node has become DTOR and the receiver node becomes DTEE in case of being IDLE beforehand. When a node receives all AMIMAX messages from its active neighbours, it broadcasts NEW_STATE messages to its active neighbours. NEW_STATE message contains the state of a node for the next round (IDLE, DTEE or DTOR).

If a node receives all NEW_STATE messages from its active neighbours, it checks if its state is IDLE or has any

```
upon receiving AMIMAX_n msgs from node n
  if AMIMAX_n = true then
      n−>state ← DTOR
      n−>newRound ← false
      amIMaxCount ← amIMaxCount+1
    if STATE = IDLE then
        STATE ← DTEE
    end if
    if amIMaxCount = |Γ_act| then
        send NEW_STATE to Γ_act
        amIMaxCount ← 0
    end if
  end if
upon receiving NEW_STATE_n msg from node n
  n−>state ← NEW_STATE_n
  newStateCount ← newStateCount+1
  if newStateCount = |Γ_act| then
      DONE ← checkPhaseEnd( )
      broadcast DONE_m msgs to Γ_act
      newStateCount ← 0
  end if
upon receiving DONE_n msgs from node n
  n−>newRound ← DONE_n
  if (round = 0 ∧ doneCount = |Γ_m|) ∨ (round ≠ 0 ∧
  doneCount = |Γ_act|) then
      Γ_act = {n | n−>newRound = false, ∀n ∈
  Γ_act},
      checkPhaseEnd()
      doneCount ← 0
      round ← round + 1
    if DONE = false then
        AMIMAX ← false
        send WEIGHT msg to Γ_act
    end if
  end if
  doneCount ← doneCount + 1
checkAnyIDLE():
  if state_n = IDLE , ∃n ∈ {Γ_m ∪ m} then
    finished ← true,
  else
    finished ← false
  end if
  return finished
checkPhaseEnd():
  if state_n ≠ IDLE, ∀n ∈ {Γ_m ∪ m} then
    finished ← true,
  else
    finished ← false
  end if
  return finished
```

neighbours in IDLE state using the **checkAnyIDLE** function. In the case where none of them are true, it broadcasts DONE (having a true value) message to its active neighbours, that informs it is finished and will not join the next rounds. Otherwise, it sends DONE message with a false value to its active neighbours that informs its readiness for the next rounds.

The first phase terminates when there exist no IDLE nodes in the WSN. The resulting structure of the network constitutes the dominating set in which each node is dominated by at least one DTOR neighbour node, or the node itself is a DTOR. In the second phase, in order to connect the DTORs, the dominating set will be connected by selecting some DTEE nodes as DTORs.

To further elaborate the proposed algorithm, CDSSEHA, two finite state machines (FSM) corresponding to the harvester nodes and ordinary nodes are given in Appendix. The states of the nodes are described in Table 1.

TABLE 1. FSM states.

Node Type	States	Explanation
Harvester Nodes	IDLE DTOR	The role in CDS has not been defined yet. It is used as a backbone node in CDS.
Ordinary Nodes	IDLE P_DTOR DTEE DTOR	The role in CDS has not been defined yet. The WEIGHT message has just been sent. It is covered by any DTOR node It is used as a backbone node in CDS.

The organization of FSM for harvester nodes is straightforward which is given in Figure 6 in Appendix. The harvester nodes can be either in IDLE state or DTOR state. When the algorithm is started, harvester nodes are in IDLE state. They commence the algorithm by sending HARV_STATE messages as well as DONE messages to their neighbours. After they send these messages, they become DTOR. HARV_STATE messages indicate that the harvester node is alive and in DTOR state. DONE messages stand for synchronization with the ordinary nodes. When all neighbours of a harvester node finished the first phase of the algorithm, the harvester node is ready to start the second phase in which the connection process of DTOR nodes is conducted.

The FSM diagram of the ordinary nodes is given in Figure 7 in Appendix. The ordinary nodes can be in one of the four states during CDSSEHA algorithm. These states are identified by IDLE, P_DTOR, DTEE, DTOR. Similar to the harvester nodes, the ordinary nodes start the algorithm in IDLE state. If there are any harvester neighbours within one hop proximity, these nodes wait for those harvester nodes' HARV_STATE messages. After all ordinary nodes synchronize with their harvester neighbours through HARV_STATE messages, they broadcast their current state messages by sending STATE messages. If neither the state of a node and nor the state of its neighbours are IDLE in this point, this node is ready to start the second phase, hence, it does not have to be active in the subsequent rounds of the phase 1. Thus, each ordinary node announces if it continues to the first phase by broadcasting a DONE message having the false value. Upon receiving all DONE messages from the active ordinary neighbours, an active ordinary node broadcasts its remaining energy in WEIGHT messages to their ordinary nodes to become a DTOR, namely P_DTOR (possible DTOR). Upon receiving all WEIGHT messages, the node having the max WEIGHT among nodes in its one hop neighbourhood becomes DTOR and broadcasts AMIMAX message with the true value. When a neighbour of an IDLE node becomes DTOR, it becomes DTEE. After the exchange of WEIGHT messages has been completed, nodes inform other nodes if it has the maximum WEIGHT value using AMIMAX messages. Then, the ordinary nodes start to broadcast their final state with NEW_STATE messages. If a DTEE node receives all NEW_STATE messages from its all active neighbours, it first checks whether it continues to compete with their neighbours in next rounds to become a DTOR or not. If an IDLE node receives all NEW_STATE messages from its all active neighbours, independent from the state of their neighbors it will continue the next round. However, after an ordinary node whose state is either DTOR or DTEE and none of their neighbours are in IDLE state, it will start to execute the DTOR connection algorithm. Otherwise, i.e. any neighbour of an ordinary node is in IDLE state, this node continues to the execute next rounds of the first phase.

B. DTOR CONNECTION ALGORITHM

In order to connect DTOR nodes for CDS construction, firstly GHS [48] algorithm is employed to find partial trees (already connected DTORs). Before starting the GHS algorithm, every node knows its neighbours (Γ_m) and their states. Initially, DTOR nodes constitute partial trees. DTOR nodes set their tree ID values with the ID value of the root node of the tree which they belong to.

