IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 25, 2020, accepted September 27, 2020, date of publication September 30, 2020, date of current version October 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027821

LOAF: Load and Resource Aware Federation of Multiple Sensor Sub-Networks

SOOKYOUNG LEE[1], MOHAMED YOUNIS[2], (Senior Member, IEEE), MOHAMMED ALSOLAMI[2], AND MEEJEONG LEE[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea
[2]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

Corresponding author: Meejeong Lee (lmj@ewha.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant 2018R1D1A1B07043671.

ABSTRACT Wireless sensor networks (WSNs) serving in hostile environments are susceptible to multiple collocated failures due to explosives, and natural calamities such as avalanches, landslides etc., which could partition the network into disjoint segments. A similar scenario is encountered when autonomous WSNs need to collaborate for achieving a common task; therefore, federating the segments or individual WSNs would be essential for sharing data between them. 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 that makes the problem more challenging. We exploit the use of a limited number of mobile relays to provide intermittent inter-segment connectivity. We propose LOAF, a novel algorithm for LOad and resource Aware Federation of multiple sensor sub-networks. LOAF strives to group the segments into multiple clusters considering the amount of energy consumed to support the inter-segment traffic as well as proximity between segments. The formed inter-cluster topology is star-shaped where the segments in each cluster are served by a distinct mobile relay and a central cluster provides inter-cluster data delivery. During forming a central cluster, LOAF opts to balance the load on the individual mobile relays in terms of energy consumed in travel and in wireless communication. We analyze the properties of LOAF mathematically and validate its performance through extensive simulation experiments.

INDEX TERMS Intermittent connectivity, federation, mobile data carrier, network partitioning, topology repair, wireless sensor networks.

I. INTRODUCTION

WSNs have attracted increased interest in recent years due to their numerous civil, scientific and military applications. In a WSN, 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 a major damage which results in simultaneous failure of multiple collocated nodes and causes the network to get partitioned into disjoint segments. For example, in a battlefield, parts of the deployment area may be attacked by explosives and some nodes get destroyed. Restoring the connectivity among segments is essential for enabling full network operation. Another scenario is when autonomous networks are to be federated in order to aggregate their capabilities and achieve an emerging task such as search-and-rescue, military situation awareness, criminal hunting, etc.

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

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 segments S into multiple groups or clusters. This is mapped into a set cover problem which is NP-hard. Therefore, we pursue heuristics. While grouping segments in multiple clusters, we opt to balance the energy consumed by the MDCs for touring the segments of the individual clusters and uploading/downloading data over wireless links. We propose a novel algorithm for LOad and resource Aware Federation of multiple sensor sub-networks (LOAF) by forming a k-clustered topology where the energy consumption overhead in touring and data transporting is balanced among the k MDCs. LOAF consists of two phases.

In the 1st phase, LOAF first tries to find the energy-based center of mass of S while considering the inter-segment data volume, hereafter denoted as eG. Basically, to calculate eG, LOAF considers the in/out data volume for each segment S_i and calculates the corresponding energy consumption for up/downloading data at S_i via wireless communication. Based on the found eG, LOAF opts to form $(k-1)$ *non-central-clusters* C_i's around eG while minimizing the overall energy consumed in visiting the segments of C_i, $i < k$ by an MDC_i. During the 2nd phase, LOAF opts to form a *central-cluster* C_k around eG while balancing energy consumption among MDC_i's, $\forall i$ in terms of travelling a cluster and transporting data. The clustering results in a star-shaped inter-cluster topology via which every pair of segments in different clusters can exchange data at most within two cluster-hops. LOAF is validated through simulation experiments and is shown to outperform competing schemes.

The rest of the paper is organized as follows. LOAF is compared to related work in Section II. In Section III, the problem is formally defined and the considered system model is described. The details of LOAF are provided in Section IV. The validation results are presented in Section V. The paper is finally concluded in Section VI.

II. RELATED WORK

Published techniques for federating the distinct segments or autonomous WSNs can be categorized into two groups [2], [3]: (i) approaches that employ relay nodes and form stable data paths 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.

A. 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 the 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 restoring connectivity in a partitioned network [5], [8]. In addition to reducing the number of relays, the second group considers more objectives, such as high energy efficiency to prolong network lifetime [9], [10], or high degree of connectivity by providing multiple disjoint inter-node paths in the formed network [11], [14]. Solutions in the third group tackle the federation problem under node position constraints [15], [16] or various quality of services [17], [18]. Overall, the approaches in this category provide perpetual connectivity and do not consider resource-availability constraints. LOAF solves the federation problem when the relay count is insufficient for forming stable data paths and establishes intermittent connectivity instead.

