A Distributed Multilevel Data Collection Mechanism Using Bus in WSNs

Chih-Yung Chang, *Member, IEEE*, Chung-Chih Lin, Wen-Hwa Liao, *Member, IEEE*, and Diptendu Sinha Roy

***Abstract*—Data collection using mobile sink has received much attention in the past years. In the literature, most studies assumed that the mobile sink was controllable and the proposed algorithm made decisions to control the path, speed, and stop locations. However, there still exists some challenges in using the controllable mobile sink, including limited energy and the adaptive capability for overcoming the obstacles. This article considers the bus as the uncontrollable mobile sink to collect data from a given set of deployed static sensors. A distributed data collection mechanism, called energy balanced multihop data collection (EBMDC), is proposed that aims at maximizing the transmitted data volume from sensors to the bus while prolonging the network lifetime of the wireless sensor networks (WSNs). To prolong the network lifetime, each sensor node distributes its data to its multiple parents in trees according to their remaining energies. Then, each anchor node locally schedules its transmission slots based on its and its neighbor's data volumes and remaining energies such that all data can be forwarded to the bus without collision and the lifetime of each anchor can be balanced. Experimental studies reveal that the proposed EBMDC algorithm outperforms related studies in terms of throughput, network lifetime, slot utilization, data loss ratio, and fairness.**

***Index Terms*—Data collection, energy balance, multihop, slot scheduling, wireless sensor network.**

I. Introduction

Wireless sensor networks (WSNs) have been widely applied in many applications. A wireless sensor network consists of few sink nodes and a large number of sensor nodes that are randomly deployed in a given region. Each sensor is usually embedded with sensing, communication, and computing components and can perform monitoring task and report the reading to the sink node.

Data collection is one of the most important research topics in WSNs and has received much attention in the past years. Since sensors are battery powered, energy conservation has been an important issue when designing the data collection mechanisms. In literature, many studies [1]–[4] have proposed data collection algorithms. These studies can be further be classified into two classes: fixed sink and mobile sink.

Manuscript received April 12, 2020; revised July 3, 2020; accepted August 3, 2020. Date of publication August 28, 2020; date of current version August 26, 2021. This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2622-E-032-001-CC3, Grant MOST 109-2221-E-032 -026 and Grant MOST 108-2622-E-032 -002 -CC3. *(Corresponding author: Chih-Yung Chang.)*

Chih-Yung Chang and Chung-Chih Lin are with the Tamkang University, Taipei 10650, Taiwan (e-mail: cychang@mail.tku.edu.tw; 804410016@s04.tku.edu.tw).

Wen-Hwa Liao is with the National Taipei University of Business, Taipei 10051, Taiwan (e-mail: whliao@ntub.edu.tw).

Diptendu Sinha Roy is with the National Institute of Technology, Delhi 110040, India (e-mail: diptendu.sr@nitm.ac.in).

Digital Object Identifier 10.1109/JSYST.2020.3015553

In the class of fixed sink node, studies [5]–[8] proposed data collection approaches which aimed to construct a tree topology for sensor nodes to transmit their data to the fixed sink. However, the sensors neighboring the fixed sink usually have heavy forwarding workloads, leading to the situation of energy exhaustion.

To overcome the problems of unbalanced energy consumption and network disconnection, many studies [9]–[12] considered the mobile sink which visited each sensor to collect data from them. These studies can be further partitioned into two classes: the controllable and uncontrollable mobile sinks. The controllable mobile sink was usually implemented by the mobile device which is controlled by an algorithm. By applying the proposed algorithm, the mobile sink can move along the constructed path and stop for collecting data until it returns to the base station. A typical example is the MobiCar [13], which is used for collecting data from static sensors. Most of them selected some anchor nodes from the sensors and constructed a path to visit each anchor for data collections. However, the challenges of controllable mobile sink, including high hardware cost, limited energy as well as the unpredictable environmental statuses, such as obstacles, are still needed to be overcome.

Another class of studies considered the uncontrollable mobile sink to collect data from the deployed static sensors. Typical examples of the uncontrollable mobile sinks were animals, climbers, buses, or bicycles which were embedded with simple data collection hardware components such that they can wirelessly communicate with the static sensors even though they are moving. Using an uncontrollable mobile sink to collect data has several advantages. The first one is the lower cost as compared with the controllable mobile sink which requires customized hardware. The bus used in this article only needs to be embedded with a simple transceiver. The second advantage is that there is no need to develop a special purpose mobile sink that should be able to detect and overcome the obstacles. The third advantage is that there is no energy constraint for the uncontrollable mobile sink like bus.

However, there are some challenges needed to be overcome when using an uncontrollable mobile sink to collect data. One challenge is the throughput constraint. That is, the velocity and stop time of the bus are uncontrollable [14], which might constrain the contacting time between the bus and the anchors. Under this constraint, how to maximize throughput is a big challenge. The proposed EBMDC mechanism makes good use of the shared slots and adopts the transmission power adjusting scheme to increase maximal throughput. The second challenge is to prolong the network lifetime of the WSNs. In general, the anchors will consume more energy than the other sensor nodes. The proposed EBMDC mechanism adopts the transmission power adjusting scheme to reduce the energy consumption of those anchors which have lower remaining energy. As a result, energy balancing among anchors can be achieved. The third challenge is the collision problem which occurs when the mobile sink falls in the overlapped communication ranges of two or more anchors. The proposed EBMDC mechanism partitions the contacting time into several slots. Then, the neighbors negotiate the schedule of the shared slots in a distributed manner.

This article considers the scenario that a number of static sensors have been randomly deployed in a big region for environmental monitoring. The buses or official vehicles are considered as the mobile sinks which are uncontrolled in terms of stops and arrival time. Each bus is assumed to move along the road and is capable of receiving the data from the anchor nodes which are those sensors located within its communication ranges.

The objective of this article is to achieve two purposes: maximizing the data amount transmitted from anchors to the bus and prolonging the lifetime of the given WSN. By applying the proposed EBMDC algorithm, each anchor negotiates with neighboring anchors and locally schedules the time slots for data transmissions based on the data volume, energy consumption, and the contacting time slots with the bus. The proposed EBMDC adopts the power level adjustment mechanism such that the anchor with higher remaining energy can increase its transmission power for increasing the transmission rate and hence release more shared time slots for the neighboring anchors, aiming at jointly balancing the lifetime between neighbors and guaranteeing that the data can be transmitted from each anchor to the bus. As a result, the proposed EBMDC helps neighboring anchors better utilize the time and energy resources, aiming to maximize the transmitted data volume and prolong the lifetime of the anchor with the least remaining energy.

The key contributions of the proposed EBMDC are itemized as follows.

1) *A larger amount of collected data:* By applying the distributed EBMDC, the anchors having more data can be locally allocated more transmission opportunities. Moreover, the anchors with higher remaining energy can enlarge their transmission power to increase the transmission rate. This helps reserve more shared time slots to be borrowed by the neighboring anchors that buffer more data. Compared with the existing studies [15], [16], the amount of collected data can be significantly improved.
2) *Achieving the energy balance of anchors and prolonging the network lifetime of the WSNs:* The power control mechanisms further balance the remaining energy of the anchors. The anchors with higher remaining energy can enlarge their transmission power to speed up the data transmission. This can release more time slots for those anchors that have lower remaining energy. Since the length of data transmission time is increased, the anchors with low remaining energy can adopt low power for data transmission, thus increasing its lifetime. Compared with existing studies [15], [16], the network lifetime of the WSNs can be significantly improved.
3) *Fair resource allocation:* The proposed EBMDC guarantees that the anchors having more data but fewer connection time with the bus can be allocated more shared time slots. Moreover, EBMDC also dynamically adopts the power level of each anchor such that the anchor with lower remaining energy can borrow the shared time slots from the neighboring anchor with higher remaining energy. Compared with existing studies [15], [16], the fairness of resource allocation in terms of shared time slots and remaining energy can be significantly improved.

The remaining part of this article is organized as follows. Section II presents the existing related studies. Section III introduces the network environment and problem formulation. Section IV presents the proposed distributed data collection mechanism. Section V reveals the experimental study and compares the performance of the proposed EBMDC algorithm against the related studies. Finally, a conclusion of the proposed algorithm and future work are drawn.

II. Related Work

In the literature, previous studies which proposed the data collection mechanisms can be mainly divided into two classes: controllable mobile sink and uncontrollable mobile sink.

A. *Controllable Mobile Sink*

In the first category, the mobile sink visited all or some of the sensors such that the visited sensors can directly transmit their data to the mobile sink. In study [17], a path construction algorithm was proposed to establish a path along which the mobile sink visits all the sensors for data collection before the sensors buffer overflow. All sensors directly transmitted their data to the mobile sink without creating the forwarding load, hence balancing their energy consumptions. However, the long path increased the delay of data collection, especially when the size of the network was large. Sugihara and Gupta [18] proposed an algorithm to construct a shortest path for minimizing the delay of data collection. The work in [19] proposed a progressive optimization-based scheme to construct a shortest path for minimizing the delay of data collection. The path construction problem was formulated as an integer linear programming problem. Though the proposed algorithm is easy to be applied to the wireless sensor networks, it is very difficult to be applied in a large sensor network because of high computational complexity and long time for data collection.

