Ad Hoc Networks 71 (2018) 88–103

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

LEEF: Latency and energy efficient federation of disjoint wireless sensor segments

Sookyoung Lee[a], Mohamed Younis[b], Ben Anglin[b], Meejeong Lee[a,*]

[a] Department of Computer Science and Engineering, Ewha Womans University, Seoul, Republic of Korea
[b] Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, USA

ARTICLE INFO

Article history:
Received 24 April 2017
Revised 13 November 2017
Accepted 28 December 2017
Available online 3 January 2018

Keywords:
Wireless sensor networks
Network partitioning
Topology repair
Federation
Mobile data carrier
Intermittent connectivity

ABSTRACT

In hostile environments where explosives and natural calamities probably occur, wireless sensor networks (WSNs) are susceptible to multiple collocated failures and could be partitioned into disjoint segments. Federating the segments would be essential for restoring connectivity and enabling data sharing in the network. The federation may be achieved by populating relay nodes and providing perpetual inter-segment paths. In this paper, we tackle the federation problem while considering constrained relay availability, i.e., a limited number (k) of mobile relays to provide intermittent inter-segment connectivity that makes the problem more challenging. We propose LEEF, a novel algorithm for achieving energy-efficient federation with low inter-segment data delivery latency. LEEF strives to group the segments into k clusters in a star topology where a cluster at the center of the area serves as a hub between each pair of segment-clusters. Each cluster is served by a distinct mobile relay. In addition, LEEF opts to equalize the energy consumed by the k mobile relays due to travel and wireless communication. We analyze the properties of LEEF mathematically and validate its performance through extensive simulation experiments.

1. Introduction

Recently WSNs have attracted increased interest due to their numerous civil, scientific and military applications. In WSNs, a large set of sensors are deployed to form a mesh topology and coordinate their actions to carry out a common task [1]. Thus the inter-sensor connectivity has a significant influence on the effectiveness of WSNs and should be sustained all the time. Moreover, a WSN often operates in harsh environments and may suffer from major damage which results in the simultaneous failure of multiple collocated nodes, and causes the network to become partitioned into disjoint segments. For example, in a battlefield, some nodes in the deployment area may be attacked by explosives and get destroyed. Restoring the connectivity among segments is essential for enabling full network operation. Another scenario is when autonomous networks are to be federated to aggregate their capabilities and accomplish a common task such as search-and-rescue, military situation awareness, criminal hunting, etc.

Federating a set of disjoint segments or standalone networks has recently received growing attention from the research community. Most published solutions exploit the deployment of stationary relay nodes and formulate the federation problem as finding the locations of the least relay count to form a stable inter-segment topology [2]. In other words, a connected topology is formed by populating sufficient relays to provide stable inter-segment data paths. However, resource scarcity makes the federation problem more challenging. In this paper, we consider the situation where the relay count is not sufficient to form a perpetual topology. Instead, multiple mobile data carriers (MDCs) are employed to form intermittent communication links among the segments. Particularly, we solve the federation problem when the MDC count is so constrained that it is not feasible to assign an MDC to each link in a minimum spanning tree of the segments.

Given the MDC availability constraint, each MDC has to serve more than two segments. We thus ought to divide the set of the segments S_T into multiple groups or clusters. This is mapped into a set cover problem which is NP-hard. Therefore, we pursue heuristics. We consider two objectives while clustering S_T: (i) *minimizing the maximum inter-segment communication delay* incurred due to the lack of stable connectivity, i.e., use of the intermittent links, which will be affected by the intra-cluster topology, i.e., a tour path of each MDC in a cluster and the inter-MDC communication; (ii) *prolonging the lifetime of the formed network* by balancing the energy consumed by the MDCs for touring in each cluster and uploading/downloading data over wireless links. The second ob-

* Corresponding author.
E-mail addresses: sookyounglee@ewha.ac.kr (S. Lee), younis@umbc.edu (M. Younis), jama1@umbc.edu (B. Anglin), lmj@ewha.ac.kr (M. Lee).

https://doi.org/10.1016/j.adhoc.2017.12.008
1570-8705/

jective incurs higher computation complexity in forming the MDC tours; yet, it is essential in order to extend the lifespan of MDCs in serving the federated network and to avoid premature depletion of the energy supply of any of them. We propose a novel algorithm for Latency and Energy Efficient Federation (LEEF) by forming a star inter-cluster topology where the energy consumption overhead in touring and data transporting is balanced among the *k* MDCs. LEEF consists of two phases.

In the 1st phase, LEEF reviews the layout of segments with respect to the MDC communication range, *R*, by modeling the damaged area as a grid based on square-cells whose side is $R/\sqrt{2}$ in length. Then, LEEF tries to find a set, CS_T, of the fewest cells which if visited, the MDCs will reach all segments. Modeling the geographical area as a cell-based grid results in a revised layout which is different depending on a value of *R*. LEEF focuses on federating cells in CS_T without considering *R*. In the next step, LEEF groups the cells of CS_T into *k* virtual clusters (VCs) with VC_k serving as a *hub-virtual cluster* that is formed by placing an MDC at the center of the damaged area, *G* around which (*k*-1) virtual clusters are found. The resulting topology of VCs is thus star-shaped. During the 2nd phase, LEEF leverages the VCs to form *k* energy balanced clusters through two steps. The first step operates in rounds. Each C_i is initially formed as C_k=VC_k, C_i={G, the closest cell to $G \in VC_i$}. Then, in each round the cluster, C_{least}, whose MDC consumes the least energy is expanded by adding a new cell from CS_T and updating the other clusters. This step continues until every cell in CS_T is assigned to a cluster. In the second step, LEEF optimizes the formed clusters by readjusting the least and most energy consumed clusters to further balance the load on the MDCs. LEEF is validated through simulation experiments and is shown to outperform competing schemes.

The rest of the paper is organized as follows. LEEF is compared to related work in Section 2. In Section 3, the problem is formally defined and the considered system model is described. The details of LEEF are provided in Section 4. The validation results are presented in Section 5. The paper is finally concluded in Section 6.

2. Related work

Published techniques for federating the distinct segments or autonomous WSNs can be categorized into two groups [2]: (i) Approaches that employ relay nodes and form a stable data path among segments. The main objective of these approaches is usually to minimize the number of populated relays to achieve full connectivity; (ii) approaches that exploit mobility for solving the federation problem. These mobility-assisted techniques opt to create intermittent links. This section discusses related work in these two categories. It is worth noting that some work has addressed network partitioning caused by the loss of a critical node by either reconfiguring the network topology to tolerate the failure [2], or adjusting the data routes to prevent critical nodes from exhausting their onboard energy supply [3]. Given the scope of the contribution, we focus on scenarios in which the partitions are multi-hop apart and cannot be interconnected with just one node.

2.1. Relay node placement

In general, deterministic placement of relay nodes that are richer in computation, energy and communication resources than sensor nodes, has been pursued as a means for shaping a WSN topology in order to satisfy some desired performance goals. While minimizing the required relay count has been the prime optimization objective, many variants of relay placement have been proposed to achieve additional goals, e.g., degree of connectivity, path length, etc. Achieving connectivity with the least relay count is equivalent to identifying a Steiner minimum tree with minimal Steiner points and bounded edge-length which is shown to be NP-hard by Lin and Xue [4]. Therefore, heuristics have been pursued. Published heuristics can be categorized into three groups.

In the first group, unconstrained setups are considered where the nodes can be virtually placed anywhere in the area of interest. The placement algorithms in this group strive to minimize the relay count needed for establishing connectivity between end-terminals [5–7], or for restoring connectivity in a partitioned network [8–10]. In addition to reducing the number of relays, the second group considers more objectives, such as high energy efficiency to prolong network lifetime [11–13], or high degree of connectivity by providing multiple disjoint inter-node paths in the formed network [14,15]. Solutions in the third group tackle the federation problem under node position constraints [16,17]. Overall the approaches in this category provide perpetual connectivity and do not consider resource-availability constraints. LEEF solves the federation problem when the relay count is insufficient for forming stable data paths, and establishes intermittent connectivity instead.

2.2. Mobility-assisted solutions

Mobile agents have been employed to transport data in sparse networks by playing a role of a base-station or a data carrier. A mobile base-station moves in the area of interest and gathers data from sensor nodes over multi-hop paths. Meanwhile, a data carrier tours sensors and downloads their readings in one hop to relay or carry them to a sink or another node in the network. Therefore, a MDC eventually forms an intermittently connected topology. Published MDC-based federation schemes fall into one of two categories depending on whether more objectives are considered in addition to establishing connectivity.

The approaches in the first category do not consider a constraint on the availability of the mobile nodes [18–20]. For instance, in [18] Almasaeid and Kamal model the mobile-assisted federation problem as a closed queuing network and focus on the effect of the network parameters related to mobile relays on the end-to-end delay. They expanded their studies in [19] by including two other roles of mobile agents, i.e., mobile sinks and mobile collectors. Li and Hua [20] deploy mobile nodes to form a second tier mesh network to transfer data from sensors to a sink such that the delay due to buffer space limitation is minimized. Unlike LEEF the approaches introduced in this group do not consider the limited supply of mobile nodes and thus careful assignment of MDCs is not studied. Some studies, e.g., [21], have focuses a single MDC setup and how to limit the length of data paths. A survey can be found in [22]. However, such work does not deal with segments of nodes and address inter-segments connectivity.

Like LEEF, the second category solves the federation problem with a limited MDC count. IDM-kMDC [23], FeSMoR [24], and MINDS [25] first find a stable inter-segment topology based on which MDC tours are formed. Basically, an MDC may be dedicated to tour the link between a pair of segments or tour a subset of the segments. In addition, MINDS strives to even tour lengths among MDCs and is compared to LEEF in Section 5. Moreover, MiMSI and RCR strive to solve a more constrained version of the problem where only few relays are movable and the others are stationary [26,27]. However, unlike LEEF these approaches focus more on tour lengths rather than energy consumed by the MDCs and do not factor in the inter-segment communication latency.

On the other hand, there are some approaches that focus on reducing data delivery latency. Senturk and Akkaya [28] propose a delay-aware clustering algorithm which assigns segments to *k* MDCs to reduce the maximum tour length among MDCs and data delivery delay between segments to a sink. In addition, FOCUS [29] factors in the data delivery delay in the federation by scheduling the motion of MDCs. It first forms overlapped *k* clusters served by k MDCs and then tries to synchronize the arrival of two MDCs by carefully setting their motion speed in order to avoid storing data and holding MDCs at the intersection segment of two overlapping clusters. However, FOCUS does not care for energy concerns. Meanwhile, TOCS [30] opts to reduce the average and maximum delay for delivering data between segments by finding the balanced tour paths among MDCs. Like LEEF, TOCS groups a set of segments into k clusters that form a star topology in order to reduce the inter-segment latency. Then, it strives to equalize the MDC tours by adjusting the size of the cluster at the center. However, TOCS does not consider balancing the energy consumption among MDCs during operation. Unlike TOCS, LEEF strives to even energy required for each MDC to tour and transport distinct volume of data between the terminals which it serves. The consumed energy and the communication delay of LEEF are compared to those of MINDS, TOCS and FOCUS in Section 5.

3. System model and problem formulation

The mobility-assisted federation problem tackled in this paper may arise in two scenarios; (i) restoring lost inter-segment connectivity after a major node damage, e.g., inflicted by explosives in a battlefield or natural calamities such as landslides or avalanches, and (ii) linking individual batches of sensor nodes, for example to enable collaboration among multiple standalone WSNs in order to achieve a common mission such as search-and-rescue, or environment/creature monitoring. In both scenarios, meeting some inter-segment requirements, such as link capacity, may be necessary to achieve the mission; in this paper, we consider the data volume, represented as bits per data collection round, to be transported between every pair of segments. Such an inter-segment requirement can be: (i) just a byproduct of the damage depending on the size of the individual segments, or (ii) required to provide the application service though the federated segments.