B. 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 mobile data carrier (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 [20]–[23]. For instance, in [20] Almasaeid *et al.* 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 [21] by including two other roles of mobile agents, i.e., mobile sinks and mobile collectors. On the other hand, Li and Hua deploy mobile nodes to form a second-tier mesh network to transfer data from sensor nodes to a sink such that the data delivery latency due to buffer space limitation of the mobile nodes is minimized. Unlike LOAF, the approaches introduced in this group do not consider the limited supply of mobile nodes and thus careful assignment of MDCs is not studied. In addition, Wang *et al.* [22] and Ma [23] have focused on a single MDC setup and how to limit the length of data paths. However, such work does not deal with segments of nodes and address inter-segments connectivity.

Like LOAF, the second category solves the federation problem with a limited MDC count. IDM-kMDC [24], FeSMoR [25], MINDS [26] and LEEF [27] 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 and LEEF strive to even tour length among MDCs and is compared to LOAF in Section V. 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 [28], [29]. However, unlike LOAF 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. I. F. Senturk *et al.* propose a delay-aware clustering algorithm which groups segments for k MDCs to reduce the maximum tour length among MDCs and data delivery delay between segments to a sink [30]. In addition, FOCUS factors in the data delivery delay in the federation by carefully scheduling the motion of MDCs [31]. It first forms overlapped k clusters served by k MDCs. The intersection segment (IS) of two overlapping clusters plays the role of a rendezvous point. In order to avoid storing data and holding MDCs at the ISs, FOCUS tries to synchronize the arrival of two MDCs by carefully setting their motion speed. However, FOCUS does not care for the data delivery delay and energy concerns.

Meanwhile, ToCS pays more attention to reducing the average and maximum delay for delivering data between segments by finding the balanced tour paths among MDCs [32]. Like LOAF, ToCS groups a set of segments into k clusters that form a star topology in order to reduce the inter-segment latency. Then it opts to equalize the tour length of clusters 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, LOAF strives to even the energy overhead for each MDC to tour and transport a distinct volume of data between the terminals which it serves. The consumed energy and the communication delay of LOAF are compared to that of MINDS, LEEF, ToCS, and FOCUS in Section V.

III. PROBLEM STATEMENT

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.

The federation is to be achieved using k MDCs that provide intermittent data paths among the segments subject to the data volume requirement. The main optimization objective for such a 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 of n distinct terminals $\{S_1, S_2, \ldots S_n\}$, $n > 2$ which play 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 LOAF, S is to be clustered into C_i's, $i = 1,.., k$, i.e., $C_i \subset S$ and $\bigcup_{\forall i} C_i = S$. 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 inter-segment data between 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.

$$\begin{aligned} D\left(S_s, S_d\right) &= D_T\left(Trip_{s,d}\right) + \sigma D_C(Data_{s,d}) + D_R\left(wt_{s,d}\right) \\ &= \left(\frac{1}{V_M} \cdot Trip_{s,d}\right) + \left(\frac{\sigma}{B_M} \cdot Data_{s,d}\right) + \left(wt_{s,d}\right) \end{aligned} \tag{1}$$

$$\begin{aligned} E\left(M_i\right) &= E_M(TR(M_i)) + E_C\left(Data_{M_i}\right) \\ &= (\mu \cdot TR(M_i)) + \left(Data_{M_i} \cdot P_C(R)\right) \end{aligned} \tag{2}$$

In (1), the delay incurred while carrying data from S_s to S_d is primarily determined by a trip latency D_T, a data transmission latency D_C, and a relaying latency D_R. D_T 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 V_M for all MDCs and D_T is thus a function of only the tour length of the individual MDC. In addition, D_C 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 B_M in (1), respectively. Recall that LOAF forms a star-like inter-cluster topology; thus, inter-cluster relaying may take place. The constant σ captures that fact that the communication delay is incurred more than once since two MDCs will be involved in exchanging 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 central 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 based on the size of the corresponding segments.