The above-mentioned studies applied mobile sink to visit each sensor can cope with the problem of unbalanced forwarding workload. However, when the network is large, it leads to long propagation delay. To overcome this problem, many studies [11]–[13], [20] selected some nodes to play the role of anchors. All the sensors transmitted their data to the nearest anchor; hence, the mobile sink only needed to visit anchors, instead of visiting all sensor nodes. All the other sensors are organized as several trees each of which is rooted by one anchor. The mobile sink only visited the anchors for collecting the data since all the sensor nodes will transmit their data to its tree root.

Almi'ani *et al.* [21] proposed a cluster-based algorithm to find efficient paths for mobile sinks. The algorithm attempts to partition the network into energy-aware clusters, such that in each cluster, a single cluster head is eventually selected. Hence, the mobile sink can move along the path for collecting data from all cluster heads. Though above-mentioned algorithms constructed a path for the mobile sink to collect data, the hardware cost and limited energy still need to be overcome.

B. Uncontrollable Mobile Sink

Some other work used the uncontrollable mobile sinks for data collection from the sensor networks. Similarly study [22] proposed a data collection algorithm in an energy renewable sensor network using a mobile sink that travelled along a predefined path by adopting multirate data transmission mechanisms. Since the mobile sinks were uncontrollable, the time required for collecting all data is difficult to be predicted, though there is no energy consumption problem. Some other previous work assumed that the mobile sink is uncontrollable but predictable. In [23], the vehicles were used as the mobile sinks which move along certain paths. An algorithm was proposed to collect data from one-hop sensors, called anchors, to the vehicles periodically. All the other sensors send their data to the anchors in a multihop manner. The work in [24] proposed an algorithm aiming to predict the arrival time of the uncontrollable mobile sink. However, the mobile sink visiting each sensor is time consuming, which leads to a long delay for the data collection. Study [25] considered the public transport vehicles to visit all sensors. They proposed a graph theory-based data gathering scheme aiming to optimize the energy consumption of all nodes. Though [23]–[25] adopted uncontrollable mobile sinks to perform the task of data collection, most of them did not consider the issues of transmission scheduling and throughput enhancement. This article proposed a distributed EBMDC algorithm which aims to achieve the two purposes including maximizing the amount of collected data from the static sensors to the bus and prolonging the network lifetime.

III. Network Environment and Problem Formulation

This section presents the environment of wireless sensor network, problem formulation as well as the goal and constraint of this article.

A. Network Environment

This article considers a WSN which consists of a set of *n* static sensors $S=\{s_1, s_2, \ldots, s_n\}$ randomly deployed in a monitoring region. There is a road *L* passing through the region. Each node is assumed to be aware of its unique ID, location, remaining energy, and the buffer capacity. Through the exchange of beacon messages with one-hop neighbors, each sensor can collect neighboring information, including IDs, locations, remaining energy, and size of data in buffer. A public bus *b* is considered as the mobile sink which periodically moves along the road *L* at an average velocity v. Let $A = \{a_1, a_2, \ldots, a_m\}$ denote the set of *m* anchors which are the one-hop neighboring sensors of bus *b*. It is obvious that *A* is the subset of *S* and $m \leq n$. The anchors are responsible for forwarding the data of other sensors in *S* to the bus.

B. Problem Formulation

This article aims to propose a distributed algorithm for data collection, aiming to maximize the received data from anchor nodes to bus *b* while prolonging the network lifetime of *S*. Unlike the controllable mobile sink, the bus *b* moves only along the road, resulting in the challenge that the bus *b* cannot receive data from each static sensor directly. Only anchors $a_i \in A$ can directly send data to the bus when the bus falls in their communication ranges. Hence, the farther sensors can only forward their data to the anchors in a multihop manner.

Let notation $|L|$ denote the length of the segment of *L* covered by *A*. Let T denote the total time length of one round which is measured by the contacting time between bus *b* and anchors in A. That is, the mobile sink *b* can collect data from the anchors during T in each round, where $T = |L|/v$. Let T be equally partitioned into q time slots. Let π denote the length of each partitioned time slot. We have

$$\pi = T/q = |L| / (v \times q).$$

The T can be presented as $T = [t_1, t_2 \ldots, t_q]$ and we have

$$|T| = \pi \times q.$$

Let e_i denote the initial energy of each sensor s_i. Let d_i denote the distance between sensor s_i and *L*. Assume that there is a multitree topology formed by the sensors. Along with the tree topologies, all sensors can forward data to the bus in a multihop manner. Each tree is rooted by one anchor $a_i \in A$. Let l_i denote the level of tree topology of sensor s_i. The sensors that play the root of each tree are marked with level 0 and the children of a node in level l_i are marked with level l_{i+1}. It is obvious that $A = \{a_i | \; l_i = 0, a_i \in S\}$ denotes the set of anchor nodes which are the set of sensors in level 0. Let $\tilde{A}$ denote the set of nonanchor nodes. We have

$$S = A \cup \tilde{A} \text{ and } \tilde{A} = S \backslash A.$$

Let r_i denote the sensing rate of sensor $s_i \in S$. Let D_i^{rec}, D_i^{gen}, and D_i^{tra} denote the data volume received, generated, and transmitted by sensor $s_i \in S$, respectively. We have

$$D_i^{\text{gen}} = \sum_{i=0}^{q} (t_i \times r_i) \; .$$

Let P_i denote the set of parents of sensor s_i. Let $C(s_i)$ denote the set of all child nodes which treat sensor s_i as their forwarder such as

$$C(s_i) = \{s_k | s_i \in P_k, \; \forall s_i, \; s_k \in \tilde{A}.\}$$

There, we have

$$D_i^{\text{rec}} = \sum_{s_k \in C(s_i)} (D_k^{\text{rec}} + D_k^{\text{gen}}).$$

For each anchor node $a_i \in A$, let $\tilde{D}_i$ denote the remaining data volume of anchor a_i in each round. We have

$$\tilde{D}_i = D_i^{\text{rec}} + D_i^{\text{gen}} - D_i^{\text{tra}}, \forall a_i \in A.$$

The communication range of each anchor node can overlap with the road *L* and can directly transmit data to the bus. However, the contact time of the bus and each anchor highly depends on the overlapped region between the communication ranges of bus and each anchor. The data bufferred by each anchor also depend on the tree topology of the WSN and the amount of remaining data of each anchor also depend on data transmission scheduling. The main goal of this article aims to transmit maximal data volume from anchors to the bus in each round. Equation (1) reflects the main goal of this article given as

$$\text{Min}\left(\sum_{s_i\in A}(D_i^{\text{rec}}+D_i^{\text{gen}}-D_i^{\text{tra}})\right). \quad (1)$$

In addition to minimize the total data remaining in anchors, as shown in (1), another important goal is to prolong the network lifetime of the given wireless sensor network. The following models the energy consumption of each sensor s_i. Let e_i^{tra}, e_i^{rec}, and e_i^{gen} denote energy consumed by sensor s_i for transmitting, receiving, and generating one bit data, respectively.

Let b_i^{tra}, b_i^{gen}, and b_i^{rec} denote the transmitted, generated, and received data volume of sensor s_i in each round, respectively. Let E_i^{tra}, E_i^{gen}, and E_i^{rec} denote the energy consumption of sensor s_i for transmitting, generating, and receiving data in each round, respectively, such as

$$E_i^{\text{tra}}=b_i^{\text{tra}}\times e_i^{\text{tra}}$$
$$E_i^{\text{gen}}=b_i^{\text{gen}}\times e_i^{\text{gen}}$$
$$E_i^{\text{rec}}=b_i^{\text{rec}}\times e_i^{\text{rec}}.$$

The total energy consumption of sensor s_i in each round, denoted by E_i, is measured by

$$E_i=E_i^{\text{tra}}+E_i^{\text{gen}}+E_i^{\text{rec}}.$$

The second objective of this article is to minimize the energy consumption of the sensor with the lowest remaining energy. Let E be the battery capacity of each sensor and $\tilde{E}_i$ denote the remaining energy of sensor s_i. Equation (2) presents the second goal of this article as

$$\text{Max}\left(\min_{s_i\in S}\left(\tilde{E}_i=E-E_i\right)\right). \quad (2)$$

The constraints (3)–(6) should be satisfied when the goals given in (1) and (2) are achieved. Let $N(s_i)$ denote the set of neighboring sensors of sensor s_i. The *forwarder existence constraint* asks each sensor s_i in $\tilde{A}$ to find a neighboring node s_j with a lower level to forward its data to bus *b*.