Segments are assumed to include only stationary nodes and topology adjustment through cascaded relation is not feasible. Instead, the federation is to be achieved using k externally-supplied MDCs that provide intermittent data paths among the segments subject to the data volume requirement. The main optimization objective for such federation is to *determine the MDC tours that reduce inter-segment communication delay and balance the energy overhead experienced by the MDCs while travelling and transporting data*. In other words, we employ a set M of k MDCs $\{M_1, M_2, \ldots, M_k\}$ to connect a set S_T of n distinct terminals $\{S_1, S_2, \ldots S_n\}$, $n > 2$; a terminal plays the role of a gateway node that serves as an interface for the segment. In the rest of the paper the terms terminal and segment are used interchangeably. In LEEF, S_T is to be grouped into clusters C_i, i=1,.., k, i.e., $C_i \subset S_T$ and $\cup_{\forall i} C_i = S_T$. Each cluster C_i is served by M_i along a travel route $TR(M_i)$ that contains a list of coordinates at which M_i stops to upload or download data from/to the terminals in C_i. The delay incurred while delivering the stored data from segment S_s to S_d denoted as $D(S_s, S_d)$ and the energy consumed by M_i, $E(M_i)$, could be calculated by (1) and (2), respectively.

$$D(S_s,\ S_d) = D_{Trip}(Trip_{s,d}) + \sigma D_{Comm}(Data_{s,d}) + D_{Relay}(ht_{s,d}) = \left(\frac{1}{Speed_{MDC}} \cdot Trip_{s,d}\right) + \left(\frac{\sigma}{BW_{MDC}} \cdot Data_{s,d}\right) + (T_{hold\,s,d}) \tag{1}$$

$$E(M_i) = E_M(TR(M_i)) + E_C(Data_{M_i}) = (\mu \cdot TR(M_i)) + (Data_{M_i} \cdot P_C(R)) \tag{2}$$

In (1), the delay incurred while carrying data from S_s to S_d is primarily determined by a trip latency D_{Trip}, data transmission latency D_{Comm}, and relaying latency D_{Relay}. D_{Trip} mainly depends on the length of the travel path between two terminals S_s and S_d, denoted as $Trip_{s,d}$ and the speed of the involved MDCs. We assume a constant speed $Speed_{MDC}$ for all MDCs and D_{Trip} is thus a function of only the tour length of the individual MDC. The speed of an MDC would depend on the sensing rate and/or a schedule of data collection. In addition, D_{Comm} is determined by the volume of data transfer between S_s and S_d, and the capability of the radio transceiver (bits per second), represented as $Data_{s,d}$ and BW_{MDC} in (1), respectively. Recall that LEEF forms a star-like inter-cluster topology; thus inter-cluster relaying may take place. The constant σ captures that fact that the communication delay be incurred more than once since other segments will be involved in buffering the data. The value of σ is set to 1 if S_s and S_d are in the same cluster, to 2 if either S_s or S_d belong to the center cluster, and to 3 if S_s and S_d are part of two distinct outer clusters. The data volume transported between every pair of terminals is assumed to vary, e.g., based on the size of the corresponding segments or on the application served by the federated network.

Lastly, D_{Relay} means the buffering time for inter-cluster data to be relayed from an MDC to another if S_s and S_d do not belong to the same cluster. In LEEF, the MDCs do not need to meet directly to exchange data and the inter-cluster data relaying is supported via buffering data at commonly visited terminals by the MDCs depending on their own trip schedule. For example, a segment S_k could serve as an inter-cluster gateway where an MDC M_i uploads data at S_k to be downloaded at a later time by another MDC M_j in order to be delivered to a segment in a different cluster.. Therefore, the holding time at S_k (T_{hold}) for the data uploaded by M_i until M_j arrives at S_k to pick up is included as D_{Rrelay}. This may be encountered once or twice depending on which clusters S_s and S_d belong to, as explained above when discussing σ.

Meanwhile, the consumed energy by M_i in serving C_i, i.e., $E(M_i)$, includes the energy for motion $E_M(M_i)$, and the ancillary energy for communication $E_C(M_i)$. In detail, $E_M(M_i)$ represents the consumed energy during M_i motion while it tours the segments of C_i once. In addition, $E_C(M_i)$ includes the required energy for M_i to up/download two types of data traffic: the intra-cluster data in C_i and inter-cluster data, i.e., data imported from or exported to another cluster C_j. Therefore, E_M predominantly scales with the tour length, i.e., $TR(M_i)$ while E_C is dependent on the volume of data, $Data_{M_i}$, uploaded/downloaded over a wireless link, i.e.,

$$Data_{M_i} = \sum_{\forall S_s, S_d \in C_i} Data(S_s, S_d) + \sum_{S_x \in C_i, S_y \in C_j \ \forall j \neq i,} Data(S_x, S_y)$$

We assume using an energy cost model for E_M that is proportional to the travel distance, as seen in (3), while E_C is primarily proportional to the power for transmitting a single bit over a wireless link seen in (4).

$$E_M(\ M_i) = \mu \cdot TR(M_i) \tag{3}$$

$$E_C(\ M_i) = Data_{M_i} \cdot P_C(R),\ \ where\ P_C(R) = \alpha + \beta \cdot R^{\partial} \tag{4}$$

In (3), μ ranges from 0.1 to 1 J/m [31]. In addition, the energy required to transmit 1 bit is $2 \cdot 10^{-6}$Joule, where $\alpha = 100$nJ, $\beta = 0.1$nJ/m^{∂}, $\partial = 2$ in (4) [32]. During the federation, LEEF strives to balance $E(M_i)$,$\forall i$ by factoring in the motion and communication related energy overhead, i.e., $E_M(M_i)$ and $E_C(M_i)$, respectively. Overall, the problem that we tackle in this paper is captured mathematically by the following formula:

$$Find\ a\ set\ of\ C_i,\ \ i = 1,\ 2, \ldots k, \tag{5}$$

$$where \cup_{\forall i} \boldsymbol{C_i} = S_T\ such\ that\ max_{\forall S_p, S_q \in S_T} D(S_p, S_q)$$
$$and \left(\ \sum_{i \neq j}^{\forall C_i, C_j} |E(M_i) - E(M_j)|\right) are\ minimized$$

The problem presented in formula (5) is to find a set of clusters that covers S_T, each cluster includes a set of terminals that an MDC M_i visits such that the inter-terminal data delivery delay is reduced and energy overhead for the individual MDCs is balanced. In addition, we assume that all MDCs have the same capabilities with no data storage constraint. Without considering any additional objectives, the formulated problem can be mapped into solving a k-means clustering problem that is known NP-hard [4], and thus LEEF pursues heuristics.

A sensor does not need to have a global map of the area. The network is modeled at the level of segments rather than the level of sensors. Discovering the segments can be done by robots or UAVs. Once the segment locations are known, the area is mapped into a grid and the segment cells are identified, as explained in details in the next section. LEEF could be executed at a centralized command center or by one of the MDCs and then the tours are communicated to the individual MDCs. An MDC is assumed to have sufficient buffer space for the data transported in one tour. Additionally, the paper focuses on the algorithmic aspect of the internetworking problem without considering diversity of the physical, link and network layers. It is also assumed that all MDCs have the same communication range R which is equal to that of a sensor. This is a simplifying assumption to ease the presentation. In addition, coverage is not the focus of LEEF although MDCs may have sensing capabilities that can be leveraged during their tours in order to mitigate coverage loss caused by the damage.

4. The LEEF approach

To provide delay conscious and energy balanced federation of the n terminals using k MDCs, LEEF groups the terminals into a set of k clusters in a star inter-cluster topology where a *hub-cluster* C_k facilitates data relaying between pairs of clusters. In the first phase, LEEF models the area of interest as a grid of equal-size square-shaped cells, based on which terminal clustering is performed. Then a set CS_T of the fewest cells that cover all terminals in S_T is identified by evaluating each cell's *reachability* to terminals, and *proximity* to the center of the area G. LEEF groups the cells in CS_T into k VCs considering inter-cell proximity. In the second phase, LEEF opts to use the VCs to guide the formation of energy balanced k-clusters in a greedy manner through a two-steps process: *greedy-expansion* and *optimization*. During the *greedy-expansion* step, LEEF operates in rounds. Starting with $C_k = VC_k$ and $C_i = \{G$, the closest cell to $G \in VC_i\}$, in each round r energy overhead of C_i, denoted as E_i^r, is computed based on the intra- and inter-cluster data up/download and MDC tours between the involved segments in C_i up to round r. Then, the least energy consumed cluster C_{least} is selected for expansion through adding a segment. In the *optimization* step, balancing energy between clusters is performed by adjusting cluster membership to enable C_{least} to grow. The details of LEEF are provided in the balance of this section.

4.1. 1st phase: Modeling the damaged area and forming virtual clusters

LEEF first analyzes the damaged network topology by modeling the area as a grid and then forms a virtual *hub-cluster* VC_k around which (k-1) virtual clusters are established. The details of the first phase are provided below.

4.1.1. Modeling the damaged area into a grid

Prior to performing segment grouping into clusters, LEEF reviews the layout of terminals with the respect to R. In other words, it tries to find a geographical location at which two or more terminals can be simultaneously reached by an MDC and thus opts to simplify the segments to MDC assignment. This analysis will result in a revised layout that differs depending on a value of R. To do so, LEEF models the damaged area as a grid whose cell size is determined based on R. For the working example seen in Fig. 1(a) and (b) shows the modeled area. The side of each cell c_i is $R/\sqrt{2}$ in length and a non-boundary cell has thus eight neighboring cells each of which is reachable from an MDC that would stay at the center of c_i. In Fig. 1(b), an 1700 × 1100 m^2 area is modeled into (17 × 11) cells; each is 100×100 m^2 assuming R is about 140 m. In addition, the numbers shown in each cell c_i represent a list $L(c_i)$ of the IDs of the segments whose terminals are reachable from cell c_i. For example in Fig. 1(b), an MDC located at cell (6, 5) can reach the terminals S_5 and S_{18}.

Based on $L(c_i)$ $\forall i$, LEEF simply groups segments using three attributes: $access(c_i)$, $1hop(c_i)$, and $prox(c_i)$. $access(c_i)$ is the number of accessible terminals from c_i, i.e., $|L(c_i)|$, and $1hop(c_i)$ equals the number of terminals that are reachable from c_i via one of the neighboring cells, i.e., $1\mathrm{hop}(c_i) = |\cup_{c_j \in NBR(c_i)} L(c_j)|$, where $NBR(c_i)$ is the set of neighboring cells of c_i, with $|NBR(c_i)| \leq 8$. Meanwhile, $prox(c_i)$ represents *proximity* to the center of the area G from c_i which is calculated as a cell count for the shortest path between G and c_i. The cell count between a pair of cells, $c_i(x_i,y_i)$ and $c_j(x_j,y_j)$, hereafter denoted as $CellDist(c_i,c_j)$, is computed as $max(|x_i - x_j|, |y_i - y_j|)$. Thus, $prox(c_i)$ equals $max(|x_i - x_g|, |y_i - y_g|)$, where G is located at $c_g(xg,y_g)$.

Using $access(c_i)$, $1hop(c_i)$, and $prox(c_i)$ $\forall i$, LEEF tries to identify the smallest subset CS_T of cells which covers all terminals in S_T, i.e., $CS_T = \cup_{i=1}^{n}\{c_i\}$, where $min_n(\cup_{i=1}^{n}\{S_a |\ S_a \in L(c_i)\} = S_T)$. Since the optimization objective of LEEF is to reduce inter-terminal data transmission latency and to balance energy consumed for MDCs in travelling and data transporting between terminals, *a central cluster is formed around G* and *the selected cells in CS_T ought to be located inward towards G* so that an MDC can collect the data on its path to the central cluster. In other words, the identification of CS_T can be mapped into the set cover problem while considering $access()$, $1hop()$, and $prox()$ as criteria. Thus, preference is given according to the following sequence:

(a) c_i which can reach more terminals within a distance R, i.e., the cell with more $access(c_i)$;
(b) c_i that can be reachable to more terminals in 2R, i.e., the one with larger $1hop(c_i)$. By favoring a higher value of $1hop(c_i)$, we can increase the chance for finding a reduced travel path along which an MDC can cover (more) segments;
(c) c_i which is closer to G is picked, i.e., the one that has smaller $prox(c_i)$ is preferred given the proximity to the central cluster. Basically, more inward cells will increase the probability of being on the shortest travel path of an MDC.

In the example seen in Fig. 1(b), $c(4,1)$, $c(3,3)$, and $c(9,14)$ illustrate the three cases, where the selection criterion is $access()$, $1hop()$, and $prox()$, respectively. Through the formation of CS_T, LEEF simplifies the federation problem of 18 terminals while factoring R into the problem of providing the chosen 14 cells with intermittent links. Fig. 1(b) shows the selected cells in CS_T marked in shade. For instance, $L(c(4,1))$ and $L(c(5,11))$ are $\{S_8, S_9\}$ and $\{S_{13}\}$, respectively.

4.1.2. Forming virtual clusters based on inter-cell proximity with respect to g

For federating the terminals of S_T with k MDCs in a low delay topology, LEEF groups the cells in CS_T into clusters and forms an inter-cluster topology, where each cluster is served by a distinct MDC and every pair of cells can be reachable to each other within at most two cluster hops. Thus, the maximum inter-terminal data delivery delay between any two segments in different clusters will be determined by three tours of a central cluster, and two clusters

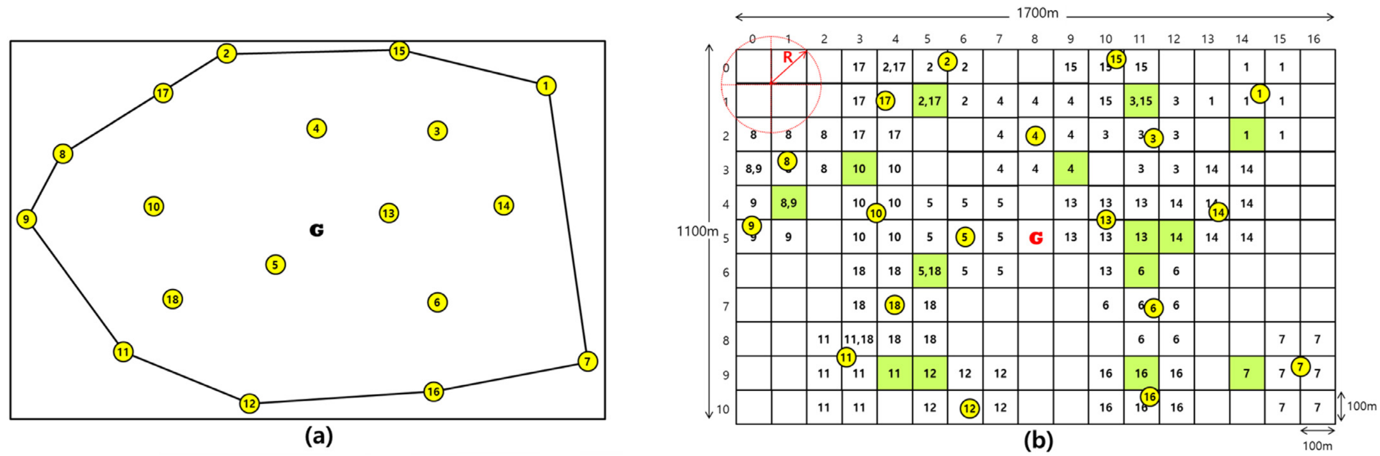

Fig. 1. For simplification of a layout of terminals in (a), LEEF first models the damaged area into the cell-based grid based on R and analyzes each cell. The number in a cell seen in (b) represents terminals covered by an MDC that would be placed at a center of the cell.