In this phase, DTOR nodes send the tree ID values to their DTEE neighbours in each round. After receiving these values from all DTOR neighbours, a DTEE node sends the Γ^{tree_m} set including these tree IDs and their remaining energy levels to their DTEE neighbours. Upon collecting all these sets from DTEE neighbours, DTEE nodes calculate their *cost* and broadcast it to the neighbour trees. A DTEE node calculates its *cost* value as given in Equation 1.

$$min(\left\{\frac{1}{e_m|\Gamma^{tree}|}, \bigcup_{n\in\Gamma_m} \frac{(\frac{1}{e_m}+\frac{1}{e_n})}{|\Gamma^{tree_m}\cup\Gamma^{tree_n}|}\right\}). \tag{1}$$

A DTEE $node_m$ set its cost value to $\frac{1}{e_m|\Gamma^{tree}|}$ only if its cost value ($\frac{1}{e_m|\Gamma^{tree}|}$) is less than $\frac{(\frac{1}{e_m}+\frac{1}{e_n})}{|\Gamma^{tree_m}\cup\Gamma^{tree_n}|}$ for $\forall$ $node_n \in \Gamma_m$. In that case selecting $node_m$ only itself as DTOR is plausible in terms of energy usage, otherwise selecting $node_m$ with another DTEE $node_n$ makes sense.

After receiving all *cost* values from its DTEE neighbours, the DTORs send the ID of a DTEE neighbour having the minimum *cost* to its parents in the partial tree. Eventually, the root of a partial tree decides the node(s) with the minimum *cost* as connector(s) and the tree informs its DTEE neighbours about the ID(s) of DTEE nodes having the minimum *cost*(s). The ties are broken using ID values of the nodes. A DTEE node that receives all ID values from its neighbour trees checks if each of the received values is equal to its ID. If so, it sends a MERGE message to its neighbour trees to connect all of these neighbour trees and this DTEE node becomes the root of the newly constructed tree. This operation is repeated until there is only one tree left in the network. This algorithm constructs a tree where the DTORs are connected. After CDS is constructed, a sample application which uses the constructed CDS will be started.

After connecting the DTOR nodes to each other with the DTOR connection algorithm, a sample application given in Section 4 is started over the network. This application makes nodes forward the sensed messages through the constructed CDS tree. A node in the network, may sense any kind of environmental value according to their setup in the agriculture application and sends the data to its neighbours.

IV. SIMULATION

In order to illustrate and measure the lifetime of WSNs in different topologies, Cooja simulator, which is one of the most popular platforms in IoT applications in recent years, is used in this study. Our proposed algorithm has been tested on many different networks with different number of nodes which are Tmote Sky CC2420 ultra low power IEEE 802.15.4 modules [49].

According to Tmote Sky's datasheet, the total energy consumption of a node is the sum of the consumption of its operations such as transferring, receiving, as well as the processes of idle and sleeping stages [50]. The current flow is calculated as 21.8 mA in the receiving mode (I_{rx}), 19.5 mA in the transferring mode (I_{tx}), 1.8 mA in the CPU usage mode (I_{cpu}) and 0.0545 mA in the sleep mode (I_{lpm}). It is obvious that the node spends much more energy while receiving messages than when sending them.

In order to calculate the level of the produced energy of harvester nodes and thereby to simulate the amount of the energy harvested, the open source SensEH [51] simulator tool which is based on Cooja and its built-in "Energy Harvester" library is used. SensEH computes the energy produced from a solar panel for a harvester node.

For determining the energy consumption, the built-in modules of SensEH's "Energy Consumption Estimation" library is used. It is similar to Cooja's Powertrace tool [52] which tracks energy consumption of the nodes during their different states. Simply, energy consumption estimation library gives the instant energy consumption levels of a node for number of clock ticks per given seconds.

The power consumption levels are measured based on a hardware clock timer, *RTIMER_ARCH_SECONDS*, which has 32768 ticks per seconds. So, the total consumption energy of a Tmote Sky node is calculated by using the

TABLE 2. Simulation parameters.

Network Model	UDGM
Simulator	SensEH
Mote Type	Tmote Sky
Battery Type	$Zn - MnO_2$ Alkaline AA
Initial Energy of a Node	30780 J
Energy Harvesting Source	Solar PV Energy
MAC Model	IEEE 802.15.4
Network Area	100 $\times$100 m^2
Transmission Range	5 to 20 m
Degree of a Node	1 to 4
Number of Topologies on Each Simulation	50
Ratio of Harvester Nodes on Each Simulation	33.3 %
Number of Nodes	30 in FLD-30, FLDH-30, APRCDS-30 and CDSSEHA-30 50 in FLD-50, FLDH-50, APRCDS-50 and CDSSEHA-50
Number of Ordinary Nodes	30 in FLD-30 50 in FLD-50 20 in FLDH-30, APRCDS-30 and CDSSEHA-30 33 in FLDH-50, APRCDS-50 and CDSSEHA-50
Number of Harvester Nodes	0 in FLD-30 and FLD-50 10 in FLDH-30, APRCDS-30 and CDSSEHA-30 17 in FLDH-50, APRCDS-50 and CDSSEHA-50

following formula;

$$\frac{(tx \times I_{tx} + rx \times I_{rx} + cpu \times I_{cpu} + lpm \times I_{lpm}) \times 3V}{RTIMER\_ARCH\_SECONDS} \tag{2}$$

where $I_{tx} = 19.5\ mA$, $I_{rx} = 21.8\ mA$, $I_{cpu} = 1.8\ mA$ and $I_{lpm} = 0.0545\ mA$ according to the datasheet [49].

Every node in the generated network boots with the same energy level due to the characteristics of two AA batteries [53] according to the energy formula given below. For an $Zn-MnO_2$ alkaline AA battery, its initial energy is calculated as;

$$\begin{aligned} E &= P \times t \\ &= V \times I \times t \\ &= (1.5\ V) \times (2.85\ Ah) \times (3600\ s) \\ &= 15390\ J. \end{aligned} \tag{3}$$

Therefore, initially, two AA batteries have the theoretical capacity of 30780 J.

In this paper, all of the aforementioned methods are implemented and analysed in SensEH simulator with Unit Disk Graph Medium (UDGM) model. All of the nodes generated for network topologies are randomly placed in a $100m \times 100m$ area. The transmission range between the nodes is 5 meters to 20 meters. According to this transmission range, the degree of a node which means the maximum number of node's neighbours is set to 4. The initial energy of the nodes is set to 30780 J. The proposed algorithms are tested on two types of topologies in which the first type has 30 nodes and the second type has 50 nodes. For each simulation setup, 50 different network topologies are generated. In topologies possessing energy harvester nodes, 33.3% of the nodes are assigned as harvester nodes. For the best results of solar energy harvesting, it is supposed that all of the harvester nodes get the same sunlight intensity on an agricultural field but their energy consumption differs according to their local processes and communication needs. The parameters used in the simulations are listed in Table 2.