1) Forwarder existence constraint:

$$\forall s_i\in\tilde{A},\ \exists\, s_j\in N(s_i)\cap S \text{ such that } l_j<l_i. \quad (3)$$

The second constraint, called *collision-free constraint*, guarantees that each slot in T_b can only allow at most one anchor node a_k to transmit data to bus b. Recall that T_b can be presented as $T_b=[t_1,t_2\ldots,t_q]$. Let Boolean variable $\lambda_{i,j}$ denote whether or not anchor node $a_i\in A$ can transmit data to bus b in slot t_j in the final schedule such as

$$\lambda_{i,j}=\begin{cases}1 & s_i \text{ transmits data to bus } b \text{ in } t_j\in T_b\\ 0 & \text{otherwise.}\end{cases}$$

The following collision-free constraint guarantees that there is no collision for a feasible solution of the transmission schedule.

2) Collision-free constraint:

$$\sum_{s_i\in A}\lambda_{i,j}\times t_j\le 1,\ \forall\, t_j\in T_b\,. \quad (4)$$

The third constraint, called *flow constraint*, guarantees that each nonanchor node should transmit all its data to the forwarder in each round.

3) Flow constraint:

$$D_i^{\text{tra}}=D_i^{\text{gen}}+\sum_{s_k\in C(s_i)}(D_k^{\text{rec}}+D_k^{\text{gen}}),\ \forall s_i\in\tilde{A}. \quad (5)$$

The fourth constraint, called *buffer constraint*, guarantees that buffer capacity of each sensor $s_i\in S$ should be enough for storing those data that are received from sensor nodes of $C(s_i)$ and generated by itself in each round. Let D_i^{cap} denote the buffer capacity of each sensor $s_i\in S$.

4) Buffer constraint:

$$D_i^{\text{cap}}\ge D_i^{\text{gen}}+\sum_{s_k\in C(s_i)}(D_k^{\text{rec}}+D_k^{\text{gen}}),\ \forall s_i\in S. \quad (6)$$

IV. Energy Balanced Multihop Data Collection (EBMDC) Algorithm

This section presents the proposed algorithm which aims to achieve maximal throughput from WSNs to the bus and prolong the network lifetime of the WSNs.

There are some challenges that needs to be overcome when developing the data collection algorithm. The static sensors are randomly deployed in the monitoring region. One of the challenge is to construct an efficient tree such that each sensor can forward its data to bus *b* along with the tree topology. This article constructs a multitree topology where each sensor may have several parents and dynamically send its data to the parents according to their remaining energies. The sensors forward their data to the anchors along the constructed tree topologies in a multihop manner. To avoid the occurrences of collision and contention, the transmission scheduling algorithm applies the TDMA scheme [27] when sensors forward data along the tree topologies.

The second challenge is the energy balance among all anchors. It is assumed that all sensors are battery powered. Since the anchor nodes have different remaining energies, the data forwarding load from each sensor to the anchor should be balanced according to the remaining energy of each anchor node.

Third, the contacting time between anchor $a_k\in A$ and bus b is limited. Each anchor a_k can only transmit data in the allocated slots. The contacting time of neighboring anchors might be overlapped. Therefore, a careful scheduling mechanism should be developed for anchor nodes to maximize the transmission data from anchor nodes to the bus by considering fairness and collision issues.

The proposed algorithm is organized into three phases. In the first phase, each sensor $s_i\in S$ exchanges information with neighbors $s_j\in N(s_i)$. According to the information, each anchor node $a_k\in A$ will construct a weight-based tree topology, aiming to receive data from sensors in a multihop manner and then forward data to bus *b*. In the second phase, each sensor aims to distribute its data to its parents based on the parents' remaining energies. As a result, all the data of the sensors can be forwarded to the anchors. In the third phase, each anchor node a_k will adopt the proposed data collection mechanism to schedule the transmission slots such that all data can be transmitted to the bus while the collisions and contention among neighboring anchors can be avoided.

A. Initialization Phase

This phase aims to construct multiple spanning trees for forwarding data from each sensor to bus. Initially, the concept of Voronoi diagram [26] is adopted to partition the region into different areas such that each sensor connects to the closest anchor. The Voronoi region of each anchor is determined by a linear equation which is a vertical line between two anchors. To avoid the message flooding from each anchor to all sensors, each anchor only sends tree construction message, which contains the Voronoi edge information, to those sensors belonging to its Voronoi region. There are three steps in this phase. The first step is *region partitioning* step which helps each anchor partition the region into different areas. The second step is *broadcasting* step which helps sensor exchange messages between neighbors for constructing the tree. The third step is the *forwarder identification* step which helps each sensor locally identify its forwarders and negotiate with its forwarders.

In the *region partitioning* step, each anchor node $a_k \in A$ will determine a linear equation as the Voronoi edge to partition the region into different areas. In the *broadcasting* step, each anchor node $a_k \in A$ will play the role of tree root and broadcast *Tree_Construction* packet, which contains fields, including Anchor Id, Anchor Location, Sender Id, Sender Location, Tree Level, and Voronoi Edge Equation, to its neighbors $N(a_k) \in S$. On receiving the packet, each sensor node $s_j \in \tilde{A}$ will replace the Sender Id field with its own ID. The Tree Level field is also increased by one. Then, the sensor rebroadcasts the message to its neighbors.

In the *forwarder identification* step, each sensor aims to identify its forwarders for constructing the tree. Each sensor will choose the closest anchor as its tree root and maintain the information including the Anchor Id and Anchor Location of the selected root. Then each sensor will select the sender nodes which have the lowest tree level as its forwarders, aiming to prolong the network lifetime. Let $A_k(s_i)$ denote the anchor node closest to sensor s_i. Each sensor node $s_j \in \tilde{A}$ selects neighbors $s_u \in N(s_j)$ which have lowest level to be the forwarder set P_j of node s_j. In addition, the level of sensor node s_j will increase by one according to the level of forwarder $s_u \in P_j$ such as

$$P_j = \{s_u \mid \min(l_u) \text{ where } s_u \in N(s_j) \text{ and } s_u.anchor\_\text{id} = s_j .anchor\_\text{id}\}$$

$$l_j = 1 + l_u.$$

Then each sensor node $s_j \in \tilde{A}$ replies to the forwarders with a *Forwarder_Identification* packet. Till now, the *multiforward tree* has been constructed.

In the proposed EBMDC algorithm, the farther sensors can only forward their data to the anchors in a multihop manner. For transmitting all data of the sensors to the bus, the proposed EBMDC algorithm will construct a tree for each anchor, aiming at collecting data from sensors to the anchor along the tree. In addition, for prolonging the lifetime of WSN, the links between each child node and its parent nodes are temporary links. The child node can dynamically construct a physical link to one of the parent nodes according to the weights of the parents.

Fig. 1 gives an example to explain the operations of this phase. As shown in Fig. 1(a), there are 52 sensors deployed along the road. Among these sensors, there are six sensors that play the anchor nodes which can contact with the bus. These anchors aim to collect data from the other sensors and then forward the collected data to the bus based on transmission schedule. Fig. 1(b) shows that all anchors use Voronoi edges to partition the region into different areas. Fig. 1(c) depicts the construction packet broadcasted by the anchor a_1. Fig. 1(d) further depicts the constructed multiple trees rooted by the anchor nodes.

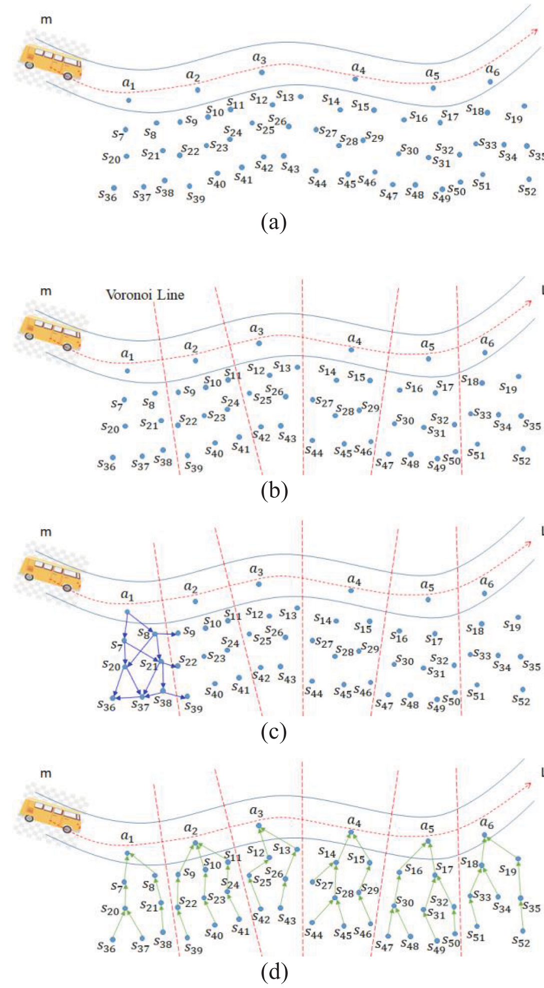


Fig. 1. Example to illustrate the operations of phase one. (a) Considered scenario. There are 52 sensor nodes deployed around the road. (b) Region partitioned by Voronoi edges. (c) Construction packet broadcasted by the anchor a_1. (d) Constructed multitree rooted by the anchor nodes.