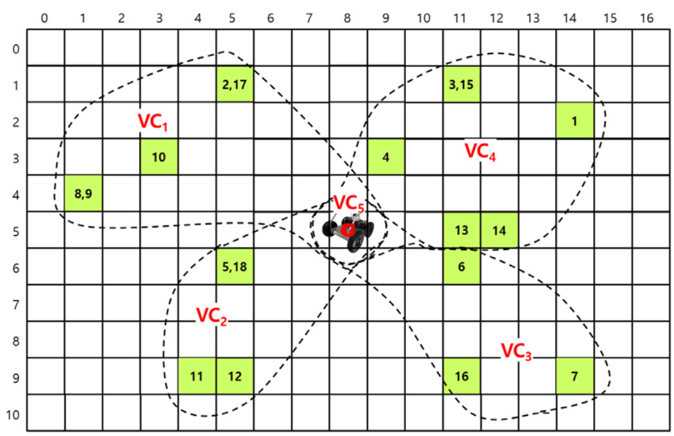

Fig. 2. In the 1st phase, LEEF forms k virtual clusters (VCs), where VC_k is formed by placing an MDC, M_k at G around that (k-1) VCs are formed, each of which includes M_k and minimizes a mean of inter-cell proximity.

which include segments furthest apart from each other. Maintaining low delay data delivery service between cells, LEEF additionally strives to equalize the energy overhead experienced by the individual MDCs while touring and transporting data. Since the data volume exchanged between terminals differs, the energy consumed in transferring data over wireless links is unbalanced even in the case that every cluster includes the same number of terminals.

For identifying the proximity-based VCs, LEEF first forms a *virtual hub-cluster* VC_k at G by virtually placing an MDC, M_k, around which ($k-1$) virtual clusters are formed. The non-central virtual clusters are formed using the approach of [30], which initially forms a distinct cluster for each segment that also contains G, and then clusters are merged in a greedy manner to reduce the tour length. In the context of LEEF, initially each $c_i \in CS_T$ becomes a VC, and the two neighboring VCs with the least combined tour cost along with G among all VC pairs are merged. The travel length of each non-central VC is thus minimized. This merge process is repeated until only ($k-1$) VCs are left. Then each VC is assigned an ID number from 1 to ($k-1$) in a counterclockwise direction, and every cell $c \in VC_i$, for $i < k$ is tagged with such ID, denoted hereafter as $VCID(c)$. Fig. 2 shows that 14 cells $\in CS_T$ are grouped into 5 VCs where a *virtual hub-cluster* VC_5 contains only an MDC with no cell. For each cell $c \in VC_i$, $VCID(c)=i$. The VC_i $\forall i$ are referenced in the 2nd phase which forms k of energy balanced clusters C_i $\forall i$ in terms of data volume exchanged between segments.

4.2. 2nd phase: k-clustering while balancing energy of MDCs

Based on the VCs of the first phase, the major objective of the second phase is to form k clusters C_i $i=1,..., k$, each is served by an MDC M_i, such that the energy overhead experienced by M_i is balanced. Basically virtual clusters are determined while considering the MDC tours without factoring in the data communication overhead; the second phase of LEEF forms C_i $\forall i$ based on the layout of virtual clusters while changing the cells in the individual VCs to equalize the MDC load incurred in both motion and data communication. Since LEEF opts to form a star cluster topology, VC_k initially becomes C_k, i.e., by keeping M_k placed at G to serve as a stationary relay node. Then depending on the ratio of the energy consumed in all data up/downloading to that of the total MDC tours, C_k may be kept at G, moved to another position by re-locating M_k, or expanded by including two or more cells $\in CS_T$ between which M_k travels. The 2nd phase consists of *greedy-expansion* and *optimization* steps as detailed later in this section. Both steps operate in rounds opting to balance the energy overhead experienced by MDCs in data communication and touring. We first provide the following notation first.

- ***VCID***(***c***): Represents an identification of the virtual cluster to which a cell $c \in CS_T$ belongs.
- ***ID***(***c***): Is the identification of the cluster to which a cell $c \in CS_T$ currently belongs. The initial value of $ID(c)$ $\forall c$ is set to *not_clustered*.
- ***Anchor***(C_i): To form a star inter-cluster topology, each M_i i=1,…,($k-1$) ought to visit a *hub-cluster* C_k by either directly communicating with M_k if it acts as a stationary relay or downloading the data at a terminal in C_k to be buffered until the arrival of M_k. We call the position where M_i rendezvous with C_k, *Anchor*(C_i), which may be one of the neighboring cells to that where a stationary M_k is placed at or one of the cells $c \in C_k$, whose $ID(c)$=k. Overall, for each cluster C_i LEEF keeps only one *Anchor*(C_i) which is the closest to any cell $\in C_i$ in order to shorten the travel path of M_i. Therefore, each time C_k is modified during the 2nd phase, non-hub clusters ought to be adjusted by calling the function *Adjust_Clusters_to_Hub*(), which checks and changes *Anchor*(C_i) if needed.
- ***TR***($\boldsymbol{M_i}$): The shortest path that M_i takes while visiting the terminals in C_i is denoted as $TR(M_i)$. It is equivalent to the short-

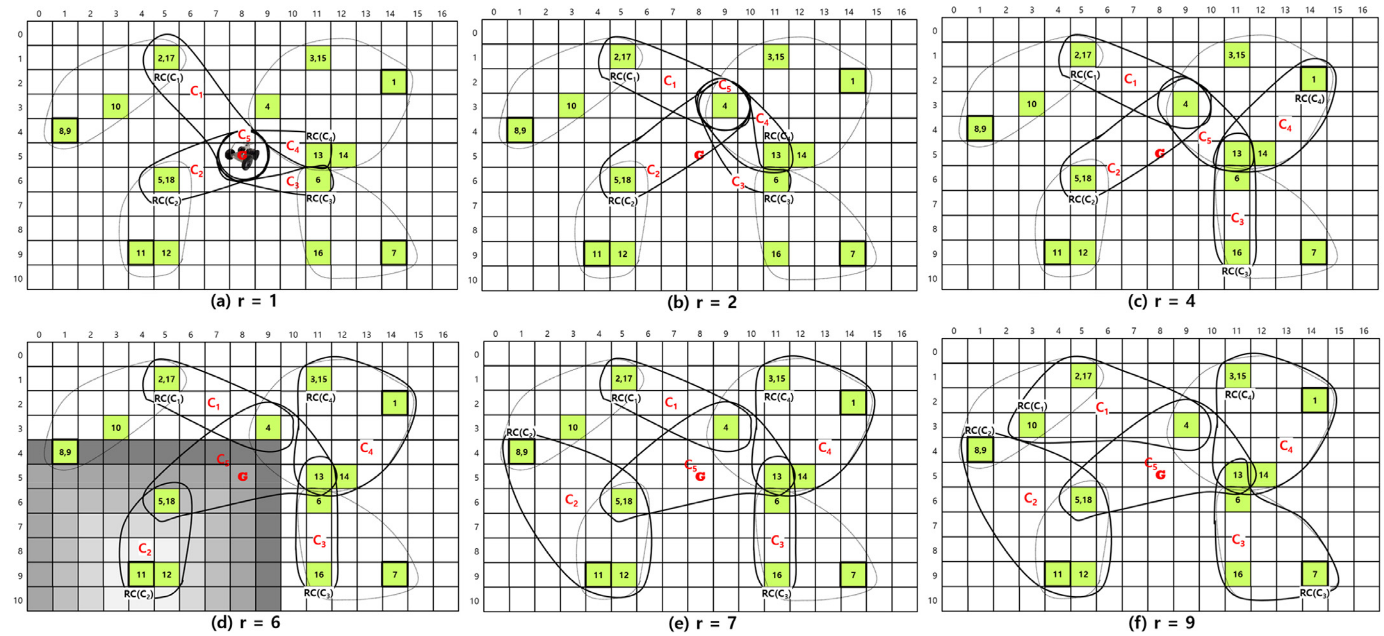

Fig. 3. During the 2nd phase, LEEF performs two steps referring to the found VCs in the 1st phase. (a)–(f) show the first step during which the least energy consumed cluster takes a turn to grow by including a cell.

est path where M_i visits every cell c whose $ID(c)$ equals C_i once, and returns back to $Anchor(C_i)$. Since finding $TR(M_i)$ is mapped into the *Travelling Salesman Problem with a radio range*, the heuristics [23,34,35] can be used to find $TR(M_i)$.

- $\boldsymbol{E_i^r}$: Denotes the energy experienced by M_i in round r (while executing the 2nd phase of LEEF) for up/downloading intra-cluster data in C_i and inter-cluster data at $Anchor(C_i)$, i.e., $E_C(M_i)$ and also during completing one tour of the segments of C_i, i.e., $E_M(M_i)$, based on the membership of C_i in round r. $E_C(M_i)$ and $E_M(M_i)$ are computed based on Eqs. (3) and (4), respectively.

In Fig. 2 assume that $Data(S_1,S_{13})=25$Mb, $Data(S_4,S_{17})=10$Mb, $Data(S_7,S_{16})=15$Mb, $Data(S_{10},S_{17})=40$Mb, and $Data$(between every pair of the remaining terminals) = 5Mb and the cell size is about 120 m. Thus, the required energy to move across one cell is 120 J according to the Eq. (3) where $\mu=1$ J/m, while 10 J is required to transmit 5Mb over wireless communication according to (4). In such a configuration, the overall cell-based tour length of MDCs is 53 cells and require $53 \times 120=6360$ J. The total energy to up/download data equals 4140 J. We use the parameters setup as an example while explaining the 2nd phase of LEEF.

4.2.1. Greedy-expansion step

The VCs are formed based on the MDC tours without factoring in the data communication overhead. Starting with the VCs, this step strives to balance the energy consumption between MDCs serving neighboring clusters and operates in rounds. In each round, the cluster whose MDC consumes the least energy is considered for expansion. During this step, LEEF tries to keep C_k at a center. In the first round ($r=1$), C_k equals VC_k, i.e., $C_k=\{c_g\}$, where c_g is the cell corresponding to G which M_k is positioned at. The initial non-hub cluster includes just the c_g and the closest cell in VC_i, i.e., $C_i=C_k \cup \{c \mid \min_{c \in VC_i} CellDist(c, c_g)\}$ for $i<k$. Then the selected c for C_i becomes a most recently joined cell of C_i hereafter denoted as $RC(C_i)$. For C_k, c_g becomes $RC(C_k)$. Based on the initial clusters, E_i^1 $\forall i$ are computed. Since M_k initially acts as a stationary relay, i.e., $E_M(M_k)$ is zero, E_k^1 equals $E_C(M_k)$. For the VCs in Fig. 2, the clusters' layout after the first round is shown in Fig. 3(a). The cluster formation process continues iteratively by growing the initial C_i $\forall i$, where in each iteration LEEF tries to expand the cluster whose MDC experiences the least energy overhead. Thus, from the 2nd round ($r \geq 2$), the cluster C_i with the least E_i^{r-1}, C_{least}, is processed. Two scenarios can be experienced as follows:

(i) C_{least} is C_k: In this case, there are two possible actions for adjusting C_k: (1) re-locating M_k from c_g to the nearest cell $c_{to} \in CS_T$. This applies for sure for $r=2$ seen in Fig. 3(b) since the other clusters have only one cell each; (2) expanding C_k by adding the closet cell $c \in CS_T$ to $RC(C_k)$ which we refer to as the best new cell (BC) for C_k. Fig. 3(c) and (d) illustrate this scenario, where $r=4$ and 6 respectively. For both scenarios, LEEF adjusts C_i $\forall i<k$, accordingly such that each M_i may rendezvous with C_k at different $Anchor(C_i)$ in order to shorten its $TR(M_i)$.