Lastly, D_R means the loading 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 the proposed solution MDCs meet directly to exchange data at a computed position and the inter-cluster data relaying may thus involve the waiting time for rendezvous between MDCs depending on their trip schedule. Therefore, like D_C, D_R is computed depending on the locations of the communicating segment pair S_s and S_d. In case S_s and S_d belong to the same cluster C_i, i.e., both segments are served by the same MDC M_i, D_R becomes zero. Otherwise, the extra rendezvous time 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, i.e., $E_M(M_i)$, and the ancillary energy for communication, i.e., $E_C(M_i)$. 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 upload/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 a trip distance of M_i serving C_i, i.e., $TR(M_i)$ in meters while E_C is dependent on the volume of data, $Data_{M_i}$, uploaded/downloaded over a wireless link, i.e.,

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

We assume using an energy cost model for E_M that is proportional to the 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).

$$E_M\left(M_i\right) = \mu \cdot TR(M_i) \quad (3)$$

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

In (3), μ ranges from 0.1 to 1 J/m [33]. In addition, the energy required to transmit 1 bit is $2 \cdot 10^{-6}$ Joule, where $\alpha = 100\text{nJ}$, $\beta = 0.1\text{nJ}/m^{\partial}$, $\partial = 2$ in (4) [34]. During the federation, LOAF strives to balance $E(M_i)$'s, $\forall i$ 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:

$$\begin{aligned} &\textit{Find a set of } C_i, \quad i = 1, 2, \ldots k, \\ &\quad \textit{where } \bigcup\nolimits_{\forall i} \boldsymbol{C_i} = S \textit{ such that} D(S_p, S_q) \textit{ and} \\ &\left(\sum\nolimits_{i \neq j}^{\forall C_i, C_j} \left|E\left(M_i\right) - E\left(M_j\right)\right|\right) \textit{ are minimized} \end{aligned} \quad (5)$$

The problem presented in formula (5) is to find a set of clusters C_i's that covers S, each of which 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 suppose that all MDCs have the same capabilities with enough data storage to handle the inter-segment data traffic in each cluster, i.e., we do not consider any buffering constraints in our solution. 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 LOAF 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 segments can be done by land-based 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 detail in the next section. LOAF 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 inter-networking 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 r, i.e., $R = r$. This is a simplifying assumption to ease the presentation. In addition, coverage is not the focus of LOAF although MDCs may have sensing capabilities that can mitigate coverage loss caused by the damage during their tours. Finally, we assume that the data volume generated by sensors, i.e. sampling rate, is fixed and determined by various application-specific missions [38], [39]; thus the communication load between segments stays constant during network operation. Nonetheless, we study the effect of variation in the data volume in Section V.

IV. THE LOAF APPROACH

To provide energy balanced federation of the n terminals using k MDCs, LOAF groups the terminals into a set of k clusters in a star inter-cluster topology, where a *central-cluster* C_k serves for data relaying between pairs of clusters. In the first phase, LOAF computes eG, i.e., an energy-based center of mass, of a set of segments S based on the inter-segment related communication energy which is proportional to traffic volume exchanged between segments. Then it strives to form *non-central-clusters* C_i's $i<k$ around eG while minimizing the overall energy consumption in the clusters in terms of motion. In the second phase, LOAF opts to form a central cluster around eG while balancing energy among C_i's, $\forall i$. The details of LOAF are provided in the balance of this section.

A. 1st PHASE: FORMING NON-CENTRAL CLUSTERS CONSIDERING HETEROGENEOUS INTER-SEGMENT DATA VOLUME

LOAF first tries to find the center of mass eG of S considering energy consumption required for the inter-segment