B. Bottom-Up Requesting Phase

In the region partitioning step of the initialization phase, each anchor broadcasts the information including Anchor ID, Anchor Position, Sender ID, Sender Position, Tree Level, and Voronoi Edge Parameters in a top-down manner. In the forwarder identification step, each leaf node will select the neighboring nodes which have the minimum level as its candidate parents. Since the number of candidate parents might be more than one, the leaf node will treat all of these possible parents as its forwarders and include them in its parent set. The links between the leaf node and the parents are temporal links which can be dynamically constructed as a communication link between the leaf node and one of its parent nodes in the parent set, aiming to achieve the goal of energy balancing among parents. The operations of leaf

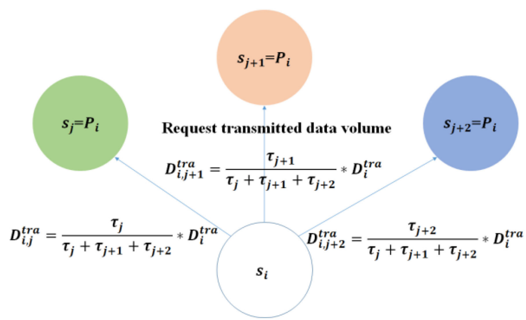


Fig. 2. Example that sensor s_i transmits its data to three parents, aiming for balancing remaining energies.

node are also similarly applied to each intermediate node of the tree in a bottom-up manner. As a result, each child node has a parent set and can dynamically determine the forwarder according to the weight of each candidate parent in the parent set.

In this phase, each sensor aims to calculate the ratio of data transmitted from itself to each of its parents. Recall that P_i denotes the set of parents of sensor s_i. To prolong the network lifetime, all sensors should be energy balanced. Let weight τ_j denote the remaining energy of sensor $s_j \in S$. Let $D_{i,j}^{\text{tra}}$ denote the data volume transmitted from sensor s_i to its parent $s_j \in P_i$. To balance the workload of each parent $s_j \in P_i$, the total data volume D_i^{tra} of sensor s_i should be distributed to each parent $s_j \in P_i$ according to their weights.

To balance the remaining energies of parent $s_j \in P_i$ of sensor s_i, the value of $D_{i,j}^{\text{tra}}$, which is the data amount that sensor s_i is sending to each parent, $s_j \in P_i$, will be determined as

$$D_{i,j}^{\text{tra}} = \frac{\tau_j}{\sum_{s_j \in P_i} \tau_j} \times D_i^{\text{tra}}. \tag{7}$$

Each sensor will adopt the bottom-up approach to forward its data to its parents by applying (7), aiming to balance the remaining energies of its parents. Fig. 2 illustrates an example that sensor s_i transmits its data to three parents according to their remaining energies.

C. BDC Determining Phase

In this phase, each anchor $a_k \in A$ aims to determine the data transmission schedule from itself to the bus such that the following goals can be achieved. The first goal is the energy conservation. The policy of the proposed algorithm is that each anchor should try to adopt a low power level to transmit its data to the bus. The second goal is that all data of each anchor can be fully transmitted. To achieve this, the transmission slots allocated to each anchor should be proportional to its data amount. The third one is that the collision and contention problems can be avoided. To achieve this goal, the shared time slots should be scheduled such that no contention and collision can occur. The following defines some notations which will be used to present the algorithms. Then the BDC *(bus-based data collection) mechanism* is proposed.

The following discusses the transmission schedule of each anchor a_k. The schedule includes the mode of a_k in each time slot, which can be either communication or sleep. In case that a_k stays in communication mode at some slot, the transmission power adopted by a_k should be further determined. Let T_k denote the contacting time period between bus b and anchor a_k. The period T_k consists of two subperiods: DT (dedicated timeslots) and ST (shared timeslots). Let DT denote the time period during which bus b can only contact anchor a_k. Conversely, let ST denote the time period during which bus b can contact anchor a_k and its neighboring anchors. Let ST be partitioned into two time periods: Left ST and Right ST, which are two sets of time slots that anchor a_k shares with the left or right neighbors, respectively.

Let T_k^{DT}, T_k^{LST}, and T_k^{RST} denote the time period of DT, Left ST, and Right ST of anchor a_k, respectively. Let $t_{k,j}$ denote the jth slot of time period T_k. Let $t_{k,j}^{\text{DT}}$, $t_{k,j}^{\text{LST}}$, and $t_{k,j}^{\text{RST}}$ denote the jth slot of T_k^{DT}, T_k^{LST}, and T_k^{RST} of anchor a_k, respectively. Since each anchor can exchange the location information with its neighbors, each anchor a_k can calculate the values of T_k^{DT}, T_k^{LST}, and T_k^{RST}, independently. Let p_k denote the transmission power level of each anchor a_k and $P = \{p_1, \ldots, p_m\}$ denote the set of m possible power levels. Let $p_{k,j}^{\text{threshold}}$ be the threshold of power level of anchor a_k, which guarantees that bus can safely receive the data transmitted by a_k in the jth slot. Let A_k denote the average available remaining energy of anchor a_k for each bit. This is

$$A_k = \frac{\tilde{E}_k}{D_k^{\text{tra}}}.$$

To balance the energy consumption of anchors in a distributed manner, each anchor should consider its remaining energy and the amount of data. To achieve this, each anchor compares its A_k value with those of the left and right neighboring anchors and determines its transmission power level. The key principle is that the anchor with larger A_k value should enlarge its transmission power, which increases its transmission rate. Based on the higher transmission rate, it can complete the data transmission earlier and hence reserve more shared time slots for neighboring anchors which have smaller remaining energies. As a result, the neighboring anchors with smaller A_k value can adopt a lower power level to further save its energy consumption, aiming at balancing the remaining energy among neighboring anchors. Let $A_k^{\min}$ be the minimal value of A_{k-1}, A_k, and A_{k+1}such as

$$A_k^{\min} = \text{Min}\left(A_{k-1}, A_k, A_{k+1}\right).$$

Therefore, each anchor a_k determines its initial power $p_{k,j}^{\text{init}}$ for each slot $t_{k,j}$ in T_k according to the following:

$$p_{k,j}^{\text{init}} = \begin{cases} p_{k,j}^{\text{threshold}} \times \frac{A_k}{A_{k-1}}, & \text{if } A_{k-1} = A_k^{\min} \\ p_{k,j}^{\text{threshold}}, & \text{if } A_k = A_k^{\min} \\ p_{k,j}^{\text{threshold}} \times \frac{A_k}{A_{k+1}}, & \text{if } A_{k+1} = A_k^{\min}. \end{cases} \tag{8}$$

Based on (8), the remaining energy of each anchor can be balanced. This occurs because the initial power level of each anchor has been determined according to the ratio of its and its neighbors' remaining energies.

To calculate the initial power of anchor a_k at slot $t_{k,j}$, the anchor a_k should be aware of the value of $A_k^{\min}$, which is determined by the anchor a_i with the smallest value A_i, where a_i could be a_{k-1}, a_k, or a_{k+1}. Therefore, each anchor a_i can adopt a backoff scheme to announce its value A_i. The backoff function should guarantee that the anchor a_i with smallest value A_i can broadcast its value A_i earliest. To achieve this, the backoff time of anchor should be positively related to its average available remaindering energy. Let B_i denote the backoff time of anchor a_i. We have

$$B_i = \alpha \times 2^{\beta \times A_i} \times \pi \tag{9}$$

where α and β are weight coefficients representing the important ratio between energy and data volume. According to the backoff policy given in (9), each anchor a_k can calculate its initial power $p_{k,j}^{\text{init}}$ at slot $t_{k,j}$ based on (8). When the backoff time of anchor a_k has been counted down to zero, it will broadcast its initial power and its ID, aiming to help neighboring anchors calculate their initial power.

Since the energy of each anchor has been balanced, the next issue is to determine the number of slots which should be occupied by each anchor a_k such that the data of anchor a_k can be completely transmitted to the bus. There are two possible methods to cope with the remaining data problem for each anchor a_k when its DT slots are not enough. One is to occupy some ST slots. The other is to enlarge its transmitting power level, which help increase the transmission rate of anchor a_k. The key policy is that each anchor a_k first calculates its requirement ratio of the ST slots by considering the transmission rate and the data amount of its own and its neighbors. Based on the DT and ST slots, if its data still can not be completely transmitted to the bus, it adopts the power control scheme to increase its power level.