(ii) C_{least} is C_i, $i \neq k$: The expansion of C_i is processed by considering VCs. LEEF first searches VC_i for selecting the *not_clustered* cell that is closest to $RC(C_i)$. If no *not_clustered* cell exists in VC_i, i.e., every cell in VC_i has become part of C_i, the neighbors of VC_i $VC_{NBR(i)}$, are searched for an appropriate choice. Since the fundamental objective of LEEF is to minimize a tour length of MDCs such that the inter-segment delivery delay is reduced, only the cell $\in VC_{NBR(i)}$ that is adjacent to VC_i can become BC of C_i. Fig. 3(d) shows the case in which the best cell of C_2 in round 6 denoted as BC_2^6 is to be looked for in a neighboring virtual cluster VC_1. By searching around $RC(C_2)$, i.e., $c(9,4)$, the first found cell c, $c(4,1) \in VC_1$ that is *not_clustered* becomes BC_2^6, and get added to C_2. This is illustrated in Fig. 3(e) where the search starts from the square centered at $c(9,4)$ and extend outward while searching. If the found c is already a member of another cluster, i.e., $ID(c) \neq not\_clustered$, then clustering C_i is not further processed, i.e., C_i is completely formed. After that the identified BC_i^r becomes a new $RC(C_i)$.

In Fig. 3, for each of (a) to (f) a set of E_i^r, $i=1,2,3,4,5$ in a round r is represented as $E_i^r=\{E_1^r, E_2^r, E_3^r, E_4^r, E_5^r\}$, i.e., $E_i^1=\{1050, 810, 770, 770, \underline{260}\}$, $E_i^2=\{1080, 1070, 780, 540, \underline{260}\}$, $E_i^4=\{1140, 1370, 1130, \underline{1050}, \underline{1050}\}$, $E_i^6=\{1220, \underline{1100}, 1210, 1820, 2500\}$, $E_i^7=\{1260, 2110, \underline{1250}, 1900, 2860\}$, $E_i^9=\{2000, 2190, 1820, 1980, 3340\}$ for a round

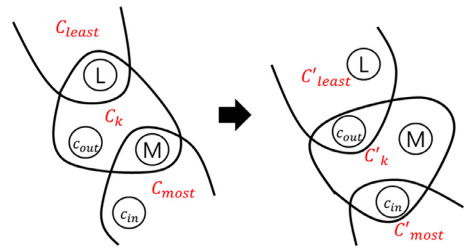

Fig. 4. An example of the optimization plan for the case C_k is neither C_{most} nor C_{least}; $L = Anchor(C_{least})$ and $M = Anchor(C_{most})$.

1, 2, 4, 6, 7, and 9 respectively (unit: Joule). The underlined value means the least consumed energy consumption in the round and its corresponding cluster is considered for expansion in the next round. Finally Fig. 3(f) shows a resulting clustering topology after the first step.

4.2.2. Optimization step

The greedy nature of the least-energy based cluster expansion may cause the formed clusters to be unbalanced. Basically, the growth of a cluster in a specific round may widen the gap between the clusters with the least and most energy overhead C_{least} and C_{most}, respectively. Therefore, LEEF conducts optimization between non-hub clusters by adjusting the cluster membership between C_{least} and C_{most} and opting to further balance $E(M_i)$ among the MDCs. The *optimization* step operates in rounds. In each round, C_{least} is expended and C_{most} is shrunk with respect to C_k. There are two different cases for *shrinking* C_{most} and *expanding* C_{least}:

① C_k is neither C_{most} nor C_{least}: in order to downsize C_{most} the cell $c_{in} \in C_{most}$ which is the closest to $Anchor(C_{most})$ is selected. By excluding c_{in} from C_{most}, i.e., $C_{most} - =\{c_{in}\}$ $E(C_{most})$ would be reduced by a sum of E_C(data volume exchanged between c_{in} and $Anchor(C_{most})$) and E_M(tour to c_{in}). Next, C_{least} will grow towards the updated hub-cluster $C_k' = C_k \cup \{c_{in}\}$ by including the cell $c_{out} \in C_k'$ that is closest to $Anchor(C_{least})$, i.e., $C_{least} \cup = \{c_{out}\}$. Consequently, $E(C_{least})$ would increase by a sum of E_C(data volume exchanged between c_{out} and $Anchor(C_{least})$) and E_M(tour to c_{out}). Fig. 4 illustrates the idea. Thus, C_k finally would become $C_k \cup = \{c_{in}\} - \{c_{out}\}$ and $E(C_k)$ is adjusted by substituting E_C(data volume exchanged between c_{out} and $Anchor(C_{least})$) and E_M(tour to c_{in}) and by adding E_C(data volume exchanged between c_{in} and $Anchor(C_{most})$) and E_M(tour to $Anchor(C_{least})$). For the special case where C_k includes only one cell C_{out}, C_k will be relocated by moving M_k from C_{out} to C_{in}.

② C_k is either C_{most} or C_{least}: the procedure is the same as ① however, only *shrinking* (if C_{most} is C_k) or *expanding* (if C_{least} is C_k) becomes necessary.

Based on the planned optimization in round r, LEEF recalculates $E_i^r, \forall i$. The computed E_i^r also factors in any changes to $Anchor(C_i)$ for non-hub clusters other than C_{least} and C_{most}. Then the optimization is performed according to the planned scenario in round r only if the standard deviation (SD) of E_i^r $\forall i$ is less than that of the previous round (r-1), i.e., SD of E_i^{r-1}. Otherwise the *optimization* step terminates and clusters formed in the round (r-1) becomes a final topology. Fig. 5(a) shows a resulting topology of k clusters ($k = 5$). The final federated topology of the 18 terminals seen in Fig. 1(a) is depicted in Fig. 5(b). A summary of the notations and the pseudo code of LEEF can be found in Appendices A and B, respectively.

4.3. Algorithm analysis

In this subsection we analyze the time complexity of LEEF using the following lemmas and theorem:

Lemma 1. *The time complexity of modeling the area into the grid structure and computing CS_T during the 1st phase of LEEF is bound to $O(\frac{L}{R^2} + n_c \cdot |S_T|)$, where L is the area size of the deployment region, S_T is a set of segments, and n_c is the number of the selected cells from which multiple segments are reachable, i.e., $n_c \le |S_T|$.*

Proof. During the 1st phase of LEEF, the deployment region is represented as a grid of square-shaped cells whose side is $R/\sqrt{2}$ in length and each cell c_i is evaluated considering its *reachability* to segments and *proximity* to G around which a hub-cluster is formed. The time complexity of this step is proportional to the number of cells in L that is $\frac{2L}{R^2}$. CS_T is determined by mapping to a set-covering problem which can be solved in $O(M \cdot N)$, where M is the number of sets and N is the total number of elements in the union of all sets [33]. In our case, M equals the number of selected cells in CS_T from which multiple segments are accessible via an MDC, $M = n_c$ and N is the number of terminals i.e., $|S_T|$. Therefore, the time complexity of modeling the deployment region and finding CS_T is proportional to n_c and $|S_T|$ and inversely proportional to R as $O(\frac{L}{R^2} + n_c \cdot |S_T|)$, where $n_c \le |S_T|$. □

Lemma 2. *The time for forming virtual clusters is primarily determined by computing a travel path in a cluster and is bound to $O(n^2 \log n)$, where $n < (n_c - k) \le |S_T| - k$.*

Proof. The formation of k virtual clusters (VCs) is performed in rounds; in the first round every segment forms an individual clus-

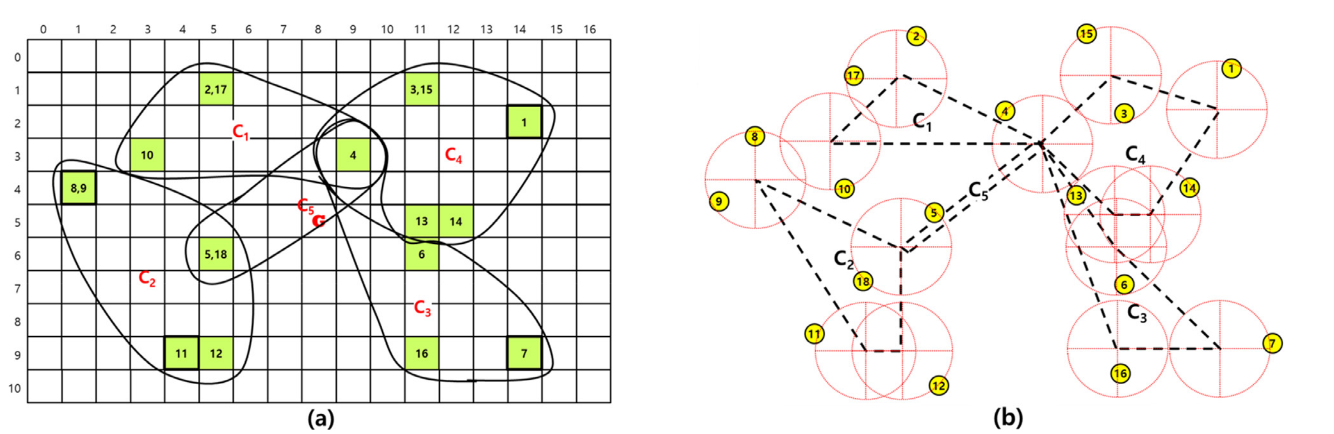

Fig. 5. (a) shows the final formation of k clusters after the *optimization* step in order to minimize a gap of energy consumption between clusters during the *greedy-expansion* step. Finally, 18 terminals seen in Fig. 1(a) are federated by 5 MDCs in a star topology as seen in (b) where each dotted line represents $\boldsymbol{TR}(\boldsymbol{M_i})$.

ter and then in subsequent rounds the two clusters whose merging achieved the most reduction in the length of the combined tour are merged. Hence, it takes$(n_c - k + 1)$ rounds to conclude. Assuming that the tour of each cluster C_i is determined using the heuristic in [23], the execution time will be O($n\log n$), where n is the number of segments in C_i. In our case, n is 1, 2, 3, ..., $(n_c - k + 1)$ based on the round; thus the time complexity of forming k VCs is equal to $\sum_{n=1}^{(n_c-k+1)} n\log n$, where $n_c \le |S_T|$ which bounds to $O((|S_T| - k)^2 \log(|S_T| - k))$. □

Lemma 3. *The time complexity of each iteration of the greedy-expansion and optimization step during the 2nd phase of LEEF is primarily dependent on the complexity to compute the tour length of a cluster, and is $O(n_c + n\log n)$, where $n < (n_c - k) \le |S_T| - k$.*

Proof. In the *greedy-expansion* step of the 2nd phase, LEEF opts to form k clusters in rounds by expanding the least energy consuming cluster and shrinking the most energy consuming cluster while opting to balance E_i^r $\forall i$ in round r. Thus, the main computation in each round is for finding the best cell $BC_i^r \in CS_T$ for inclusion or exclusion by searching cells $\in VC_i$ formed in the 1st phase and computing a tour of the updated cluster in the round. In addition in the *optimization* step in each round, a similar process is performed for two clusters between which there is the most energy gap and the tour is updated accordingly. Therefore, the time complexity of each round of the 2nd phase is primarily affected by the number of the searched cells and the tour computation which are O(n_c) and O($n\log n$), respectively, where n is the number of cells in a cluster $C_i, i = 1, \ldots, k$ in round r i.e., $|C_i^r| \langle n_c - k \le |S_T| - k$ in any r, which is $O(n_c + n\log n)$. □

Theorem 1. *The time complexity of LEEF is proportional to the gap between the number of terminals, $|S_T|$ and the MDC count, k, and thus bounds to $O(\frac{L}{R^2} + |S_T|^2 + n^2\log n)$, where $n \le |S_T| - k$.*

Proof. As proven in Lemmas 1 and 2, the time complexity of the 1st phase of LEEF is bounded to O($\frac{L}{R^2} + n_c \cdot |S_T| + n^2\log n$), where $n < (n_c - k) \le |S_T| - k$. In addition, the 2nd phase consists of a *greedy-expansion* step and an *optimization* step, both of which operate in rounds; the time complexity of each round is $O(n_c + n\log n)$, where $n < (n_c - k)$ as seen in Lemma 3. Moreover, the *greedy-expansion* step is iterated O(n_c) times and the number of iterations of the *optimization* step is O(1). Therefore, the 2nd phase's time complexity bounds to $O(n_c \bullet (n_c + n\log n))$. Thus, the overall time complexity of LEEF is bound to O($\frac{L}{R^2} + n_c \cdot |S_T| + n^2\log n + (n_c \cdot (n_c + n\log n))$), that is O($\frac{L}{R^2} + |S_T|^2 + n^2 logn$), where $n < (n_c - k) \le |S_T| - k$. Hence given S_T, the time complexity of LEEF diminishes as k increases. □

4.4. Treatment of special cases

LEEF federates segments by providing the intermittent connectivity among them with limited number of MDCs (k) by forming the energy balanced star-shaped topology. During federation, LEEF focuses on the case $k \ge 3$, and for $k < 3$ a special treatment is needed. For k=1, some heuristics [23,34,35] for solving the *travelling salesman problem* can be applied for providing intermittent connectivity among terminals in S_T with only one MDC. On the other hand, for k =2, S_T is bifurcated into two clusters C_1 and C_2 based on the two furthest pair of terminals. Then the tours of the individual MDCs, namely, M_1 and M_2, is determined in the same way as the case of k=1 and the inter-cluster communication is performed via two closest terminals $S_a \in C_1$ and $S_b \in C_2$ that are shared between M_1 and M_2 for their rendezvous.