In order to measure the lifetime of the WSNs using CDSSEHA, after connecting the DTOR nodes to each other, a sample application given in Algorithm 3 with a small modification (in line 9) is started over the network. This application makes nodes forward the sensed messages through the constructed CDS tree. A node in the network, may sense any

Algorithm 3 A Sample Application Algorithm for an Agricultural WSN

data:
  $\Gamma_m \leftarrow$ neighbour list of node m
  $e_m \leftarrow$ remaining energy of node m
  finished $\leftarrow$ *true if the algorithm terminates*
initially:
  $e_m \leftarrow$ *battery_level*
  finished $\leftarrow$ *false*
upon the algorithm started:
  broadcast *SENSED_DATA* to Γ_m
upon receiving SENSED_DATA msgs from node n, $\forall n \in \Gamma_m$
  broadcast sensing message to Γ_m
if $e_m \leftarrow 0$ **then**
  finished $\leftarrow$ *true*
else
  broadcast sensing message to Γ_m

kind of environmental values according to their setup in the agriculture application and sends the data to its neighbours. Since the tree of the network is constructed with the CDS construction, every node knows its parent and child(ren) if any exist. The application proceeds as follows, every node periodically senses the environmental features and sends the collected data to its parent instead of Γ_m. In the case this node is a parent node, it waits for all values to be received from its children before sending the data to its parent.

Before a node runs out of energy, first its parent node and then the network will be warned, and all of the communications will be stopped. To that end, a threshold for the remaining battery level is specified. Aforementioned, this means the end of the network lifetime. So, the lifetime of the network would be calculated as the next step.

The proposed CDSSEHA is fistly compared with the traditional flooding methods having two different settings. The first type of the WSN system includes only ordinary nodes and these nodes communicate with each other using flooding, whereas the second type of the WSN system includes both ordinary and harvester nodes. Second, CDSSEHA is compared to the energy efficient distributed CDS construction method, namely APRCDS, having both ordinary and harvester nodes, as well. The details of the traditional systems are given, in subsections IV-A and IV-B respectively.

A. A SAMPLE APPLICATION POSSESSING ONLY ORDINARY NODES USING FLOODING

In order to compare the lifetime of the networks with the same topology, a sample application in Algorithm 3 is also simulated in the network without any CDS backbone, but including only ordinary nodes. In this set of simulations, nodes communicate through flooding. This type of simulations are called FLD. The sample simulation of this method is given in Figure 1.

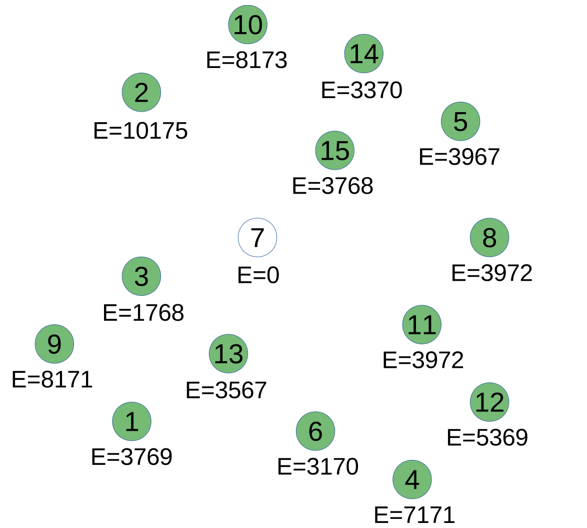


FIGURE 1. Ordinary node 7 runs out of energy during sample application in a WSN which has 15 nodes using FLD method. All green nodes are ordinary nodes (not energy harvester ones).

Before starting the sample application algorithm, it should be noted that all nodes are ordinary nodes and know their neighbours. After the algorithm is started, every node checks whether or not its remaining energy is over the threshold. If it is so, the node starts to get the data from its sensing unit and sends it to the neighbours periodically. Once a node receives this message, it senses the data and sends it to the neighbours. This operation lasts until a node runs out of energy in the network.

The energy threshold is needed for a node to warn the neighbours about its critical energy level. When a node receives this message from a node, it will stop sending any sensed data messages. Then, it warns its neighbours about the neighbour node that is almost out of energy. With this message traffic, every node is warned and they will stop all of the connections in the network and the lifetime of the network without any CDS reaches to the end.

B. A SAMPLE APPLICATION POSSESSING BOTH ORDINARY AND HARVESTER NODES USING FLOODING AND CDS BASED ALGORITHMS

In order to compare the lifetime of the networks with the same topology, a typical sensing application in PA has also been simulated in the network without any CDS backbones but containing both ordinary and harvester nodes. This application has the same steps and conditions with the sample application given in Algorithm 3. In brief, this type of simulations use the flooding method on a network possessing harvester nodes and these simulations are named as FLDH.

Three different communication approaches mentioned before, namely, FLDH, APRCDS and the proposed CDSSEHA are tested on these topologies. Before the simulation results are given, the sample simulations based on these methods are presented in Figure 2, Figure 3 and Figure 4.

As shown in Figure 1, in FLD method, the ordinary node 7 is the first one that runs out of energy in the network,

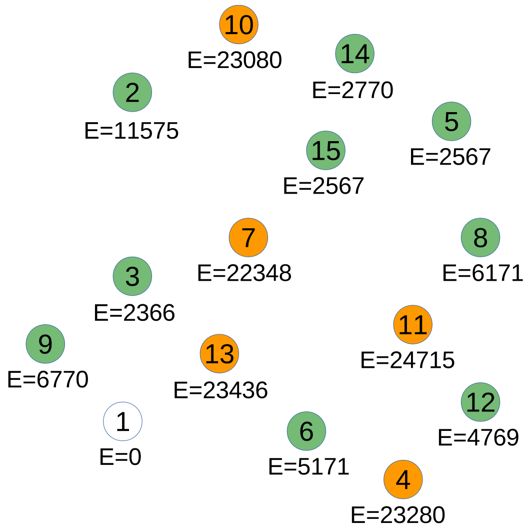


FIGURE 2. Ordinary node 1 runs out of energy in sample application in a WSN using FLDH and having 15 nodes. All green nodes are ordinary nodes and all yellow nodes are energy harvester nodes in a WSN without CDS.