The next step is that each anchor a_k should calculate the number of ST slots it should occupy. Recall that D_k^{rec}, D_k^{gen}, and D_k^{tra} denote the data volume received, generated, and transmitted by anchor a_k in each round, respectively. In addition, $\tilde{D}_k$ denote the remaining data volume of anchor a_k in each round such as

$$\tilde{D}_k = D_k^{\text{rec}} + D_k^{\text{gen}} - D_k^{\text{tra}}.$$

Let $\tilde{\rho}_k^{\text{DT}}$, $\tilde{\rho}_k^{\text{LST}}$, and $\tilde{\rho}_k^{\text{RST}}$denote the numbers of slots which will be reserved for anchor a_k in T_k^{DT}, T_k^{LST}, and T_k^{RST}, respectively. Similarly, let $p_{k,j}$ denote the power level used by anchor a_k for transmitting data in the *j*th slot, and $r_{k,j}$ denote the transmission rate under the power level $p_{k,j}$. Since slots in *ST* are shared by a_k and its neighboring anchors, anchor a_k evaluates the values of $\tilde{\rho}_k^{\text{LST}}$ and $\tilde{\rho}_k^{\text{RST}}$ according to the transmission rate and data amount of its own and its neighboring anchors. The relation between transmitted data volume and transmitted power can be calculated based on Shannon's theorem such as

$$C = B \times \log_2\left(1 + \frac{S}{N}\right)$$

where *C*, *B*, and $\frac{S}{N}$ denote the channel capacity, bandwidth of the channel, and signal-to-noise ratio (SNR), respectively. Let r_k^{avg} denote the average data rate of anchor a_k in this round as

$$r_k^{\text{avg}} = \frac{\sum_{j=0}^{|T_k|} r_{k,j}^{\text{init}}}{|T_k|}.$$

Let ω_k denote the average number of required slots of anchor a_k, whcih is determined based on average data rate and the buffered data volume D_k^{tra} in this round. We have

$$\omega_k = \frac{D_k^{\text{tra}}}{\gamma \times r_k^{\text{avg}}}$$

where γ is a weight coefficient representing the important ratio between energy and data volume. After that, anchor a_k broadcasts its ω_k value and receives the ω_{k-1} and ω_{k+1} from neighboring anchors and then calculates its own LST and RST based on the following:

$$\tilde{\rho}_k^{\text{LST}} = \left|T_k^{\text{LST}}\right| \times \frac{\omega_k}{\omega_{k-1} + \omega_k} \tag{10}$$

$$\tilde{\rho}_k^{\text{RST}} = \left|T_k^{\text{RST}}\right| \times \frac{\omega_k}{\omega_k + \omega_{k+1}}. \tag{11}$$

Since anchor a_k has calculated the values of $\tilde{\rho}_k^{\text{LST}}$ and $\tilde{\rho}_k^{\text{RST}}$, the schedules for T_k^{LST} and T_k^{RST} can be obtained. These schedules can significantly reduce the energy consumption of anchor a_k. The following presents the schedules of T_k^{LST} and T_k^{RST}:

$$\lambda_{k,j}^{\text{DT}} = 1, \text{ for all } t_{k,j}^{\text{DT}}\left[t_{k,1}^{\text{DT}}, t_{k,\,\tilde{\rho}_k^{\text{DT}}}^{\text{DT}}\right] \tag{12}$$

$$\lambda_{k,j}^{\text{LST}} = \begin{cases} 1 & \text{if } t_{k,j}^{\text{LST}}\left[t_{k,1}^{\text{LST}}, t_{k,\,\tilde{\rho}_k^{\text{LST}}}^{\text{LST}}\right] \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\lambda_{k,j}^{\text{RST}} = \begin{cases} 1 & \text{if } t_{k,j}^{\text{RST}}\left[t_{k,1}^{\text{RST}}, t_{k,\,\tilde{\rho}_k^{\text{RST}}}^{\text{RST}}\right] \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

In case that anchor a_k applies the initial power but some data still remains in its buffer even if it utilizes all slots in T_k^{DT}, $\tilde{T}_k^{\text{LST}}$, and $\tilde{T}_k^{\text{RST}}$, anchor a_k will increase its power level. This can help anchor a_k increase its transmission rate and hence completely transmit all of its data to the bus. Let D_k^{init} denote the transmitted data volume if anchor a_k applies the initial power and utilizes all slots in T_k^{DT}, $\tilde{T}_k^{\text{LST}}$, and $\tilde{T}_k^{\text{RST}}$to transmit its data to the bus such that

$$D_k^{\text{init}} = \sum_{j=1}^{\tilde{\rho}_k^{\text{DT}}} \lambda_{k,j}^{\text{DT}} \times r_{k,j}^{\text{init}} + \sum_{j=1}^{\tilde{\rho}_k^{\text{LST}}} \lambda_{k,j}^{\text{LST}} \times r_{k,j}^{\text{init}} + \sum_{j=1}^{\tilde{\rho}_k^{\text{RST}}} \lambda_{k,j}^{\text{RST}} \times r_{k,j}^{\text{init}}$$

where $r_{k,j}^{\text{init}}$ denotes the transmission rate under the power level $p_{k,j}^{\text{init}}$.

Anchor a_k will calculate the amount of increased data volume if it enlarges its own energy power level. Let $\delta_{k,j}^q$ denote the amount of increased rate in *j*th slot if anchor a_k increases transmitting power by *q* levels. Let $\delta_{D_k}^q$denote the amount of increased data transmitted from anchor a_k to the bus if anchor a_k increases transmitting power by *q* levels for all allocated slots in each round such that

$$\delta_{D_k}^q = \sum_{j=1}^{\tilde{\rho}_k^{\text{DT}}} \lambda_{k,j}^{\text{DT}} \times \delta_{k,j}^q + \sum_{j=1}^{\tilde{\rho}_k^{\text{LST}}} \lambda_{k,j}^{\text{LST}} \times \delta_{k,j}^q + \sum_{j=1}^{\tilde{\rho}_k^{\text{RST}}} \lambda_{k,j}^{\text{RST}} \times \delta_{k,j}^q.$$

Let $\delta_{p_k}^u$ denote the least increased *u* power levels of anchor a_k corresponding to p_k^{init}, required for transmitting all data to the bus. Since anchor a_k has calculated the values of δ_{D_k}, the value of $\delta^u_{p_k}$ can be simply calculated by applying

$$\delta^u_{p_k} = \arg\min_q \left(\frac{D_k^{\text{rec}} + D_k^{\text{gen}} - D_k^{\text{init}}}{\delta^q_{D_k}} \le 1 \right). \quad (15)$$

Therefore, to transmit all the buffered data of anchor a_k to the bus, anchor a_k should enlarge its power level to satisfy

$$D_k^{\text{tra}} = \sum_{j=1}^{\tilde{\rho}_k^{\text{DT}}} \lambda^{\text{DT}}_{k,j} \times \delta^u_{r_{k,j}} + \sum_{j=1}^{\tilde{\rho}_k^{\text{LST}}} \lambda^{\text{LST}}_{k,j} \times \delta^u_{r_{k,j}} + \sum_{j=1}^{\tilde{\rho}_k^{\text{RST}}} \lambda^{\text{RST}}_{k,j} \times \delta^u_{r_{k,j}} = D_k^{\text{init}} + \delta^u_{D_k} \ge D_k^{\text{rec}} + D_k^{\text{gen}} \quad (16)$$

where $\delta^u_{r_{k,j}}$ denotes the transmission rate under the power level $(p^{\text{init}}_{k,j} + \delta^u_{p_k})$.

Finally, the schedules of anchor a_k can be arranged, and anchor a_k can apply the additional power and utilize all slots in T_k^{DT}, $\tilde{T}_k^{\text{LST}}$, and $\tilde{T}_k^{\text{RST}}$to transmit all of its data to the bus in each round according to (12)–(14) and (16). By applying the operations designed in *BDC determining phase*, each anchor a_k tries to set up its initial power level according to the average available remaining energy of anchor a_k. Then, the schedule of shared timeslots is determined based on average data rate and the buffered data volume. Finally, each anchor a_k enlarges its power level to transmit all of its data to the bus.

Consequently, the first goal that each anchor a_k transmits maximal data volume from itself to the bus can be achieved while the remaining energy of the sensor with minimal remaining energy can be maximized in the proposed EBMDC.

The following summarizes the EBMDC algorithm as shown in Table I and Fig. 3.