Moreover, there is another type of special cases due to an excessively long tour lengths or large data communication volume

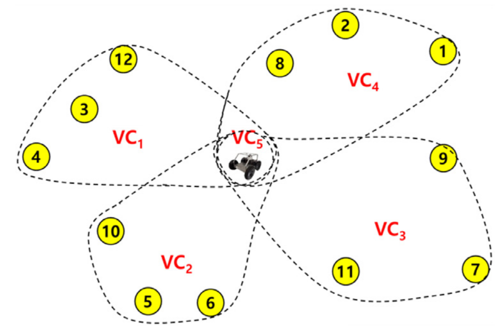

Fig. 6. In case of $E_M >> E_C$, M_i i = 1,..(k−1) suffer from heavy energy consumption in touring to C_k around at G.

between segments. These cases represent the energy overhead for the overall inter-terminal data traffic is (i) negligible or (ii) dominant relative to that for MDC motion, as elaborated below:

(i) The individual MDCs may experience excessively long tour due to a layout in which the terminals are located far from G close to the border of the area, as seen in Fig. 6. For this case, which corresponds to EM >> EC, LEEF keeps M_k at G and opts to form the energy balanced non-hub clusters Ci ∀i < k by evening tour lengths among the MDCs serving non-hub clusters Mi ∀i < k. Based on VCs, $TR(M_i)$ ∀i < k are balanced, and consequently $E_M(M_i)$ ∀i < k. Since for EM >> EC, equalizing $E_C(M_i)$ ∀i < k would make marginal impact on balancing the overall energy overhead among non-hub clusters. Therefore, VCs become final clusters. However, the energy imbalance between a hub-cluster Ck and non-hub clusters remains.
Since M_k serves as a stationary relay among the (k-1) non-hub clusters and at which the MDCs rendezvous for up/downloading the inter-cluster data, the load imbalance is relieved by making M_k share the touring overhead with the other MDCs. In other words, Mi takes turn in staying at G while M_k serves Ci. The frequency of swapping duties of Mi and M_k is determined based on $TR(M_i)$. Given that M_k is to help in alleviating the motion overhead on MDCs serving non-hub clusters, the role swapping schedule is set based on the length of the individual MDC tours relative to the total traveled distance in a single round of data collection. Basically, the frequency of M_k touring Ci is proportional to $\frac{TR(M_i)}{\sum_{\forall i<k} TR(M_i)}$. Since for EM >> EC $TR(M_i)$ ∀i < k have been balanced and energy consumption on data communication is marginal, E(Mi) ∀i < k are even among Ci ∀i < k formed based on VCs. Therefore, such a role rotation will balance E(Mi) ∀i.

(ii) When M_k stays a stationary relay at G, the volume of data exchanged between clusters that M_k will relay is:

$$\sum_{\forall c_i, c_j \in CS_T} Data(c_i,\ c_j),\ \text{where}$$

$$Data(c_i,\ c_j) = \sum_{S_{i\_} \in L(c_i),\ S_{j\_} \in L(c_j),\ S_{i\_} \neq S_{j\_}} Data(S_{i\_}, S_{j\_}).$$

For this case, which corresponds to $E_C >> E_M$, LEEF initially sets up $C_i = VC_i$ $\forall i$ and tries to equalize the data volume transported by the MDCs while keeping M_k at G in order to balance energy among M_i $\forall i < k$ on. The procedure is similar to the *optimization* step and operates in rounds. In each round, the cluster C_{most} with most energy overhead is picked and its $E(M_{most})$ is reduced by moving out one cell $c_{out} \in C_{most}$ to one of the neighboring cluster (C_{nbr}) that consumes the least energy. Namely, the closest cell $c \in C_{most}$ to C_{nbr} becomes c_{out}. Therefore, C_{most} is downsized and its adjacent cluster C_{nbr} grows instead by one cell each. Since C_k hardly becomes C_{most}

in this case, $Anchor(C_i)$ $\forall i$ would not be modified. The procedure is iterated until no more balancing among $E(M_i)$ $\forall i < k$ is achievable.

If such energy imbalance between M_k and the other MDCs is not fully addressed by the above optimization, a role swapping is pursued. Unlike the case $E_M >> E_C$, $TR(M_i)$ $\forall i < k$ are not balanced while $E(M_i)$ $\forall i < k$ are evened (since both communication and travel overhead are factored in), and $E_C(M_k)$ is excessive. Therefore, the frequency of swapping M_i and M_k is determined based on the total energy (rather than tour length) and relative to all MDCs (rather than non-hub MDCs), i.e., the frequency of M_k touring C_i is proportional to $\frac{E(C_i)}{\sum_{\forall i} E(C_i)}$. Since the goal is to even the energy load, if $E(M_q)$ is close to the average energy for all MDCs i.e., $\frac{\sum_{\forall i} E(C_i)}{k}$, it is unnecessary swapping M_k and M_q and only those non-hub MDCs whose energy is significantly below average are swapped with M_k.

5. Performance evaluation

The effectiveness of LEEF is validated through simulation. This section discusses the simulation environment, performance metrics, and varied parameters. The simulation results of LEEF are compared to recently-published competing algorithms.

5.1. Simulation environment and performance metrics

We have implemented a simulation environment in Python. The environment articulates the effect of damage on a single WSN that originally covers an area, or the presence of multiple autonomous WSNs. Basically, varying numbers of segments are located in an 1200 m × 1200 m area of interest using uniform random distribution. The following parameters are used to vary the network characteristics:

- *Communication range of relays* (R): Generally, R affects the length of tour path; MDCs need to travel a shorter path with a longer R. In addition, R impacts the segment clustering in LEEF since clustering is cell-based and the cell size is based on R. As noted earlier, both MDCs and the nodes of a segment are assumed to have the same communication range "R".
- *Number of segments* (N_{seg}): Having high segment count may increase the connectivity requirement and thus more energy may be needed for MDCs because of the longer tour and potentially larger volume of relayed data. $N_{seg} = |S_T|$.
- *Number of MDCs* (N_{MDC}): The given number of MDCs is assumed to be less than the least count of the required relay nodes to form a perpetual topology of N_{seg} segments, denoted as N_{RN}. We use the algorithm in [10] to compute N_{RN} which primarily depends on N_{seg} and the layout of segments. In the simulation, N_{MDC} is determined based on N_{RN} as explained in subsection C below. As a value of N_{MDC} gets smaller, MDCs may consume more energy due to visiting more segments.
- *Average Inter-Segment Data Volume* ($ISDV_A$): It represents the data traffic requirement between each pair of segments. The value of $ISDV_A$ would influence the latency and consumed energy by MDCs for wireless communication.

The performance of LEEF is assessed using the following three metrics:

- *Maximum inter-segment communication delay* (D_{max}): Obviously, LEEF strives to minimize the maximum data delivery latency between segments. D_{max} is affected by R and $ISDV_{SD}$.
- *Energy balance among MDCs* (EB): This is also a main objective for LEEF. Since LEEF tries to balance energy considering both the motion and communication overhead, EB is computed as a standard deviation of total energy consumption of each MDC, i.e., EB equals $\sqrt{\frac{\sum_{i=1}^{N_{MDC}} |E(M_i)-\overline{E(M)}|^2}{N_{MDC}}}$, where $\overline{E(M)} = \frac{\sum_{i=1}^{N_{MDC}} E(M_i)}{N_{MDC}}$ and $E(M_i) = E_M(M_i) + E_C(M_i)$.
- *Average energy consumption of all MDCs* (AEC): This measures the average energy consumed by all MDCs for intermittently connecting segments.

5.2. Baseline approaches

The performance of LEEF is compared to three competing approaches. Two of the baseline approaches, namely, TOCS [30] and FOCUS [29], opt to reduce inter-segment communication delay. TOCS [30] forms clusters of segments in a star topology like LEEF and balances tour lengths of MDCs, while FOCUS [29] strives to shorten the tour lengths and adjust the MDC travel speed in order to minimize the waiting time required for them during rendezvous and transfer of inter-clustering data. On the other hand, the focus of the third baseline approach, namely, MINDS [25], is more on reducing and balancing the MDC tours without considering the data delivery delay between segments. Overall, the three baseline approaches address the same problem tackled by LEEF using different solution strategies, as summarized below.

- TOCS forms clusters around the center of the area G and consists of two phases. In the first phase, each segment S_i initially becomes a cluster C_i. The tour of a cluster is defined by visiting G and each segment in the cluster. Then the two clusters whose merging has the least tour are combined. The merging is repeated until (k-1) clusters remains. Then a center-cluster C_k is formed by including (k-1) rendezvous points P_i, i=1,..,(k-1), each of which is computed as a mid-point between G and the closest point "x" to G on a convex hull of the segments of C_i. Then each tour path is determined as done in IDM-kMDC [23] and the average tour length TL_{avg} is found. TOCS then adjusts the size of C_k in order to balance tours among MDCs in the 2nd phase. Basically, C_k expands towards C_i whose tour length TL_i is larger than TL_{avg} by moving P_i along the line between G and x of C_i until $TL_i > TL_{avg}$. In the case of $TL_i < TL_{avg}$ P_i moves towards G. The same process is repeated for each C_i and the 2nd phase terminates when $TL_{avg} \approx TL_i$, $\forall i$. In TOCS the waiting time at the rendezvous point may contribute to the inter-segment communication delay due to lack of time synchronization and due to varying data uploading/downloading time.
- FOCUS operates in three phases. In the 1st phase, segments are grouped into k disjoint clusters based on proximity. In order to shorten the MDC tours, clustering is initiated by selecting the k farthest segments from the centroid and then the remaining segments join the closest cluster in a greedy manner. The set of k disjoint clusters is represented as a directed graph G=(V, E), where V reflects clusters and E contains a set of weighted directed edges $\overrightarrow{C_i, C_j}$ between every pair of clusters. The weight of $\overrightarrow{C_i, C_j}$ is equal to the increase in the tour length of C_i when C_i is extended towards C_j by including a segment from C_j. During the 2nd phase FOCUS opts to overlap k clusters in order to minimize MDC tour lengths for the inter-cluster communication by selecting an intersection segment (IS) between two adjacent clusters C_i and C_j. The pair of C_i and C_j corresponds to two end vertices in each edge of the computed minimum spanning tree (mst) of G. In the 3rd phase, FOCUS adjusts the motion speed of individual MDCs for reducing the time spent at IS for data communication between clusters.
- MINDS strives to balance tour lengths among k MDCs and consists of three phases. In the 1st phase, it calculates an mst of segments. If only one MDC is available, the MDC tours along the found mst. Otherwise, MINDS splits the longest tour L into two groups until the number of groups equals k in the 2nd phase.

Table 1
Simulation setup.

Parameter	Description	Value (unit)
Area	Area of interest	$1200\,m \times 1200$ m
R	Radio range of an MDC	[50, 250] (meters) with increment of 25
N_{seg}	Number of segments	[21, 39] with increment of 3
N_{MDC}	Number of given MDCs	[$N_{RN}-1$, $N_{RN}-2$, .., y,] N_{RN} is computed using [10], $y=\max(N_{RN}-5, 2)$, e.g., 5,7,9,11,13, and 15
$ISDV_A$	Expected value for the inter-segment data volume requirement	1,2,4,8,16,32,64 (Mbps)
$E_M(M_i),\forall i$	Energy for motion	1 (Joule/meter)
$E_C(M_i),\forall i$	Energy for wireless communication	2 (Joule/Mbit)
$Speed_{MDC}$	Speed of an MDC	6 (meter/min)
BW_{MDC}	Wireless bandwidth	100 Kbps

Splitting is done by finding a center vertex V_c of L from which the path on *mst* to the furthest segment is minimized. Then segments in L are divided into two groups g_i and g_j and V_c becomes a rendezvous point between MDC_i and MDC_j. Forming g_i and g_j is determined depending on the node degree of V_c. If there are more than two edges connected to V_c, then grouping takes three steps. First g_i is formed by the segments connected through the edge along which the furthest segment is reached from V_c. Then the remaining segments are grouped into g_j in the same way. If there are still ungrouped segments, each of them is assigned to the closest group. In the 3rd phase, the tour path of MDCs is computed as done in [23].

Overall, all three baseline approaches try to minimize tour lengths. Additionally TOCS and MINDS opt to reduce the average inter-segment data delivery latency. Also, in FOCUS and MINDS, MDCs rendezvous at a common segment for transferring inter-cluster traffic, while in TOCS they meet at a point where no segment is located. Therefore, in order to minimize the rendezvous time TOCS strives to equalize the MDC tours. However, TOCS does not include a scheduling algorithm for MDCs' rendezvous. FOCUS also opts to reduce the buffering space and the time consumed at the common segment by adjusting the motion speed of MDCs. However, unlike LEEF, all three baselines do not consider the communication energy overhead.