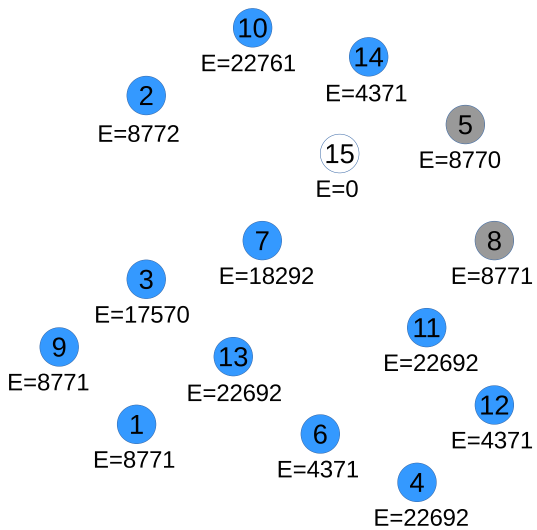


FIGURE 3. Ordinary node 15 is the first one which runs out of energy during sample application in a WSN using Al-Nabhan *et. al.*'s algorithm [20], shortly named as APRCDS and having 15 nodes. Node 4, 7, 10, 11 and 13 are harvester nodes and the rest of them are ordinary nodes.

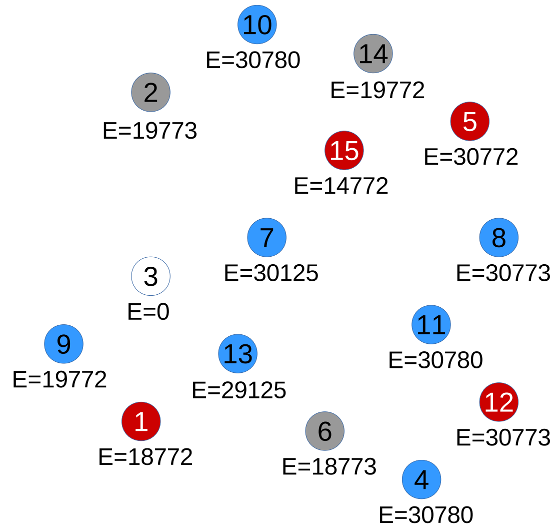


FIGURE 4. Ordinary node 3 is the first one which runs out of energy during sample application in a WSN using CDSSEHA and having 15 nodes. Node 4, 7, 10, 11 and 13 are harvester nodes and the rest of them are ordinary nodes.

containing 15 ordinary nodes, neither presenting a CDS structure, nor including any energy harvester nodes. The disconnection of node 7 leads to the end of the lifetime of the network. In this scenario, node 7 stayed alive for 20 minutes and 52 seconds, once the simulation had started.

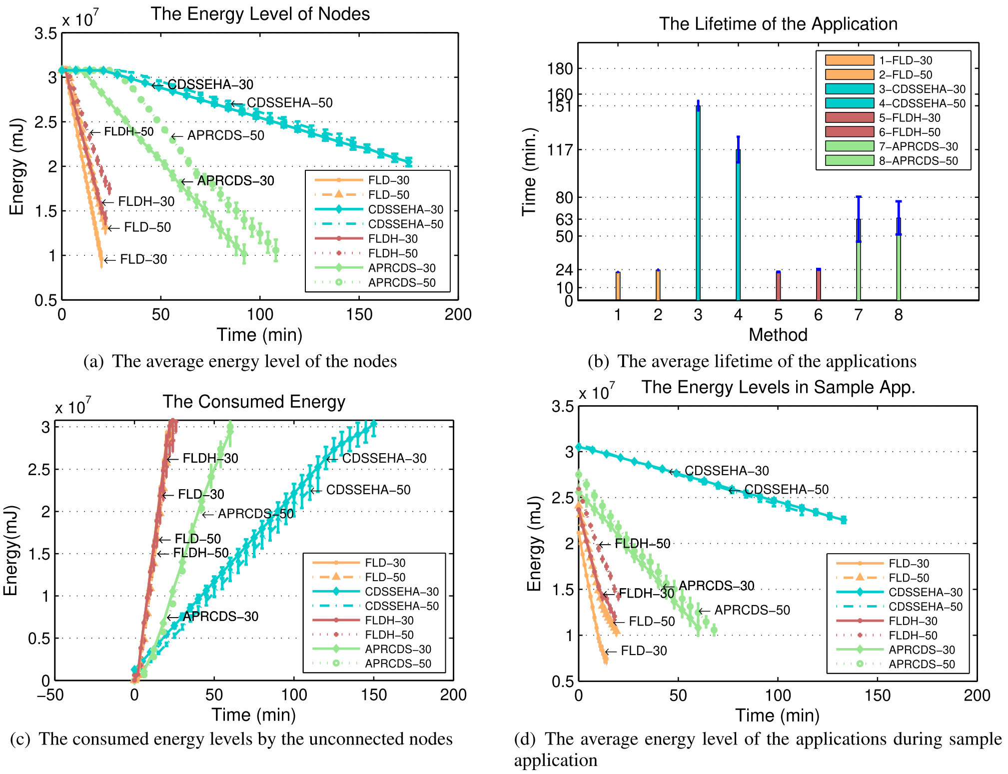
In FLDH method, the ordinary node 1 runs out of energy in the network that has 5 harvester and 10 ordinary nodes as shown in Figure 2. In this scenario, the first unconnected node depleted its energy 21 minutes and 49 seconds after the simulation had started.

In Al-Nabhan *et. al.*'s study [20], shortly named as APRCDS method, the ordinary node 15 runs out of energy in the network with CDS including 5 harvester and 10 ordinary nodes as shown in Figure 3. In this scenario, the first node was removed from the network 42 minutes and 32 seconds after the simulation starts. Nodes 4, 7, 10, 11 and 13 are harvester nodes and the rest are ordinary nodes. All grey nodes are DTEE nodes and all blue nodes are DTOR nodes according to their algorithm.

In CDSSEHA, the ordinary node 3 runs out of energy in the network with CDS including 5 harvester and 10 ordinary nodes as shown in Figure 4. In this scenario, the first node was removed from the network 51 minutes and 43 seconds after the simulation starts. Nodes 4, 7, 10, 11 and 13 are harvester nodes and the rest are ordinary nodes. All grey nodes are DTEE nodes, all blue nodes are DTOR nodes, and all red nodes are selected DTOR nodes through DTOR connection algorithm in a WSN with CDS.

Consequently, the maximum lifetime belongs to the scenario where CDSSEHA is conducted. Despite the fact that FLDH method uses harvester nodes, it has increased the lifetime just slightly.To achieve a fair comparison, effective lifetimes of the applications using these methods are also given in Table 3.