TABLE I
MAIN NOTATIONS USED IN EBMDC ALGORITHM

Notation	Definition
S	the set of n static sensors, $S = \{s_1, s_2, \dots, s_n\}$
A	the set of m anchors, $A = \{a_1, a_2, \dots, a_m\}$
d_i	the distance between sensor s_i and the road.
D_i^{rec}	the data volume received by sensor s_i
D_i^{gen}	the data volume generated by sensor s_i
D_i^{tra}	the data volume transmitted by sensor s_i
$D_{i,j}^{tra}$	the data amount that sensor s_i sending to parent $s_j \in P_i$
$\widetilde{D}_i$	the remaining data volume of anchor a_i
$\tilde{E}_i$	the remaining energy of sensor s_i
E_i	the total energy consumption of sensor s_i
P_i	the set of parents of sensor s_i
$C(s_i)$	the set of all child nodes of sensor s_i
τ_j	The weight of sensor s_i
T_k	the contacting time period between bus and anchor a_k
$t_{k,j}^{DT}$	the j-th slot of T_k^{DT} of anchor a_k
$t_{k,j}^{LST}$	the j-th slot of T_k^{LST} of anchor a_k
$t_{k,j}^{RST}$	the j-th slot of T_k^{RST} of anchor a_k
l_i	the level of tree topology of sensor s_i
p_k	the transmission power level of anchor a_k
$p_{k,j}^{init}$	initial power for slot $t_{k,j}$ in T_k
$p_{k,j}$	the power level of anchor a_k in the j-th slot
$r_{k,j}$	the transmission rate under the power level $p_{k,j}$
$t_{k,j}$	the j-th slot of T_k of anchor a_k
$\lambda_{i,j}$	whether or not anchor node a_i can transmit data in slot t_j
$\tilde{\rho}_k^{DT}$	the numbers of slots reserved for anchor a_k in T_k^{DT}
$\tilde{\rho}_k^{LST}$	the numbers of slots reserved for anchor a_k in T_k^{LST}
$\tilde{\rho}_k^{RST}$	the numbers of slots reserved for anchor a_k in T_k^{RST}

V. PERFORMANCE EVALUATION

This section investigates the performance improvements of the proposed distributed EBMDC mechanism against the existing studies MobiCluster [15] and SEECH [16] mechanisms. The MobiCluster algorithm selected a proper number of sensors to be anchors which relay the data of other sensors to the bus. The SEECH algorithm selected some sensors to be cluster headers which forward the data of members in the same cluster to the bus along with the clustering topology. The experiments compare the three algorithms in terms of throughput, network lifetime, data loss ratio, slot utilization as well as fairness index of transmission data. The MATLAB simulator is used as the simulation tool.

In what follows, the experimental settings are presented. As shown in Table II, the area size of monitoring region is set at 2000 m × 300 m. The number of deployed sensors is ranging from 400 to 800 and the deployed WSN is connected. The initial energy of each sensor node is 200 000 units. The sensing rage is set at 60 m, while the communication range is set at 80 m. There are eight power levels which can be adjusted by each sensor. Each sensor offers data volume ranging from 100 to 300 units.

In addition, the term "round" refers to the time duration that the bus contacts with the WSN. Let t_{round} denote the length of each round. Let l_{round} denote the length of the road overlapping with the WSN. Recall that notation v denotes the average velocity that the bus periodically moves along the road.

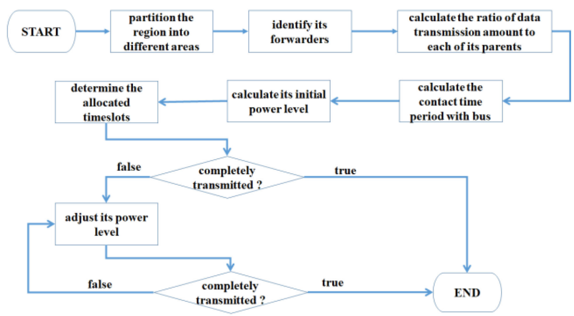
TABLE II
SIMULATION SETTINGS

Node deployment	Random
Given Region	2000 m * 300 m
The number of sensor nodes	400 – 800
Sensor node transmission Range	60 – 80 (m)
Sensing data volume	100 – 300 (units)
Transmission power levels	1-8
Data Rate	4 – 400 (units)
Energy	200000 (units)

The length of each round can be measured by

$$t_{\text{round}} = \frac{l_{\text{round}}}{v}.$$

Fig. 4 depicts the snapshot of throughputs by applying the three compared algorithms. The number of sensor nodes is 400. The experiment randomly selects six anchor nodes (marked with green color) from the deployed sensors and then observes their throughputs. As shown in Fig. 4, the proposed EBDCM has a higher throughput than MobiCluster and SEECH in all cases.



(a) The workflow of the proposed *EBMDC*.

Algorithm: ***ENERGY BALANCED MULTI-HOP DATA COLLECTION (EBMDC)***

Inputs: The unique ID, location, remaining energy and the buffer capacity of sensor $\boldsymbol{s_i}$.

Output: The transmission amount $\boldsymbol{D_{i,j}^{tra}}$, transmission power level $\boldsymbol{p_{i,j}}$ and scheduling results including $\boldsymbol{\lambda_{i,j}^{DT}}$, $\boldsymbol{\lambda_{i,j}^{LST}}$ and $\boldsymbol{\lambda_{i,j}^{RST}}$ of anchor $\boldsymbol{s_i}$ in slot $\boldsymbol{t_{i,j}}$.

If $isAnchor(s_i)$ {
/* Initialization Phase */
/* Constructs the minimum level spanning tree rooted by s_i; $l_i = 0$ */
/* BDC Determining Phase */
/* Evaluates the contacting time periods *DT*, *LST* and *RST* by exchanging messages with neighbors */
/* Evaluates its initial power level */
$A_i = \frac{\tilde{E}_i}{D_i^{tra}}$; /* average available remaining energy of s_i */
$A_i^{min} = Min(A_{i-1}, A_i, A_{i+1})$;
If $(A_{i-1} = A_i^{min})$ { $p_{i,j}^{init} = p_{i,j}^{threshold} * \frac{A_i}{A_{i-1}}$; }
Else If $(A_{i+1} = A_i^{min})$ { $p_{i,j}^{init} = p_{i,j}^{threshold} * \frac{A_i}{A_{i+1}}$; }
Else { $p_{i,j}^{init} = p_{i,j}^{threshold}$; }
/* Case: all data can be transmitted in DT */
$D_i^{tra} = \sum_{j=1}^{\tilde{\rho}_i^{DT}} \lambda_{i,j}^{DT} * r_{i,j}^{init}$; /* calculate the amount of data which can be transmitted in DT */
$\tilde{D}_i = D_i^{rec} + D_i^{gen} - D_i^{tra}$; /* the remaining data volume of s_i */
If $(\tilde{D}_i = 0)$ { return $\boldsymbol{\lambda_{i,j}^{DT}}$ and $p_{i,j}^{init}$; }
Else {
/* Occupies slots from *ST* */
/* Evaluates the average data rate r_i^{avg} and the average number of required slots ω_i */
$r_i^{avg} = \frac{\sum_{j=0}^{|T_i|} r_{i,j}^{init}}{|T_i|}$; /* average transmission rate of s_i */
$\omega_i = \frac{D_i^{tra}}{\gamma * r_i^{avg}}$; /* number of required time slots for transmitting D_i^{tra} */
/* Applying the initial power, evaluate the number of time slots required for transmission in *DT* and *ST* */
$\tilde{\rho}_i^{LST} = |T_i^{LST}| * \frac{\omega_i}{\omega_{i-1}+\omega_i}$; /* allocated slots of T_i^{LST} and T_i^{RST} */
$\tilde{\rho}_i^{RST} = |T_i^{RST}| * \frac{\omega_i}{\omega_i+\omega_{i+1}}$;
/* Case: all data can be transmitted in *DT* and *ST* */
$D_i^{tra} = \sum_{j=1}^{\tilde{\rho}_i^{DT}} \lambda_{i,j}^{DT} * r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LST}} \lambda_{i,j}^{LST} * r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RST}} \lambda_{i,j}^{RST} * r_{i,j}^{init}$;
$\tilde{D}_i = D_i^{rec} + D_i^{gen} - D_i^{tra}$;
If $(\tilde{D}_i = 0)$ { return $\lambda_{i,j}^{DT}, \lambda_{i,j}^{LST}, \lambda_{i,j}^{RST}$ and $p_{i,j}^{init}$; }
Else {
/* Case: DT and ST cannot be satisfied; Enlarges transmission power */
/* Evaluates the amount of increased data when increasing transmitting power by q levels */
$\delta_{D_i}^q = \sum_{j=1}^{\tilde{\rho}_i^{DT}} \lambda_{i,j}^{DT} * \delta_{i,j}^q + \sum_{j=1}^{\tilde{\rho}_i^{LST}} \lambda_{i,j}^{LST} * \delta_{i,j}^q + \sum_{j=1}^{\tilde{\rho}_i^{RST}} \lambda_{i,j}^{RST} * \delta_{i,j}^q$;
/* Computes the least increased power level u corresponding to p_i^{init} for transmitting all data */
$\delta_{p_i}^u = arg\ Min_q \left(\frac{D_i^{rec} + D_i^{gen} - D_i^{init}}{\delta_{D_i}^q} \le 1 \right)$;
/* Case: all data can be transmitted in *DT* and *ST* after enlarging transmission power*/
$D_i^{tra} = \sum_{j=1}^{\tilde{\rho}_k^{DT}} \lambda_{i,j}^{DT} * \delta_{r_{i,j}}^u + \sum_{j=1}^{\tilde{\rho}_k^{LST}} \lambda_{i,j}^{LST} * \delta_{r_{i,j}}^u + \sum_{j=1}^{\tilde{\rho}_k^{RST}} \lambda_{i,j}^{RST} * \delta_{r_{i,j}}^u$;
$\tilde{D}_i = D_i^{rec} + D_i^{gen} - D_i^{tra}$;
If $(\tilde{D}_i = 0)$ { return $\lambda_{i,j}^{DT}, \lambda_{i,j}^{LST}, \lambda_{i,j}^{RST}$ and $p_{i,j}^{init+u}$; }
} }
If $isNotAnchor(s_i)$ {
/* Initialization Phase */
/* Executes broadcasting and forwarder identification steps for constructing forwarding trees. */
For each $s_u \in N(s_i)$ {
If ((l_u is minimal) *and* ($s_u.anchor_{id} = s_i.anchor_{id}$)) {
add(P_i, s_u) ; /* add s_u into the parent set P_i of s_i */
$l_i = 1 + \min(l_u)$; /* setup the level of s_i */
} }
/* Bottom Up Requesting Phase */
/* Evaluates the ratio of data transmission amount to each of its parents. */
$D_{i,j}^{tra} = \frac{\tau_j}{\sum_{s_j \in P_i} \tau_j} \times D_i^{tra}$;
return $D_{i,j}^{tra}$; }

(b) The pseudocode of the proposed *EBMDC*.