5.3. Simulation results

We have simulated multiple configurations, each has different combination of values *R*, N_{seg}, N_{MDC}, and $ISDV_A$. The value of *R* is varied from 50 to 250 with increments of 25 and N_{seg} takes the values between 21 and 39 with increments of 3. In addition, the data volume between segments is randomly picked using a Gaussian distribution with mean, $ISDV_A$, that takes the value 1, 2, 4, 8, 16, 32, and 64, and uses a standard deviation of 3.0. The settings of other simulation parameters are summarized in Table 1. We note that the value for $Speed_{MDC}$ is based on nominal speeds of search-and-rescue robots, e.g., IRS Soryu [36], when operating in rough terrain. IRS Soryu has a maximum speed of 0.37 m/s, i.e., about 22 m/min. Robot speed has been found to drop by factor of 70–75% in rough train [37], and hence we used 6 m/min in the simulation. In the simulation, N_{MDC} is no less than 2 since intuitively for $N_{MDC}=1$, no inter-cluster data communication is required. The results of the individual experiments are averaged over 30 runs and all results thus stay within 10% of the sample mean due to 90% confidence interval analysis.

5.3.1. Maximum inter-segment data delivery latency (D_{max})

The results in Fig. 7(a) show that less delivery delay is experienced under all approaches as *R* increases. This is very much expected because a larger communication range enables an MDC to reach a segment from further and its tour could consequently be reduced. Meanwhile, LEEF yields the least D_{max} among all approaches, which is attributed to the following three design features: (1) forming a star-shaped inter-cluster topology where the number of clusters between every pair of segments becomes at most three (like TOCS, and unlike FOCUS and MINDS); (2) MDCs in LEEF do not need to pause and wait for exchanging data between clusters during their travel and thus no extra time for rendezvous is needed in the data delivery (unlike TOCS); (3) LEEF strives to keep a pair of segments which exchange large volume of data in the same cluster by considering the inter-segment data exchange overhead during cluster formation. This reduces the inter-cluster traffic volume.

It is worth noting that TOCS yields the worst D_{max} despite forming a star-shaped inter-cluster topology. This is because TOCS opts to even tour lengths of MDCs, without scheduling the MDC's trip to reduce rendezvous time. Thus the trip of an MDC is delayed at rendezvous points, which affects the total data delivery latency between segments. Moreover, Fig. 7(b) shows that D_{max} increases for all algorithms as N_{seg} grows. This is because longer tours are to be traveled and larger data volumes are to be transported, and thus the load on the individual MDCs accordingly rises. However, LEEF scales well and outperforms all baselines in terms of D_{max} regardless of the segment count, especially with large N_{seg}, as seen in Fig. 7(b). Such scalability is because LEEF identifies the positions from which an MDC can access multiple segments, and exploits them for calculating the shortest tours that cover all segments; the impact of such a design feature becomes apparent when more segments are to be federated.

Moreover, LEEF outperforms all baselines and yields the least D_{max} for various numbers of MDCs as seen in Fig. 7(c). Unlike the other algorithms, LEEF takes advantage of the increased MDC count in reducing D_{max} because it forms a star-shaped inter-cluster topology and includes no waiting overhead for rendezvous between MDCs for inter-cluster traffic. The star-shaped topology has also helped TOCS in reducing D_{max}, compared to the other baseline approaches. Meanwhile, FOCUS and MINDS incur more delay during the inter-segment data delivery as N_{MDC} increases since they form clusters by equally cutting the minimum spanning tree of segments. Thus, the more MDCs to be involved in FOCUS and MINDS the larger the cluster count becomes, and consequently the more clusters exist on the inter-segment data paths, which introduces extra delay for up/downloading inter-cluster data.

Fig. 7(d) shows that the increased inter-segment data exchange causes D_{max} to grow. It is quite expected since a larger volume of data involves more delay for up/downloading. However, it is worth noting that D_{max} in LEEF is much less than the baseline approaches. For example with $ISDV_A$ of 64Mb, LEEF yields about 12% of D_{max} of FOCUS. Moreover, the rate of growth in D_{max} with respect to $ISDV_A$ is much slower in LEEF than the other approaches, mainly due to the fact that LEEF considers the inter-segment data volume while forming clusters, and favors keeping a pair of segments in

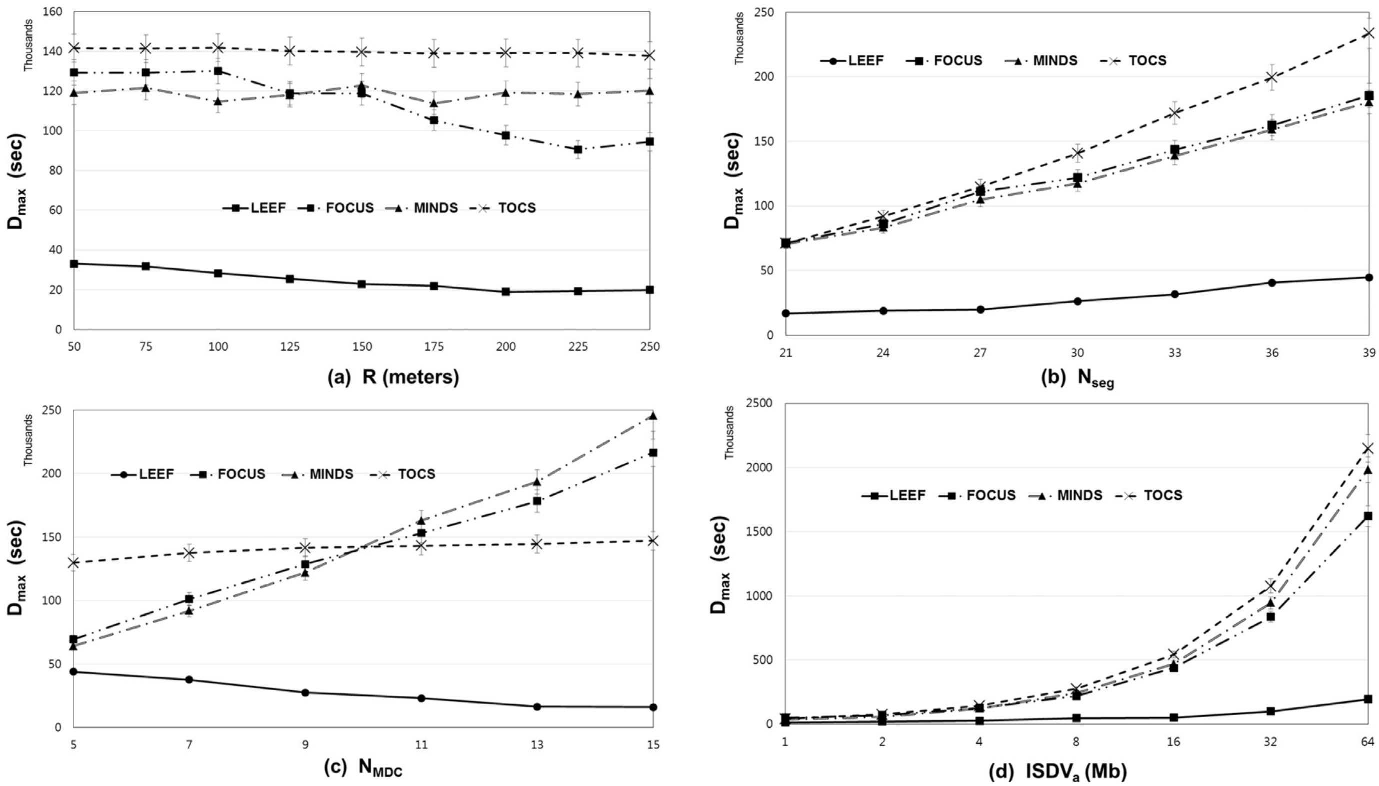

Fig. 7. The performance comparison of LEEF to TOCS, FOCUS, and MINDS in terms of the inter-segment data delivery latency ($\boldsymbol{D_{max}}$). (a)–(d) show the results with respective to various R, $\boldsymbol{N_{seg}}$, $\boldsymbol{N_{MDC}}$, and $\boldsymbol{ISDV_A}$ respectively with a fixed value of R = 100, $\boldsymbol{N_{seg}}$ = 30, $\boldsymbol{N_{MDC}}$ = 9, and $\boldsymbol{ISDV_A}$ = 4Mb.

the same cluster if they have to share lots of data. Such a feature also reduces data up/downloading overhead in LEEF.

5.3.2. Energy balance among MDCs (EB)

Fig. 8 shows the effectiveness of LEEF with respect to energy balance among MDCs. LEEF outperforms all baseline approaches regardless of the communication range (R), the number of segments (N_{seg}), MDC count (N_{MDC}), or the average inter-segment data volume ($ISDV_A$). Such a distinct performance is because during inter-cluster topology formation LEEF strives to equalize the energy consumed by MDCs while touring a cluster and up/downloading intra- and inter-cluster data.

While R has no influence on balancing the energy overhead, as shown in Fig. 8(a), all approaches form more energy-balanced inter-cluster topology with smaller N_{seg} as seen in Fig. 8(b). This phenomenon is evident especially in FOCUS, MINDS, and TOCS, since they opt to balance only the MDC tour lengths during clustering unlike LEEF which considers both inter-segment data volume and proximity. In addition, Fig. 8(b) indicates that FOCUS and MINDS form more energy-balanced topologies than TOCS regardless of N_{seg}. This is because FOCUS and MINDS perform clustering based on an *mst* of segments. Overall, once N_{seg} grows the effect of data exchange becomes more apparent and the performance gap between LEEF and the baseline apporaches widens significantly. For example, for setups with 39 segments, LEEF outperforms FOCUS and MINDS 1:2, and TOCS 1:5.

In Fig. 8(c), as N_{MDC} increases LEEF and TOCS yield better performance, yet the performance of FOCUS and MINDS diminishes. Basically, having a central cluster in LEEF and TOCS facilitates splitting segments into k energy-balanced clusters as the value of k increases. Both FOCUS and MINDS are *mst*-based and the inter-cluster proximity becomes difficult to control as the number of clusters (MDCs) increases. In addition, the inter-segment data volume signficiantly affects the energy imbalance as seen in Fig. 8(d). When the data volume grows, LEEF delivers very distinct performance; this is because LEEF factors in intra- and inter-cluster data exchange when associating segments to clusters. Such a feature is a key advantage of LEEF over TOCS, and explains the large performance gaps between these two approachs in Fig.8(b) and (d).

5.3.3. Average energy consumption of all MDCs (AEC)

Fig. 9 presents the average energy consumed by MDCs while varying R, N_{seg}, N_{MDC}, and $ISDV_A$, respectively. Fig. 9(a) shows that all approaches consume less energy as R grows. This is because a larger communication range enables the formation of shorter MDC tours and reduces the energy consumed in travelling clusters. Considering both Figs. 8(a) and 9(a), we can conclude that LEEF forms the most effective inter-cluster topology with reduced total energy overhead and balanced energy load among the involved k MDCs for $R \geq 75$. In comparison to TOCS which, like LEEF, forms a star-shaped inter-cluster topology, LEEF not only requires about 240% less energy for MDCs operation than TOCS but also achieves energy balance among MDCs approximately 440% better than TOCS when R is 250.

Like Fig. 8(b), Fig. 9(b) shows that more energy is consumed by MDCs in all approaches as the number of segments increases; mainly due to the increased tour length and data volume. However, LEEF incurs less energy overhead than the baselines, especially when $N_{seg} \geq 24$. Again, it is because LEEF factors in both inter- and intra-cluster data sharing overhead in addition to the energy consumed in MDC tours. Moreover, in LEEF MDCs consume less energy than all other approaches when N_{MDC} is larger than 9 as seen in Fig. 9(c). It is worth noting that the total energy consumption in LEEF and TOCS is reduced as N_{MDC} increases while those of FOCUS and MINDS grow with large N_{MDC}. It is thus evident that the effectiveness of forming k-clusters in a star topology increases with larger k in terms of energy overhead, as well as data