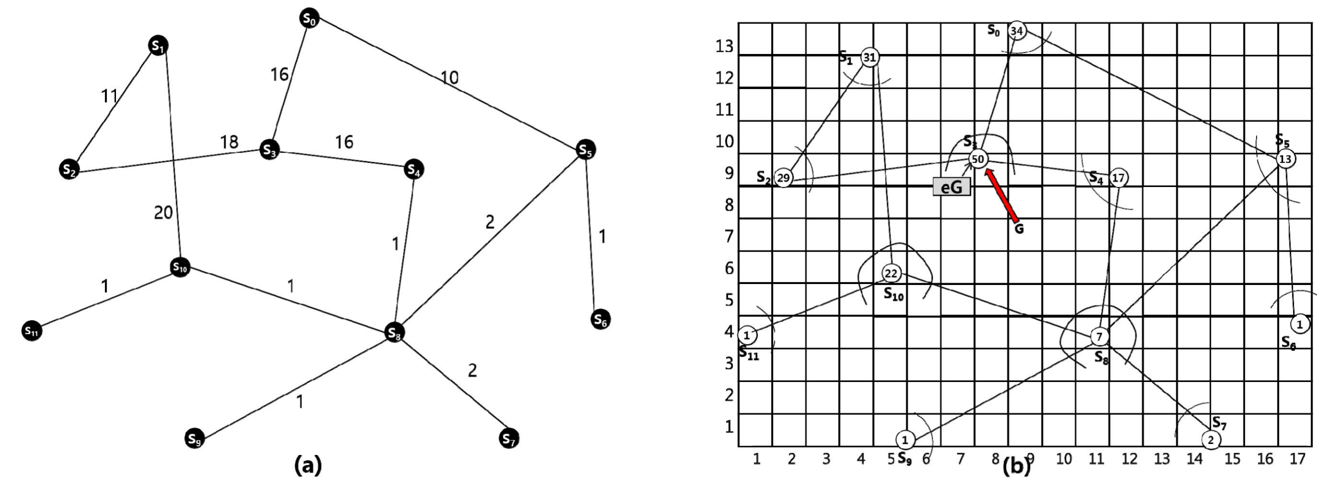

FIGURE 1. In (a) the number on an edge represents data volume exchanged between two segments in megabits. In order to intermittently connect 12 segments by *k* MDCs, LOAF first computes the weighted core of mass *eG* of 12 segments with regards to the energy consumption considering inter-segment data volume. The numbers written in each circle shown in (b) represent the aggregated amount of inter-segment data traffic involved to each segment and *eG* is thus determined towards the circles which have larger numbers.

data transportation. Then non-central clusters, C_i's $i < k$ are formed around the found *eG* while minimizing energy experienced while MDCs travel the formed non-central clusters. The details of the first phase are provided below.

1) COMPUTING AN ENERGY-CONSIDERED CORE OF MASS

Prior to grouping segments into clusters, LOAF first tries to find the appropriate location for a *central-cluster* C_k via which the inter-segment data traffic between clusters is forwarded in a star-shaped cluster topology. Since the objective of LOAF is to balance energy consumed by MDCs that serve clusters, the ideal location of the central cluster ought to be at the center of groups of segments with respect to energy consumption of the MDCs, i.e., a sum of $E_M\ (M_i)$ and $E_C\ (M_i)$ each of which is mainly affected by data traffic volume between segments and the inter-segment proximity, respectively. In other words, LOAF finds the center of mass of segments i.e., *eG*, considering data volume as well as proximity.

Therefore, for computing $eG(c_x,\ c_y)$ LOAF first calculates the total data volume coupled with each segment S_i, i.e., a sum of traffic originated from/destined to S_i which is hereafter denoted as $Data_{S_i}$ that is equal to $\sum_{\forall S_x \in S_T, S_i \neq S_x} Data\ (S_i, S_x)$. Then the x- and y- coordinates, c_x and c_y are calculated in the same way to find a center of mass of the weighted vertices. In other words, c_x and c_y equal $\frac{\sum_{\forall i} x(S_i) \cdot Data_{S_i}}{\sum_{\forall i} Data_{S_i}}$, and $\frac{\sum_{\forall i} y(S_i) \cdot Data_{S_i}}{\sum_{\forall i} Data_{S_i}}$, respectively, where $x(S_i)$ and $y(S_i)$ represents x- and y- coordinate of a segment S_i. Figure 1-(b) shows the found *eG* assuming the area of interest is $270 \times 360\text{m}^2$ and the cell width is corresponding to $\frac{R}{\sqrt{2}}$, $R = 30$m. The *eG* is located in a different position from the centroid *G* computed only considering locations of segments. In the figure, the number in a circle of S_i represents $Data_{S_i}$ in megabits based on the numbers on an edge $e(S_i, S_j)$ that means the volume of data that is exchanged between S_i and $S_j i \neq j$ in Figure 1-(a). In consideration of $Data_{S_i}$'s, the location of *eG* moves towards the position where the heavy data exchange overhead is involved like around S_3.