In the following subsection, FLD, FLDH, APRCDS and CDSSEHA methods are tested on 50 different WSN topologies including 30 and 50 nodes. The simulations are called FLD-30, FLD-50, FLDH-30, FLDH-50, APRCDS-30, APRCDS-50, CDSSEHA-30 and CDSSEHA-50, respectively. In the first set of simulations, the generated topologies are used to test the communication method explained in Section 4.1. In the second set of simulations, CDSSEHA algorithm is conducted on the same topologies. But, in these topologies 33.3% of the network are assigned as harvester nodes. As the last set of simulations, the same topologies including harvester nodes are used to test the communication method FLDH mentioned in Section 4.2. Moreover, in order to decrease the time the simulation takes, the energy consumption levels occured in communication phase of Algorithm 3 are increased in the same scale for all methods.



(a) The average energy level of the nodes
(b) The average lifetime of the applications
(c) The consumed energy levels by the unconnected nodes
(d) The average energy level of the applications during sample application

FIGURE 5. The simulation results of the methods FLD, FLDH, APRCDS and CDSSEHA for 30 and 50 nodes.

TABLE 3. Effective lifetime results.

Application	Effective Lifetime (min)
FLD-30	21.969
FLD-50	23.686
FLDH-30	22.2253
FLDH-50	24.093
APRCDS-30	63.145
APRCDS-50	64.034
CDSSEHA-30	151.032
CDSSEHA-50	117.014

C. SIMULATION RESULTS

The changes in the energy levels of the nodes for all mentioned methods are shown in Figure 5. The energy consumption rates in CDSSEHA algorithm are 92%, 91% lower compared to those in FLD-30 and FLD-50 and 90%, 89% lower compared to those in FLDH-30 and FLDH-50, 49%, 59% lower compared to those in APRCDS-30 and APRCDS-50 respectively which are obtained by Table 4. Since, the time axes of the results reach the average lifetime of the applications, the average energy levels of not all of the nodes in the methods are exhausted completely.

TABLE 4. Energy consumption rates.

Algorithm	Energy Consumption Rate (mJ/min.)
FLD-30	1001301.461
FLD-50	665528.611
FLDH-30	754945.778
FLDH-50	532066.451
APRCDS-30	139562.193
APRCDS-50	132718.010
CDSSEHA-30	71012.934
CDSSEHA-50	53813.839

The average lifetimes of the simulations for the four methods are given in Figure 5(b) in milliseconds. The average energy consumption levels of the nodes that run out of energy the earliest are also shown in Figure 5(c).

The average energy levels of the nodes in FLD, FLDH and APRCDS methods by excluding the neighbour discovery phase of the nodes and the CDS construction phase are given in Figure 5(d). Besides the time required for the neighbourhood discovery, the time required for the construction of the CDS constitutes an overhead for the agricultural application. This overhead is eliminated from the whole lifetime to get

TABLE 5. The CDS construction overhead of CDSSEHA and APRCDS.

Algorithm	Time	Energy	Energy Ratio
CDSSEHA-30	26 min.	2805-mJ	0.000091
CDSSEHA-50	33 min.	3832-mJ	0.000120
APRCDS-30	31 min.	6245-mJ	0.000202
APRCDS-50	48 min	11824-mJ	0.000384

TABLE 6. Effective lifetime of WSNs compared to CDSSEHA.

Algorithm	Lifetime Comparison
FLD-30	CDSSEHA is 6 times longer
FLD-50	CDSSEHA is 4 times longer
FLDH-30	CDSSEHA is 6 times longer
FLDH-50	CDSSEHA is 4 times longer
APRCDS-30	CDSSEHA is 1.4 times longer
APRCDS-50	CDSSEHA is 0.8 times longer

the real lifetime of the agricultural application employed on CDSSEHA. The CDS construction overheads, i.e., consumed time, consumed energy for CDS construction, the ratio of consumed energy for CDS construction to total energy, are also measured and given in Table 5.

D. RESULTS ANALYSIS

According to the results, CDSSEHA achieves the maximum lifetime for both topology sizes. The CDSSEHA presents an effective lifetime (i.e. the CDS construction time is excluded) that is approximately 6 and 4 times longer in comparison to FLD-30 and FLD-50, respectively. In addition, CDSSEHA also surpasses APRCDS in terms of effective lifetime since it increases the lifetime by 1.4 and 0.8 times for the topologies having 30 and 50 nodes. Since FLDH simulations are conducted on networks comprising harvester nodes, the results of FLDH slightly outperform FLD-30 and FLD-50 in terms of lifetime by difference rates of 1.1% and 1.7%, in consecutive order which can be obtained by Table 3. The effective lifetime comparisons of the applications are given in Table 6.

Even though the proposed CDSSEHA and APRCDS surpass the flooding based methods, they require time and extra energy consumption for the construction of the CDS. In CDSSEHA, the energy consumption in CDS construction constitutes only 0.0091% and 0.0120% of the energy consumption of the whole application, whereas the algorithms' time usage takes 14% and 16% of the whole lifetime for networks with 30 and 50 nodes, respectively, which can be observed by Table 3 and Table 5. However, APRCDS algorithm consumes 33% and 43% of the lifetime for constructing CDS. Moreover, the CDS construction process of APRCDS entails the consumption of approximately 0.020% and 0.038% of the total energy. The results reveal that the new CDSSEHA algorithm is scalable with respect to the number of nodes in WSN. Since energy consumption rates in sample application are scaled to a higher value, in real world, the lifetime of the applications have longer lifetime. Therefore, the CDS construction overhead would become smaller in terms of time. Thus, the CDS construction overhead becomes negligible for real PA applications.

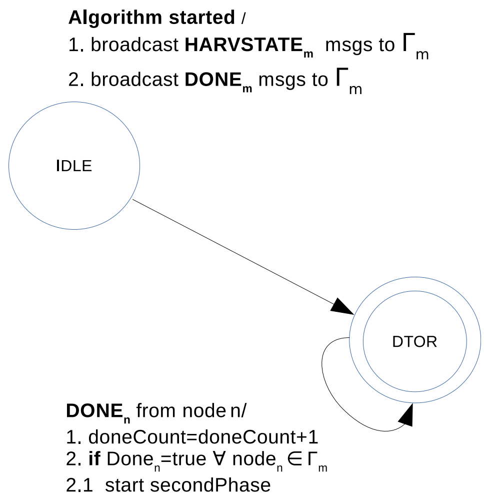


FIGURE 6. Finite state machine of a harvester node *m*.