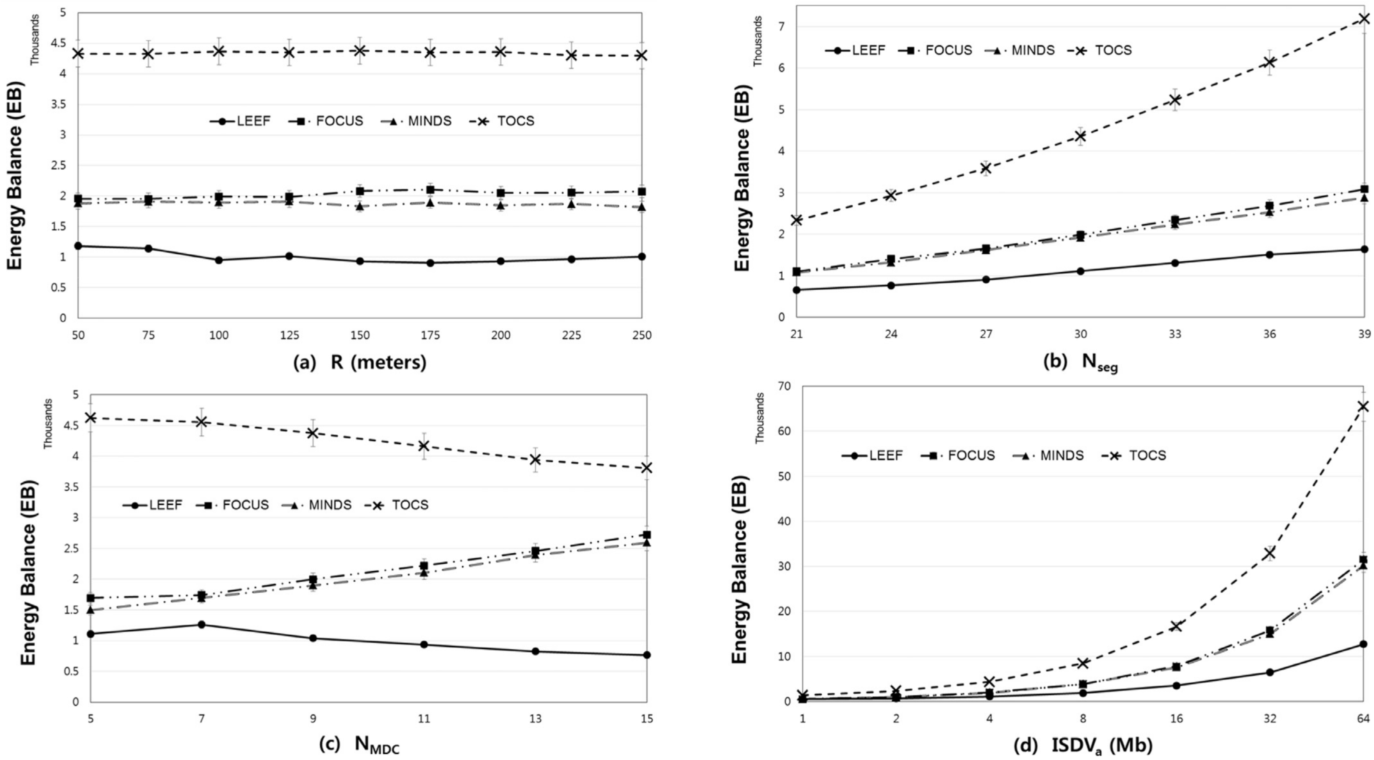

Fig. 8. (a)–(d) show the performance comparison of LEEF to TOCS, FOCUS, and MINDS in terms of energy balance with respect to various R, $\boldsymbol{N_{seg}}$, $\boldsymbol{N_{MDC}}$, and $\boldsymbol{ISDV_A}$ respectively with a fixed value of $R = 100$, $\boldsymbol{N_{seg}} = 30$, $\boldsymbol{N_{MDC}}$=9, and $\boldsymbol{ISDV_A}$=4Mb.

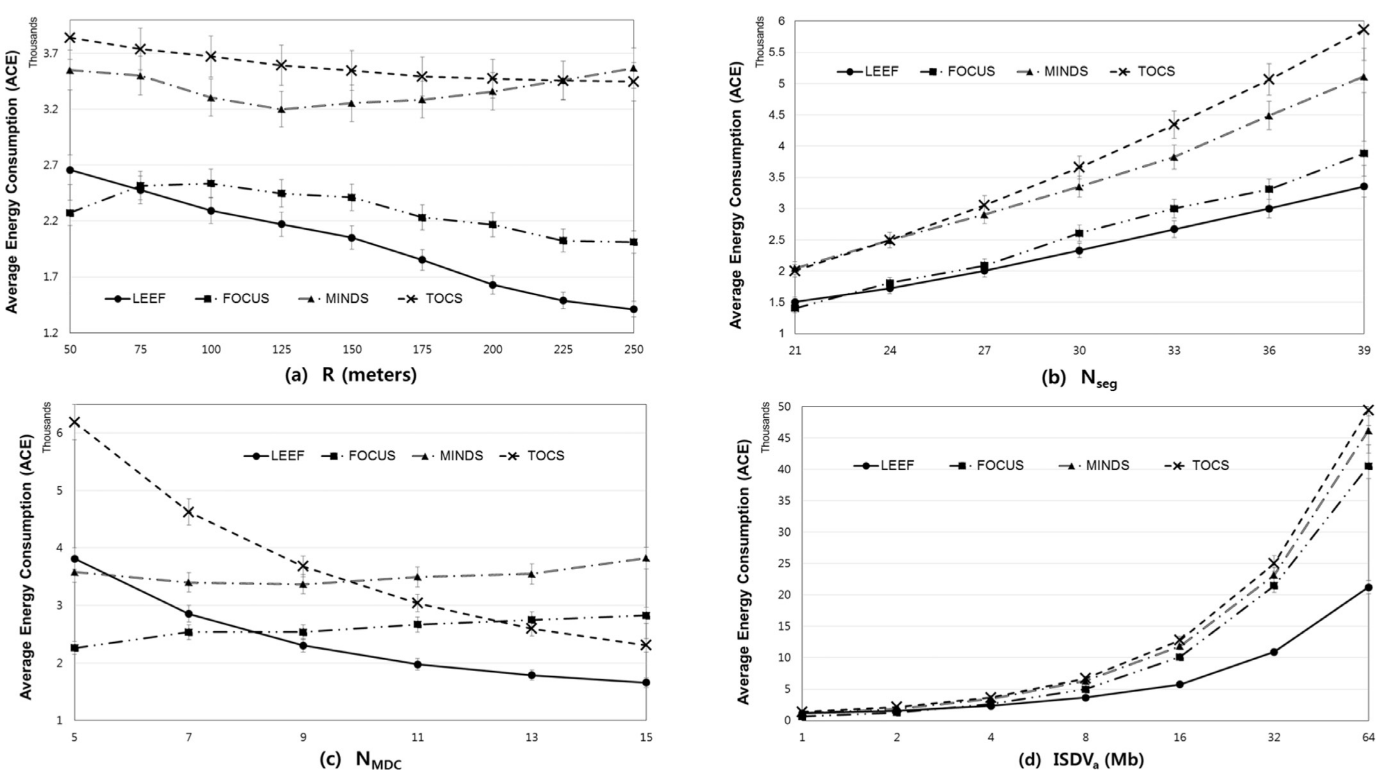

Fig. 9. The performance comparison of LEEF in terms of the average energy consumption of MDCs with a fixed value of R=100, $\boldsymbol{N_{seg}}$=30, $\boldsymbol{N_{MDC}}$=9, and $\boldsymbol{ISDV_A}$=4Mb. (a), (b), (c) and (d) show the results with respective to various R, $\boldsymbol{N_{seg}}$, $\boldsymbol{N_{MDC}}$, and $\boldsymbol{ISDV_A}$.

delivery latency and energy balance as demonstrated in Figs. 7(a) and 8(a).

It is also worth noting that like FOCUS, MINDS forms energy balanced topology, yet more energy is consumed in MINDS than FOCUS. This is because FOCUS reduces energy consumption for delivering inter-cluster data traffic by grouping segments into k-clusters considering inter-segment proximity before overlapping clusters based on an *mst* of the clusters. Therefore, FOCUS can be the best choice for short communication range ($R = 50$) or low MDC count ($k = 5$) for reducing the total energy overhead of MDCs, as shown in Fig. 9(a) and (c). In Fig. 9(d) LEEF yields the least energy overhead for MDCs regardless of the inter-segment data exchange volume. Accordingly, and based also on the results of Fig. 8(d), it can be concluded that LEEF not only balances the energy overhead, but minimizes it as well.

6. Conclusion

In this paper, we have tackled the problem of federating a set of disjoint WSN segments with a constrained number (k) of mobile relays to provide intermittent inter-segment connectivity. We have proposed LEEF which strives to achieve energy-efficient federation with low inter-segment data delivery latency. LEEF factors in both proximity and data volume to group the segments into k clusters in a star topology where a cluster at the center of the area serves as a hop between each pair of clusters. The simulation results demonstrate that LEEF outperforms competing approaches in terms of the maximum inter-segment data delivery latency, total energy overhead of MDCs, and energy balance among MDCs. The effectiveness of LEEF grows with larger MDC communication ranges, and with increased segment or MDC counts.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT). (No. 2015-0-00183, A Study on Hyper Connected Self-Organizing Network Infrastructure Technologies for IoT Service).

Appendix A. Acronyms and terminologies

Acronyms	Description
LEEF	Latency and Energy Efficient Federation
RN	Relay node
MDC	Mobile data carrier
R	Communication range of a RN
G	A center of the damaged area
S_i	i-th segment
S_T	A set of segments, $\|S_T\| = N_{seg}$
CS_T	A set of cells which covers all segments
VC_i	i-th virtual cluster, VC_k is a hub virtual cluster
C_i	i-th cluster; C_k is a hub-cluster
M_i	MDC serving a cluster C_i
$L(c_i)$	Set of the IDs of the segments whose terminals are reachable from cell c_i.
$access(c_i)$	Number of accessible terminals from c_i, i.e., $\|L(c_i)\|$,
$NBR(c_i)$	Set of neighboring cells of c_i
$1hop(c_i)$	Number of terminals that are reachable from c_i via one of the neighboring cells
$prox(c_i)$	Proximity to the center of the area G from c_i
D_{Trip}	Trip latency
D_{Comm}	Data transmission latency
D_{Relay}	Inter-cluster relaying latency
$Speed_{MDC}$	Speed of an MDC
BW_{MDC}	Wireless bandwidth of an MDC
$Data(S_s,S_d)$	Volume of data exchanged between S_s and S_d
$Data_{M_i}$	Total volume of data carried by M_i
$D(S_s,S_d)$	Total delay incurred while delivering $Data(S_s, S\ d)$ between S_s and S_d
$TR(M_i)$	Shortest path that M_i takes while visiting the terminals in C_i
$E_M(M_i)$	Energy overhead of M_i for motion
$E_C(M_i)$	Energy overhead of M_i for wireless communication
$E(M_i)$	Total energy overhead of M_i
$VCID(c)$	ID of VC_i to which a cell $c \in CS_T$ belongs.
$ID(c)$	ID of C_i to which a cell $c \in CS_T$ belongs.
$Anchor(C_i)$	Position where M_i rendezvous with C_k
E_i^r	Total energy overhead of M_i computed in round r
BC_i^r	Selected cell to be included in C_i in round r
$RC(C_i)$	Most recently joined cell in C_i with respect to which a new cell for C_i is selected later.
$Z(C_i)$	Indicator of clustering completion for C_i; Its value is initially 0 and turns 1 as C_i is completely formed.

Appendix B. Pseudo code of LEEF

LEEF(){
1. const $\mu \leftarrow 0.1$; // J/m
2. const $\alpha \leftarrow 10^{-7}$; // J/m
3. const $\beta \leftarrow 10^{-10}$; // $\frac{J}{m^{\partial}}$, $\partial = 2$
4. $S_T \leftarrow$ a set of given terminals;
5. $M \leftarrow$ a set of M_i, i=1…k; // k MDCs
6. $T \leftarrow$ a set of $TR(M_i)$, i=1…k; // Itinerary of M_i
7. $D \leftarrow$ a set of $Data(S_i, S_j)$, $S_i \neq S_j \in S_T$; // Data volume exchanged between S_i and S_j
8. $C_i\ \forall i \leftarrow \emptyset$; // Initialize each set of a cluster
9. $c_g(x_g, y_g) \leftarrow$ Position of the cell at center of the area, G;
// 1st phase: Modeling the area into a cell-based grid
10. Modeling L into a cell-based grid, a side of cell is $R/\sqrt{2}$ in length, where R is a communication range of a sensor;
11. **for** each cell c_i in the grid {// Evaluate the accessibility of c_i
12. $L(c_i) \leftarrow \{S_z \in S_T \mid S_a$ is in R from a center of $c_i\}$;
13. $access(c_i) \leftarrow |L(c_i)|$;
14. $1hop(c_i) \leftarrow |\cup_{c_j \in NBR(c_i)} L(c_j)|$;
15. $prox(c_i) \leftarrow max(|x_i - x_g|, |y_i - y_g|)$;
16. } **end for**
// Identifying the smallest set CS_T of the cells that includes $S_i, \forall i \in S_T$
17. $n_c \leftarrow \min_{n_c}(\cup_{i=1}^{n_c}\{S_a | S_a \in L(c_i)\} = S_T)$;
18. $CS_T \leftarrow \cup_{i=1}^{n_c}\{c_i\}$;
19. **if** multiple sets of CS_T are possible **then**
20. Select the one which has higher value of $access(c_i)$ and $1hop(c_i)$ or lower value of $prox(c_i)$ in order;
21. **end if**
// 1st phase: Forming the virtual clusters
22. A *hub-cluster* $VC_k \leftarrow \{c_g\}$;
23. *Non-hub clusters* $VC_i\ \forall i < k$ *is determined by the same algorithm used in the 1st phase of TOCS;*
// Prior to 2nd phase: Computes E_M & E_C and separates special cases
24. $E_M \leftarrow \sum_{\forall i} E_M(M_i)$ based on VCs using a formula (3);
25. $E_C \leftarrow \sum_{\forall i} E_C(M_i)$ using a formula (4);
26. **if** $E_M >> E_C$ or $E_M << E_C$ **then** {
27. *Handle_SpecialCases();*
28. } **end if**
// 2nd phase: *Greedy-Expansion* step
29. $C_k \leftarrow VC_k$; $C_i \leftarrow C_k \cup \{c_{rt} | \min_{c_{rt} \in VC_i} CellDist(c_i, c_g)\}$;
30. $RC(C_i) \leftarrow c_{rt}$;
31. $Z(C_i), \forall i \leftarrow 0$; $ID(c), \forall c \in CS_T \leftarrow NOT\_CLUSTERED$;
32. **for** each round r {
33. **if** $Z(C_i) == 1$ for $\forall i$ **then** break; // This step ends.
34. **end if**
35. $C_l \leftarrow$ Find the cluster that requires the least energy;
36. **if** $C_l == C_k$ **then {**
37. **if** M_k stayed at G in a round (r-1) **then** {
38. Relocate M_k to $c \leftarrow \min_{c \in CS_T} CellDist(c, G)$;
39. } **else** {
40. $C_k \cup= \{BC_i^r \leftarrow \min_{c \in CS_T} CellDist(c, RC(C_i))\}$;
41. **} end if**
42. Update $Anchor(C_i)$ for $\forall i$;
43. } **else if** C_l is a non-hub cluster C_i {
44. **if** $Z(C_i) == 1$ **then**
45. $C_i \leftarrow$ the next least energy consumed cluster;
46. Go to line 36;
47. **end if**
48. **if** $\exists c_i \in VC_i$, $ID(c_i)$ is $NOT\_CLUSTERED$ **then {**
49. $BC_i^r \leftarrow \min_{c \in VC_i} CellDist(c, RC(C_i))$ whose $ID(c)$ is $NOT\_CLUSTERED$;
50. } **else if** $Z(C_{VCID(c_{nbr})}) \neq 1$ **then** {// searching c_{nbr}'s in a neighboring VC
51. $BC_i^r \leftarrow$ First found $NOT\_CLUSTERED$ cell c_{nbr}, $|VCID(c_{nbr}) - VCID(RC(C_i))| = 1$ during searching the squared area centered at $RC(C_i)$ whose size increases one cell each;
52. } **else {**
53. $Z(C_i) \leftarrow 1$;
54. } **end if**
55. $C_i \cup= \{BC_i^r\}$;
56. $RC(C_i) \leftarrow BC_i^r$;
57. } **end if**
58. *Compute_Tour*(i) for $\forall i$; Update E_i^r for $\forall i$;
59. } **end for**
// 2nd phase: *Optimization* (*re-clustering*) step
60. $r \leftarrow 0$; // The step operates in rounds.
61. $SD^0 \leftarrow$ Standard Deviation of $E_M(M_i)$ for $\forall i$;
62. **do** {
63. $C_l \leftarrow \{C_i \mid \min_{\forall i} E(M_i)\}$; $C_m \leftarrow \{C_i \mid \max_{\forall i} E(M_i)\}$;