Fig. 3. EBMDC algorithm. (a) Workflow of the proposed EBMDC. (b) Pseudocode of the proposed EBMDC.

This occurs because of several reasons. One of the reason is the smaller number of hops from each sensor to the anchor node since each sensor forwards its data to the closest anchor. The second reason is that anchor nodes applying the proposed EBDCM led ST slots to the neighboring anchors that have larger sizes of data. In addition, the anchor node can enlarge its power level to increase its transmission rate in case that the DT and ST are exhausted.

Fig. 5 compares the three algorithms in terms of throughputs. The number of sensors is varied from 400 to 800. Fig. 5 offers each sensor with 300 units of data volumes in each round. As shown in Fig. 5, the proposed EBMDC has better performance

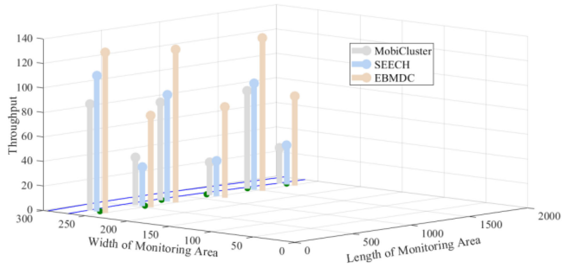


Fig. 4. Three algorithms are compared in terms of throughput by randomly selecting six anchor nodes.

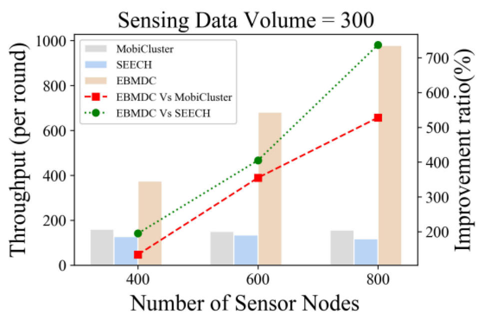


Fig. 5. Comparisons of three algorithms in terms of the throughput.

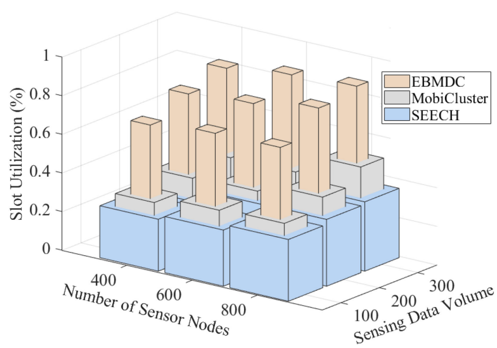


Fig. 6. Comparisons of three algorithms in terms of slot utilization.

than MobiCluster and SEECH in all cases. The EBMDC improves MobiCluster and SEECH ranging from 134% to 527% and from 195% to 736%, respectively. It occurs because of several reasons. First, the tree topology constructed by EBMDC has smaller levels because that each sensor forwards its data to the closer anchor. Second, the anchor node that has a larger amount of data can prior use the ST slots, which increases its throughput. Third, each anchor dynamically increases its transmission power if the allocated ST is not enough. The power level adjustment improves the throughputs because that the higher power level can enhance the transmission rate. Consequently, EBMDC outperforms MobiCluster and SEECH in terms of throughput.

Fig. 6 investigates the slot utilization of the three compared algorithms. The slot utilization is defined by the ratio of the number of used slots to the number of all allocated slots. The data volume offered to each sensor is varied from 100 to 300 units in each round. The number of sensors is varied from 400 to 800. In comparison, the proposed EBMDC outperforms MobiCluster and SEECH in all cases. The main reason is that the proposed EBMDC uses smaller power to transmit data in case the number

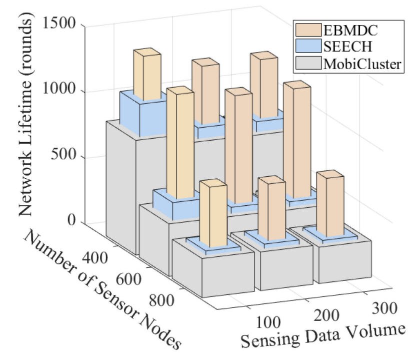


Fig. 7. Network lifetimes of the three algorithms are compared.

of DT slots is satisfied. This policy helps utilize the DT slots. In addition, the anchor with more data can use more ST slots, increasing the utilization of ST slots. In the ST slots, each sensor also first adopts a low power level, aiming to save its energy consumption and increase the slot utilization. Consequently, the EBMDC outperforms the other two algorithms in terms of slot utilization. The MobiCluster and SEECH do not schedule the shared slots, hence resulting in low slot utilization.

Fig. 7 compares the network lifetimes of EBMDC, MobiCluster, and SEECH. The network lifetime is measured from the time point that the network starts to the time point that the first node failures. The offered data volume is varied from 100 to 300 units. The number of sensors is varied from 400 to 800. The three algorithms have a similar trend that the network lifetime is increased when the number of sensors is increased or when the data volumes are decreased. The key reason is that more sensor nodes result in sharing the limited communication time. Therefore, each sensor can reserve a smaller number of time slots for data transmission, increasing its remaining energy. In comparison, the proposed EBMDC outperforms the compared MobiCluster and SEECH in all cases. The main reason is that the proposed EBMDC always adopts a proper power level. Furthermore, by applying the proposed EBMDC, those sensors which have higher remaining energy will increase their power. This can increase the transmission rate and thus reserve more ST slots for the neighbors with smaller remaining energy. Consequently, the neighbor with smaller remaining energy can adopt a lower power level to transmit all of its data.

In the proposed EBMDC, each sensor needs to exchange control messages with neighbors when constructing the multi-trees. In fact, the neighbors only need to exchange messages two times. For the first time, the control messages are broadcasted by anchors in a top-down manner. The control message is called *Tree_Construction* packet which includes Anchor Id, Anchor Location, Sender Id, Sender Location, Tree Level, Voronoi Edge Parameters, Amount of Buffered Data as well as the Remaining Energy. In addition, the anchors need to broadcast additional information, such as power level and the remaining resource of ST and DT. In the second time, the control messages are sent from leaf nodes in a bottom-up manner until the messages reach the anchor nodes. The control message is called *Forwarder_Identification* packet which includes Anchor Id and Anchor Location of the selected root. The control overheads are very small, as compared with the size of data transmitted to the bus. Table III further gives the impact of control overheads on the network lifetime. As shown in Table III, EBMDC represents the network lifetime which does not consider the control

TABLE III
PERFORMANCE BETWEEN EBMDC AND EBMDC-H IN TERMS OF NETWORK LIFETIME

Network Lifetime (rounds)	Data Volume	Number of Sensor Nodes		
		400	600	800
EBMDC	100	1445	1317	775
	200	1315	1254	745
	300	1311	1251	734
EBMDC-H	100	1444	1316	774
	200	1314	1252	743
	300	1309	1248	731

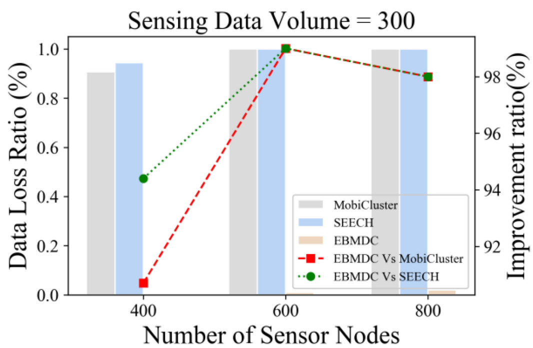


Fig. 8. Data loss ratios of three algorithms are compared.

overheads, while EBMDC-H represents the network lifetime which considers the control overheads.