Consequently, the analyses clearly show that using only harvester nodes is not enough to extend the lifetime of a WSN. The new proposed CDSSEHA algorithm prolongs the lifetime of a WSN applications in agriculture applications in comparison to other traditional methods already in use.

V. CONCLUSION

In this paper, CDSSEHA, a new distributed CDS algorithm on WSNs including solar energy harvester nodes to extend the lifetime of the WSNs used in precision agriculture applications is proposed. The backbone constructed by the new algorithm can be utilized as a communication substructure in precision agriculture, as mentioned, as well as in many other application areas. The algorithm selects the dominator nodes according to;

- the node type (harvester or ordinary),
- the remaining energy levels,
- the number of dominator neighbours.

Many other studies suggest incorporating energy harvester nodes in WSNs for extending the lifetime. However, there exist no sufficient research studies related to constructing CDS including harvester nodes in the area of precision agriculture. In the agricultural fields, solar energy is the most efficient source for energy harvesting amongst the environmental power sources. Indeed, energy harvesting is crucial to prolong the network lifetime. It is seen that solar energy harvesting with the support of CDS structure is saving more energy in real-world agricultural applications.

The proposed algorithm has been developed on SensEH simulator, which is based on Cooja, and compared with Al-Nabhan *et. al.*'s [20] energy efficient CDS construction algorithm, namely APRCDS and the traditional flooding methods used for the agricultural applications. According to the simulation results, it is clearly shown that using solar energy harvester nodes in agricultural WSN applications using the proposed CDSSEHA algorithm prolongs the network lifetime. The new algorithm also decreases the message

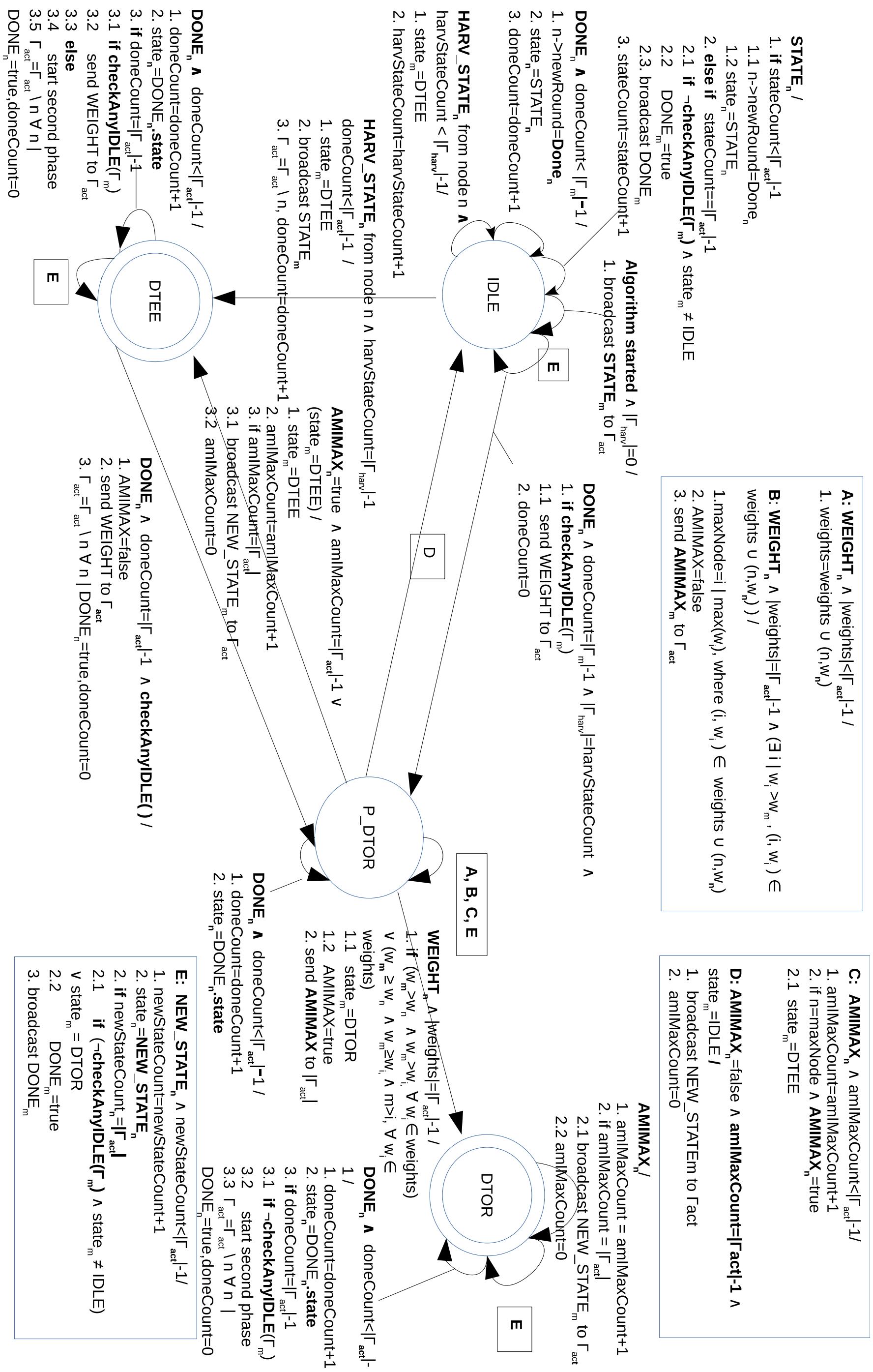


FIGURE 7. Finite state machine of an ordinary node *m*.

usage, reduces the energy consumption of the nodes and gathers data from the nodes continuously for monitoring the agriculture habitat. In addition, CDSSEHA also prevents data loss and collusion, and thus, it leads to lower energy consumption, as well. The simulation results also put forward that the positions of harvester and ordinary nodes, the neighbourhoods and the remaining energy levels result in differences in the results among separate simulations.

Besides the optimization of the proposed algorithms, the impacts of the harvester node positions on the lifetime of applications and on agricultural product qualities in the applied fields can be also considered as further study areas.

From the simulation results, it is noticed that, some nodes may have critical positions in the network topology. Due to their position, the battery depletion of these nodes may lead the network to be unconnected which results in the end of network lifetime. In the graph theory, these nodes are called cut vertices. For extending the network lifetime, the cut vertices in the network topology should be kept alive as long as possible. Therefore, as a future work, the parameters such as cut vertex detection and selection of them as harvester nodes will be considered.