(*continued on next page*)

```
    if C_k == C_l then {
        C_m −= {c_in ← min_{c_in ∈ C_k, c ∈ C_m} CellDist(c_in, c))};
        C_k ∪= {c_in}; ID(c_in) ← k;
    } else if C_k == C_m then {
        C_l ∪= {c_out ← min_{c_out ∈ C_k, c ∈ C_l} CellDist(c_out, c))};
        C_k −= {c_out};
        ID(c_out) ← l;
    } else if {
        C_l ∪= {c_out ← min_{c_out ∈ C_k, c ∈ C_l} CellDist(c_out, c))};
        C_m −= {c_in ← min_{c ∈ C_k, c_in ∈ C_m} CellDist(c_in, c))};
        C_k ∪= {c_in} − {c_out};
        ID(c_out) ← l; ID(c_in) ← k;
    } end if
    Update Anchor(C_i) and E_i^r for ∀i;
    Compute_Tour(i) for ∀i;
    SD^r ← Standard Deviation of E_M(M_i) for ∀i;
  } while ( SD^r < SD^{r−1} )
}
int CellDist( c_s( x_s, y_s), c_d( x_d, y_d)) {
return max(|x_s − x_d|, |y_s − y_d|);
}
int Compute_Tour(i) {
    return TR( M_i) ← Tour found by some heuristics, e.g., [23,34,35], for solving the travelling salesman problem with a radio range;
}
```

References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (2002) 393–422.

[2] M. Younis, I. Senturk, K. Akkaya, S. Lee, F. Senel, Topology management techniques for tolerating node failures in wireless sensor networks: a survey, Comput. Netw. 58 (2014) 254–283.

[3] S. Khan, L. Bölöni, D. Turgut, Bridge protection algorithms - A technique for fault-tolerance in sensor networks, Ad Hoc Netw. 24 (2015) 186–199.

[4] G. Lin, G. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, Inf. Process.Lett. 69 (1999) 53–57.

[5] A. Efrat, S.P. Fekete, P.R. Gaddehosur, J.S.B. Mitchell, V. Polishchuk, J. Suomela, Improved approximation algorithms for relay placement, in: Proc. of the 16th European Symposium on Algorithms, Karlsruhe, Germany, 2008.

[6] X. Cheng, D.-z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, Wireless Netw. 14 (3) (2008) 347–355.

[7] E.L. Lloyd, G. Xue, Relay node placement in wireless sensor networks, IEEE Trans. Commun. 56 (1) (2007) 134–138.

[8] F. Al-Turjman, H. Hassanein, M. Ibnkahla, Optimized relay placement to federate wireless sensor networks in environmental applications, in: Proc. of the IEEE International Workshop on Federated Sensor Systems (FedSenS11), Istanbul, Turkey, 2011.

[9] S. Lee, M. Younis, Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree, J. Parallel Distrib. Syst. 70 (5) (2010) 525–536.

[10] F. Senel, M. Younis, Relay node placement in structurally damaged wireless sensor networks via triangular Steiner tree approximation, Elsevier Comput. Commun. 34 (16) (2011) 1932–1941.

[11] J.M. Lanza-Gutiérrez, J.A. Gómez-Pulido, M.A. Vega-Rodríguez, J.M. Sánchez-Pérez, Relay node positioning in wireless sensor networks by means of evolutionary techniques, in: Proc. of the 3rd International Conference on Autonomous and Intelligent Systems (AIS 2012), Aveiro, Portugal, 2012, pp. 18–25.

[12] Y.T. Hou, Y. Shi, H.D. Sherali, On energy provisioning and relay node placement for wireless sensor networks, IEEE Trans. Wireless Commun. 4 (5) (2005) 2579–2590.

[13] Z. Cheng, M. Perillo, W.B. Heinzelman, General network lifetime and cost models for evaluating sensor network deployment strategies, IEEE Trans. Mobile Comput. 7 (4) (2008) 484–497.

[14] X. Han, X. Cao, E.L. Lloyd, C.-C. Shen, Fault-tolerant relay nodes placement in heterogeneous wireless sensor networks, IEEE Trans. Mobile Comput. 9 (5) (2010) 643–656.

[15] F. Senel, M. Younis, K. Akkaya, Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks, IEEE Trans. Veh. Technol. 60 (4) (2011) 1835–1848.

[16] S. Misra, S. Dong, G. Xue, J. Tang, Constrained relay node placement in wireless sensor networks: formulation and approximations, IEEE/ACM Transactions on Networking 18 (2) (2010) 434–447.

[17] D. Yang, S. Misra, X. Fang, G. Xue, J. Zhang, Two-tiered constrained relay node placement in wireless sensor networks: efficient approximations, in: *Proc. of the IEEE Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks. (SECON 2010)*, Boston, MA, 2010.

[18] H. Almasaeid, A.E. Kamal, Data delivery in fragmented wireless sensor networks using mobile agents, in: *Proc. 10th ACM/IEEE Int'l Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2007.

[19] H. Almasaeid, A.E. Kamal, Modeling mobility-assisted data collection in wireless sensor networks, in: *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '08)*, 2008.

[20] K. Li, K.A. Hua, Mobile data collection networks for wireless sensors, in: Proc. of the 5th International Conference on Multimedia Communications, Services and Security (MCSS 2012), Krakow, Poland, 2012, pp. 200–211.

[21] W. Wang, V. Srinivasan, K.C. Chua, Extending the lifetime of wireless sensor networks through mobile relays, IEEE/ACM Trans. Netw. 16 (5) (2008) 1108–1120.

[22] M. Di Francesco, S. Das, G. Anastasi, Data collection in wireless sensor networks with mobile elements: a survey, ACM Trans. Sensor Netw. 8 (1) (2011) Article #7.

[23] F. Senel, M. Younis, Optimized interconnection of disjoint wireless sensor network segments using k mobile data collectors, in: *Proc. of the Int'l Conf. on Comm. (ICC'12)*, Ottawa, Canada, 2012.

[24] J.L.V.M. Stanislaus, M. Younis, Delay-conscious federation of multiple wireless sensor network segments using mobile relays, in: *Proc. of the 76th IEEE Vehicular Technology Conference (VTC2012-Fall)*, Québec City, Canada, 2012.

[25] Y.K. Joshi, M. Younis, Mobility-based internetworking of disjoint segments, in: *Proc. pf 2014 27th Biennial Symposium on Communications (QBSC)*, Kingston, Canada, 2014, pp. 193–197.

[26] A. Abbas, M. Younis, Establishing connectivity among disjoint terminals using a mix of stationary and mobile relays, Comput. Commun. 36 (13) (2013) 1411–1421.

[27] Y.K. Joshi, M. Younis, Restoring connectivity in a resource constrained WSN, J. Netw. Comput. Appl. 66 (2016) 151–165.

[28] I.F. Senturk, K. Akkaya, Mobile data collector assignment and scheduling for minimizing data delay in partitioned wireless sensor networks, in: International Conference on Ad Hoc Networks, 2013, pp. 15–31.

[29] B. Kalyanasundaram, M. Younis, Using mobile data collectors to federate clusters of disjoint sensor network segments, in: *Proc. of Int'l Conf. on Comm. (ICC'13)*, Budapest, Hungary, 2013.

[30] J.L.V.M. Stanislaus, M. Younis, Mobile relays based federation of multiple wireless sensor network segments with reduced-latency, in: *Proc. of Int'l Conf. on Comm. (ICC'13)*, Budapest, Hungary, 2013.

[31] D.K. Goldenberg, J. Lin, A.S. Morse, B.E. Rosen, Y.R. Yang, Towards mobility as a network control primitive, in: Proc of the 5th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2004, pp. 163–174.

[32] V. Rodoplu, T. Meng, Minimum energy mobile wireless networks, in: Proc.of IEEE International Conference on Communications, Atlanta, GA, 1998.

[33] Q. Yang, J. McPeek, A. Nofsinger, Efficient and effective practical algorithms for the set-covering problem, in: CSC, 2008, pp. 156–159.

[34] A. Kashyap, S. Khuller, M. Shayman, Relay placement for higher order connectivity in wireless sensor networks, in: Proc. of IEEE INFOCOM, Barcelona Spain, 2006.

[35] S. Lee, M. Younis, M. Lee, Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance, Ad Hoc Netw. 24 (2015) 1–19.

[36] R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, A Erkmen, Chapter Search and Rescue Robotics, in: Springer Handbook of Robotics, Springer Verlag, 2008, pp. 1151–1173.

[37] K. Iagnemma, D. Golda, M. Spenko, S. Dubowsky, Experimental study of high--speed rough-terrain mobile robot models for reactive behaviors," *Experimental robotics VIII*, in: B. Siciliano, P. Dario (Eds.), Springer Tracts in Advanced Robotics, 5, Springer, Berlin, Heidelberg, 2003.

Sookyoung Lee is currently a research professor in the department of computer science and engineering at the Ewha Womans University, Korea. She received her B.S. and M.S. degrees in Computer Science from the Ewha Womans University, Korea in 1995 and 1997 respectively and Ph.D. degree in Computer Science from the University of Maryland, Baltimore County, USA in 2010. She has been with LG ELECTRONICS Inc., Electronics and Telecommunications Re-search Institute, Korea Electrics Technology Institute, and Samsung Electronics Co. LTD, Korea from 1998 to 2004. While at LG, she has developed the IP data server over ATM switch and implemented virtual private network service for multiprotocol label switching system. She was a volunteer of IPv6 forum Korea while at ETRI and has developed the network address and protocol translation system between IPv4 and IPv6. At Samsung, she was a broadband convergence network designer especially focusing on requirements for QoS and IPv6. Dr. Lee's primary research interest includes network architectures and protocols, topology restoration and fault tolerance in wireless sensor networks and network modeling and performance analysis for dynamic and sparse ad-hoc networks.

Mohamed Younis is currently an associate professor in the department of computer science and electrical engineering at the university of Maryland Baltimore County (UMBC). He received his Ph.D. degree in computer science from New Jersey Institute of Technology, USA. Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International Inc. While at Honeywell he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication and distributed real-time systems. He has published over 240 technical papers in refereed conferences and journals. Dr. Younis has seven granted and three pending patents. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. Dr. Younis is a senior member of the IEEE and the IEEE communications society.

Meejeong Lee received her B.S. degree in Computer Science from Ewha Womans University, Seoul, Korea in 1987, M.S. degree in Computer Science from University of North Carolina, Chapel Hill in 1989, and Ph.D. degree in Computer Engineering from North Carolina State University, Raleigh, in 1994. In 1994, she joined the Department of Computer Science and Engineering at Ewha Womans University, Seoul, Korea, where she is currently a professor. She has been engaged in research in the field of computer communications and networks, and performance modeling and evaluation. Her current research interests focus on protocols and architectures of mobile and wireless networks and future Internet.