Fig. 8 further compares the three algorithms in terms of data loss ratio. The number of sensors is varied from 400 to 800. Fig. 8 offers 300 units of data volumes in each round. A common trend of the three algorithms is that the data loss ratio is increased when the number of sensors is increased or the data volumes are increased. The main reason is that each sensor obtains fewer slots to transmit data in case that the number of sensors is increased. In comparison, the proposed EBMDC outperforms MobiCluster and SEECH in all cases. As shown in Fig. 8, the proposed EBMDC improves MobiCluster and SEECH ranging from 90% to 99% and from 91% to 99%, respectively. The main reasons are explained in the following. First, the EBMDC initially allocates proper numbers of ST to its neighbors according to *ST determining equations* in *BDC determining phase*. This allows the neighboring anchor to transmit more data in the allocated ST slots. Second, by applying the proposed EBMDC, the anchor enlarges its power level according to *power adjusting equations* in *BDC determining phase* when the allocated ST is still not enough. This can increase their throughputs accordingly. Consequently, EBMDC outperforms MobiCluster and SEECH in terms of data loss ratio.

Fig. 9 investigates the performance of the three algorithms in terms of fairness index. The number of sensors is varied from 400 to 800. Fig. 9 offers 300 units of data volumes in each round. The Jain's Fairness Index is applied as

$$\text{Fairness index} = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \times \sum_{i=1}^{n} {x_i}^2} \quad (17)$$

where x_i represents the amount of remaining data of s_i. Fig. 9 depicts that the proposed EBMDC outperforms the other two algorithms in all cases. The proposed EBMDC improves MobiCluster and SEECH ranging from 13% to 74% and from 10% to 68%, respectively. The main reason is that each anchor applying EBMDC allocates proper DT and ST slots to transmit data and hence a proper number of ST slots is reserved to its neighbors which have a larger amount of data. Another reason is that the anchor node with higher remaining energy further enlarges its power to reserve more ST slots to the neighbors which have lower remaining energy. Consequently, the sensor with lower remaining energy will have more ST slots and more rounds to transmit its data.

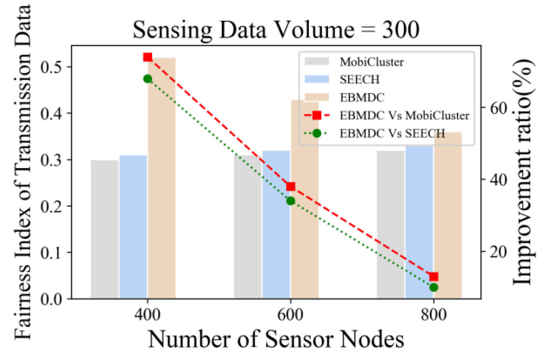


Fig. 9. Three algorithms are compared in terms of fairness index on data transmission.

VI. CONCLUSION

Data collection has received much attention in the past few years. This article proposes a distributed mechanism which considers bus as mobile sink and adopts Voronoi diagram, bottom-up requesting, slot scheduling, and power adjusting mechanisms, aiming to maximize the volume of data collected by bus while prolonging the network lifetime of WSNs. Applying the proposed EBMDC, each sensor connects to the closest anchor node for constructing multiparent spanning trees, aiming at balancing the remaining energy of sensor nodes. Each anchor node schedules its DT and ST slots by considering the remaining energy and the amount of data. As a result, the amount of data transmitted from anchors to the bus can be maximized while the network lifetimes of anchors can be balanced. Performance experiments showed that the proposed EBMDC achieves better results than the existing mechanisms in terms of throughput, network lifetime, slot utilization, data loss ratio as well as fairness index. Further studies will consider a heterogeneous sensor network which consists of different types of sensors with different sensing ranges. In addition, the time-constrained services and data aggregation of sensing data will be further considered.

REFERENCES

[1] W. M. Wen, S. H. Zhao, C. J. Shang, and C. Y. Chang, "EAPC: Energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sens. J.*, vol. 18, no. 2, pp. 890–901, Jan. 2018.
[2] R. Deng, S. He, and J. Chen, "An online algorithm for data collection by multiple sinks in wireless-sensor networks," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 93–104, Mar. 2018.
[3] X. He, S. Liu, G. Yang, and N. Xiong, "Achieving efficient data collection in heterogeneous sensing WSNs," *IEEE Access*, vol. 6, pp. 63187–63199, 2018.
[4] C. Sha, D. Song, R. Yang, H. Gao, and H. Huang, "A type of energy-balanced tree based data collection strategy for sensor network with mobile sink," *IEEE Access*, vol. 7, pp. 85226–85240, 2019.

[5] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, Mar. 2014.

[6] R. Velmani and B. Kaarthick, "An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks," *IEEE Sens. J.*, vol. 15, no. 4, pp. 2377–2390, Apr. 2015.

[7] F. Zhou, Z. Z. Chen, S. Guo, and J. Li, "Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs," *IEEE Sens. J.*, vol. 16, no. 22, pp. 8167–8177, Nov. 2016.

[8] Z. Sun, X. F. Xing, B. Song, Y. Nie, and H. Shao, "Mobile intelligent computing in Internet of Things: An optimized data gathering method based on compressive sensing," *IEEE Access*, vol. 7, pp. 66110–66122, 2019.

[9] H. Salarian, K. W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.

[10] C. Zhu, S. Wu, G. Han, L. Shu, and H. Wu, "A tree-cluster-based data-gathering algorithm for industrial WSNs with a mobile sink," *IEEE Access*, vol. 3, pp. 381–396, 2015.

[11] A. W. Khan, A. H. Abdullah, M. A. Razzaque, and J. I. Bangash, "VGDRA: A virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks," *IEEE Sens. J.*, vol. 15, no. 1, pp. 526–534, Jan. 2015.

[12] Y. C. Wang and K. C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1527–1540, Jul. 2019.

[13] K. L. M. Ang, J. K. P. Seng, and A. M. Zungeru, "Optimizing energy consumption for big data collection in large-scale wireless sensor networks with mobile collectors," *IEEE Sens. J.*, vol. 12, no. 1, pp. 616–626, Mar. 2018.

[14] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sens. Netw.*, vol. 8, no. 1, pp. 1–31, Aug. 2011.

[15] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 809–817, May 2012.

[16] A. Ridha, "SEECH: Scalable energy efficient clustering hierarchy protocol in wireless sensor networks," *IEEE Sens. J.*, vol. 14, no. 11, pp. 3944–3954, Nov. 2014.

[17] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 395–410, Apr. 2007.

[18] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 127–139, Jan. 2010.

[19] L. He, J. Pan, and J. Xu, "A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1308–1320, Jul. 2013.

[20] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 265–277, Feb. 2012.

[21] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proc. IEEE Local Comput. Netw. Conf.*, pp. 582–589, Mar. 2010.

[22] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1870–1883, Jul. 2015.

[23] C. Wu, Y. Liu, F. Wu, W. Fan, and B. Tang, "Graph-based data gathering scheme in WSNs with a mobility-constrained mobile sink," *IEEE Access*, vol. 5, pp. 19463–19477, 2017.

[24] K. Shah, M. D. Francesco, G. Anastasi, and M. Kumar, "A framework for resource-aware data accumulation in sparse wireless sensor networks," *Comput. Commun.*, vol. 34, no. 17, pp. 2094–2103, Nov. 2011.

[25] A. Chakrabarti, A. Sabharwal, and B. Aazhang. "Communication power optimization in a sensor network with a path-constrained mobile observer," *ACM Trans. Sens. Netw.*, vol. 2, no. 3, pp. 297–324, Aug. 2006.

[26] C. Qiu, H. Shen, and K. Chen, "An energy-efficient and distributed cooperation mechanism for K-coverage hole detection and healing in WSNs," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1247–1259, Jun. 2018.

[27] M. Bakshi, B. Jaumard, and L. Narayanan, "Optimum converge cast scheduling in wireless sensor networks," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5650–5661, Nov. 2018.

Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from the National Central University, Taoyuan City, Taiwan, in 1995.

He is currently a Distinguish Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include Internet of Things, wireless sensor networks, artifical Intelligence, and deep learning.

Dr. Chang was an Associate Guest Editor for several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing* from 2011 to 2020, the *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, the *Journal of Information Science and Engineering* in 2008, and the *Journal of Internet Technology* from 2004 to 2008.

Chung-Chih Lin received the B.S. and M.S. degrees in computer science and information engineering from the National Central University, Taoyuan City, Taiwan, in 1998 and 2002, respectively, and the Ph.D. degree in computer science and information engineering from Tamkang University, New Taipei, Taiwan, in 2020.

His current research interests include wireless sensor network, Internet of Things, and machine learning.

Wen-Hwa Liao (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 2002.

He is currently a Full Professor with the Institute of Information and Decision Sciences, National Taipei University of Business, Taipei City, Taiwan. His current research interests include Internet of Things, wireless sensor networks, and attificial intelligence.

Dr. Liao was an Associate Editor for the *International Journal of Distributed Sensor Networks and International Journal of Vehicle Information and Communication Systems.* He was also as an Associate Guest Editor for SCI-indexed *International Journal of Ad Hoc and Ubiquitous Computing.*

Diptendu Sinha Roy received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010.

In 2016, he joined as an Associate Professor the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, Shillong, India, where he has been the Chair since January 2017. Prior to his stint at NIT Meghalaya, he was with the Department of Computer science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing, and Internet of Things, specifically applications of artificial intelligence/ machine learning for smart integrated systems.