APPENDIX. FINITE STATE MACHINES

See Figure 6 and Figure 7.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ''Wireless sensor networks: A survey,'' *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[2] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, and M. Ismail, ''Energy-efficient wireless sensor networks for precision agriculture: A review,'' *Sensors*, vol. 17, no. 8, p. 1781, 2017, doi: 10.3390/s17081781.

[3] Y. Zhang, C.-H. Feng, I. Demirkol, and W. B. Heinzelman, ''Energy-efficient duty cycle assignment for receiver-based convergecast in wireless sensor networks,'' in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[4] Z. Li, M. Li, and Y. Liu, ''Towards energy-fairness in asynchronous duty-cycling sensor networks,'' in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 801–809.

[5] S. Lai, ''Duty-cycled wireless sensor networks: Wakeup scheduling, routing, and broadcasting,'' Ph.D. dissertation, Virginia Tech, Blacksburg, VA, USA, 2010.

[6] W. Shi, W. Liu, T. Wang, Z. Zeng, and G. Zhi, ''Adding duty cycle only in connected dominating sets for energy efficient and fast data collection,'' *IEEE Access*, vol. 7, pp. 120475–120499, 2019.

[7] X. Liu, M. Dong, Y. Liu, A. Liu, and N. N. Xiong, ''Construction low complexity and low delay CDS for big data code dissemination,'' *Complexity*, vol. 2018, pp. 1–19, Jun. 2018.

[8] F. K. Shaikh and S. Zeadally, ''Energy harvesting in wireless sensor networks: A comprehensive review,'' *Renew. Sustain. Energy Rev.*, vol. 55, pp. 1041–1054, Mar. 2016, doi: 10.1016/j.rser.2015.11.010.

[9] D. Lisiecki, P. Zhang, and O. Theel, ''CONE: A connected dominating set-based flooding protocol for wireless sensor networks,'' *Sensors*, vol. 19, no. 10, p. 2378, 2019.

[10] K. Erciyes, *Distributed Graph Algorithms for Computer Networks*. London, U.K.: Springer, 2013.

[11] O. Dagdeviren, K. Erciyes, and S. Tse, ''Semi-asynchronous and distributed weighted connected dominating set algorithms for wireless sensor networks,'' *Comput. Standards Interfaces*, vol. 42, pp. 143–156, Nov. 2015.

[12] L. Song, C. Liu, H. Huang, H. Du, and X. Jia, ''Minimum connected dominating set under routing cost constraint in wireless sensor networks with different transmission ranges,'' *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 546–559, Apr. 2019.

[13] K. S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman, ''Energy-harvesting wireless sensor networks (EH-WSNs): A review,'' *ACM Trans. Sensor Netw.*, vol. 14, no. 2, pp. 1–50, Apr. 2018, doi: 10.1145/3183338.

[14] C.-C. Hung, W.-C. Peng, and W.-C. Lee, ''Energy-aware set-covering approaches for approximate data collection in wireless sensor networks,'' *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 1993–2007, Nov. 2012.

[15] I. Dietrich and F. Dressler, ''On the lifetime of wireless sensor networks,'' *ACM Trans. Sensor Netw.*, vol. 5, no. 1, pp. 1–39, Feb. 2009.

[16] T. Shi, S. Cheng, Z. Cai, Y. Li, and J. Li, ''Exploring connected dominating sets in energy harvest networks,'' *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1803–1817, Jun. 2017, doi: 10.1109/tnet.2017.2657688.

[17] P. T. V. Bhuvaneswari, R. Balakumar, V. Vaidehi, and P. Balamuralidhar, ''Solar energy harvesting for wireless sensor networks,'' in *Proc. 1st Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2009, pp. 57–61.

[18] A. A. Babayo, M. H. Anisi, and I. Ali, ''A review on energy management schemes in energy harvesting wireless sensor networks,'' *Renew. Sustain. Energy Rev.*, vol. 76, pp. 1176–1184, Sep. 2017, doi: 10.1016/j.rser.2017.03.124.

[19] S. Senivasan, M. Drieberg, B. S. M. Singh, P. Sebastian, and L. H. Hiung, ''An MPPT micro solar energy harvester for wireless sensor networks,'' in *Proc. IEEE 13th Int. Colloq. Signal Process. Appl. (CSPA)*, Mar. 2017, pp. 159–163.

[20] N. Al-Nabhan, M. Al-Rodhaan, and A. Al-Dhelaan, ''Distributed energy-efficient approaches for connected dominating set construction in wireless sensor networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 6, Jun. 2014, Art. no. 625061.

[21] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*. Philadelphia, PA, USA: SIAM, 2000.

[22] N. Mukherjee, S. Neogy, and S. Roy, *Building Wireless Sensor Networks: Theoretical and Practical Perspectives*. Boca Raton, FL, USA: CRC Press, 2015.

[23] S. N. Daskalakis, G. Goussetis, S. D. Assimonis, M. M. Tentzeris, and A. Georgiadis, ''A uW Backscatter-Morse-Leaf sensor for low-power agricultural wireless sensor networks,'' *IEEE Sensors J.*, vol. 18, no. 19, pp. 7889–7898, Oct. 2018.

[24] R. Kamath, M. Balachandra, and S. Prabhu, ''Raspberry pi as visual sensor nodes in precision agriculture: A study,'' *IEEE Access*, vol. 7, pp. 45110–45122, 2019.

[25] R. Khan, I. Ali, M. Zakarya, M. Ahmad, M. Imran, and M. Shoaib, ''Technology-assisted decision support system for efficient water utilization: A real-time testbed for irrigation using wireless sensor networks,'' *IEEE Access*, vol. 6, pp. 25686–25697, 2018.

[26] N. S. Meda, T. G. Sadashiva, S. K. Ramani, and S. S. Iyengar, ''Mobile WSN testbed for agriculture: Plant monitoring system,'' in *Proc. 2nd Int. Conf. Emerg. Comput. Inf. Technol. (ICECIT)*, Dec. 2017, pp. 1–6.

[27] G. Villarrubia, J. F. D. Paz, D. H. D. L. Iglesia, and J. Bajo, ''Combining multi-agent systems and wireless sensor networks for monitoring crop irrigation,'' *Sensors*, vol. 17, no. 8, p. 1775, 2017.

[28] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, ''Energy efficient automated control of irrigation in agriculture by using wireless sensor networks,'' *Comput. Electron. Agricult.*, vol. 113, pp. 154–163, Apr. 2015, doi: 10.1016/j.compag.2015.02.004.

[29] D. Shinghal and N. Srivastava, ''Wireless sensor networks in agriculture: For potato farming,'' in *Neelam, Wireless Sensor Networks in Agriculture: For Potato Farming*. Rochester, NY, USA: SSRN, Sep. 2017.

[30] C. Hedley, J. Ekanayake, and P. Roudier, ''Wireless soil moisture sensor networks for precision irrigation scheduling,'' in *Proc. Workshop Abstr., Adv. Nutrient Manage., Gains Past-Goals Future*, 2012, p. 85.

[31] T. H. F. Khan and D. S. Kumar, ''Ambient crop field monitoring for improving context based agricultural by mobile sink in WSN,'' *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 4, pp. 1431–1439, Apr. 2020.

[32] I. E. Radoi, J. Mann, and D. K. Arvind, ''Tracking and monitoring horses in the wild using wireless sensor networks,'' in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2015, pp. 732–739.

[33] J. Luque, D. Larios, E. Personal, J. Barbancho, and C. León, ''Evaluation of MPEG-7-Based audio descriptors for animal voice recognition over wireless acoustic sensor networks,'' *Sensors*, vol. 16, no. 5, p. 717, 2016.

[34] P. Loreti, A. Catini, M. De Luca, L. Bracciale, G. Gentile, and C. Di Natale, ''The design of an energy harvesting wireless sensor node for tracking pink iguanas,'' *Sensors*, vol. 19, no. 5, p. 985, 2019.

[35] T. Nath and M. Azharuddin, ''Application of wireless sensor networks for rhino protection against poachers in kaziranga national park,'' *AEU-Int. J. Electron. Commun.*, vol. 111, Nov. 2019, Art. no. 152882.

[36] L. Cheng, J. Niu, Y. Gu, C. Luo, and T. He, ''Achieving efficient reliable flooding in Low-Duty-Cycle wireless sensor networks,'' *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3676–3689, Dec. 2016.

[37] L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, and Y. Gu, ''Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks,'' *Comput. Netw.*, vol. 134, pp. 66–77, Apr. 2018.

[38] C. Luo, J. Yu, D. Li, H. Chen, Y. Hong, and L. Ni, ''A novel distributed algorithm for constructing virtual backbones in wireless sensor networks,'' *Comput. Netw.*, vol. 146, pp. 104–114, Dec. 2018, doi: 10.1016/j.comnet.2018.09.016.

[39] J. P. Mohanty, C. Mandal, and C. Reade, ''Distributed construction of minimum connected dominating set in wireless sensor network using two-hop information,'' *Comput. Netw.*, vol. 123, pp. 137–152, Aug. 2017, doi: 10.1016/j.comnet.2017.05.017.

[40] X. Qi and Z. Yang, ''A Multi-MCDS scheduling based virtual backbone network construction and maintenance algorithm in FANET,'' in *Proc. MATEC Web Conf.*, vol. 189, p. 03013, Jan. 2018, doi: 10.1051/matecconf/201818903013.

[41] T. Pino, S. Choudhury, and F. Al-Turjman, ''Dominating set algorithms for wireless sensor networks survivability,'' *IEEE Access*, vol. 6, pp. 17527–17532, 2018.

[42] K. Kwon, J. Yang, and Y. Yoo, ''Adaptive control of the packet transmission period with solar energy harvesting prediction in wireless sensor networks,'' *Sensors*, vol. 15, no. 5, pp. 9741–9755, 2015, 10.3390/s150509741.

[43] J. Hou and Y. Gao, "Greenhouse wireless sensor network monitoring system design based on solar energy," in *Proc. Int. Conf. Challenges Environ. Sci. Comput. Eng.*, Mar. 2010, pp. 475–479, doi: 10.1109/cesce.2010.274.
[44] J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. A. Porta-Gandara, "Automated irrigation system using a wireless sensor network and GPRS module," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 1, pp. 166–176, Jan. 2014.
[45] O. López-Lapeña and R. Pallas-Areny, "Solar energy radiation measurement with a low–power solar energy harvester," *Comput. Electron. Agricult.*, vol. 151, pp. 150–155, Aug. 2018, doi: 10.1016/j.compag.2018.06.011.
[46] T. Zou, S. Lin, Q. Feng, and Y. Chen, "Energy-efficient control with harvesting predictions for solar-powered wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 53, 2016.
[47] H. Sharma, A. Haque, and Z. A. Jaffery, "Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101966.
[48] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. Program. Lang. Syst. (TOPLAS)*, vol. 5, no. 1, pp. 66–77, Jan. 1983.
[49] Moteiv Corporation. *Tmote Sky Datasheet*. Accessed: Jun. 20, 2019. [Online]. Available: https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf,2006,online on The University of St Andrews
[50] M. T. Nguyen and K. A. Teague, "Compressive sensing based random walk routing in wireless sensor networks," *Ad Hoc Netw.*, vol. 54, pp. 99–110, Jan. 2017, doi: 10.1016/j.adhoc.2016.10.009.
[51] R. Dall'Ora, U. Raza, D. Brunelli, and G. P. Picco, "SensEH: From simulation to deployment of energy harvesting wireless sensor networks," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw. Workshops*, Sep. 2014, pp. 566–573.
[52] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," Swedish Inst. Comput. Sci., Kista, Sweden, Tech. Rep. T2011:05, 2011.
[53] Varta GmbH & Co. (2005). *V4906 Datasheet*. Accessed: Jun. 15, 2019. [Online]. Available: http://www.farnell.com/datasheets/39632.pdf

OMER GULEC received the B.Sc. degree in electrical and electronics engineering and the M.Sc. degree in computer engineering from Pamukkale University, Denizli, Turkey, where he is currently pursuing the Ph.D. degree in electrical and electronics engineering. He is also a Lecturer with Pamukkale University. His research interests include wireless sensor networks, energy harvesting-based solutions, and distributed algorithms.

ELIF HAYTAOGLU received the Ph.D. degree in information technologies area from Ege University, İzmir, Turkey. She is currently an Assistant Professor with the Department of Computer Engineering, Pamukkale University, Denizli, Turkey. Her research interests include distributed storage systems, wireless sensor networks, energy harvesting-based solutions, regenerating codes, network coding, and computer networks.

SEZAI TOKAT received the B.Sc. and Ph.D. degrees in computer and control engineering from Istanbul Technical University, and the M.Sc. degree in systems and control engineering from Bogazici University, Istanbul, Turkey. He is currently a Professor with the Department of Computer Engineering, Pamukkale University, Denizli, Turkey. His research interests include control systems, artificial intelligence, and vehicle routing problems.