2) FORMING NON-CENTRAL CLUSTERS

For federating the segments $S_i \forall i\ \in S$ with k MDCs in an energy balanced manner, LOAF tries to group the S_i's in a star inter-cluster topology, where each cluster is served by a distinct MDC and the inter-segment data delivery can be operated within at most two cluster hops. Then, LOAF tries to equalize the energy overhead of MDCs in adjacent clusters. In order to reduce the overall energy consumed in the clusters by the MDCs, LOAF opts to group segments to minimize energy overhead experienced by each MDC serving C_i, $i < k$. Therefore, the segments are grouped into $(k-1)$ non-central clusters in rounds during which LOAF strives to reduce the sum of $E\ (M_i)\ \forall i$, i.e., $\sum_{\forall i} E(M_i)$ in each round. Initially we assume that a central cluster C_k is formed by placing an MDC M_k at *eG* as a stationary relay and includes the segments within a radio range of M_k that is R, i.e., $C_k = \{S_i |\ EuclideanDist(S_i, eG) \leq R\ \}$.

In the first round ($r = 0$), for each $S_i \in (\text{S}-C_k)$ an individual cluster $C_i = \{S_i, eCoM\}$, $i = 0, \ldots, N^0_{cluster}$ is initially formed, where $N^0_{cluster}$ equals $N_{seg} - |C_k|$, where N_{seg} is the number of segments. Then, each $E\ (M_i)\ \forall i$ is computed as a sum of $E_M(M_i)$ and $E_C(M_i)$. Since every C_i includes only two elements, *eG* and S_i, $E_C(M_i)$ is mainly determined by $Data_{S_i}$ and $E_M(M_i)$, the length of $TR(M_i)$ which follows the straight path between *eG* and S_i. Thus $|TR(M_i)|$ is equal to $(EuclideanDist(eG,\ S_i) - 2R)$, where $EuclideanDist(a,\ b)$ is an Euclidean distance between a and b. Obviously, it is sufficient for an MDC to be at a distance R from the segments and from M_k in order to establish communication links, assuming free space signal propagation model. In the second round ($r = 1$), LOAF opts to combine two clusters C_x and C_y whose merging leads to the greatest decrease in the overall tour length. This step is performed repetitively. Thus, the number of clusters in round r denoted as $N_{cluster}$ decreases by one in each round. In other words, a pair of C_x and C_y which satisfies the equation (6) is selected in each round $r \geq 1$.

$$\min_{x,y}\left(\sum\nolimits_{i=1,i\neq x,y}^{N_{cluster}^{0}-r} E\left(C_i\right)+ME\left(C_x\cup C_y\right)\right),$$
$$\text{where } ME\ \left(C_x\cup C_y\right)=E_M\left(C_x\cup C_y\right)+E_C\left(C_x\cup C_y\right) \tag{6}$$

In (6), $ME(C_x \cup C_y)$ represents energy required for an MDC to serve the merged cluster that includes C_x and C_y for transporting data and travelling. $E_C\left(C_x \cup C_y\right)$ is mainly affected by the aggregated data volume, i.e., $Data_{S_x} + Data_{S_y}$ and $E_M\left(C_x \cup C_y\right)$, the updated trip that is the shortest path in the merged cluster C_z. Since the goal of this step is to group segments in the form which minimizes $\sum_{\forall i} E_M(M_i)$ and $\sum_{\forall i} E_C(M_i)$ and $\sum_{\forall i} E_C(M_i)$ is already determined by given data transport requirements between segments, LOAF focuses on reducing $\sum_{\forall i} E_M(M_i)$ in the clustering process. Therefore, a pair of C_x and C_y whose merged tour path decreases $\sum_{\forall i} E_M(M_i)$, will be selected.

Computing such a merged path of C_x and C_y is equivalent to the problem which finds the shortest *Hamiltonian cycle* to visit S_i's$\forall i \in (C_x \cup C_y)$ considering R that is NP-hard. Therefore, LOAF uses the heuristic solution in [24], [35], [36] to compute a tour path of C_z. The clustering procedure is repeated until $N_{cluster}$ become $(k-1)$ and the 2nd step of the first phase thus terminates as $r = N_{cluster}^{0} - (k-1)$. We refer to the point P_i on the circle of radius R and centered at eG, where M_k is placed, as the rendezvous points between $M_i i < k$ and M_k. When reaching P_i, M_i will be able to establish communication with M_k and exchange relevant data payload. The position of rendezvous points will be adjusted in the 2nd phase of LOAF as we explain in the next subsection.

Figure 2 shows the formation of non-central clusters using the example seen in Figure 1 as $k = 5$. We assume that R is 30m and the required energy to move along $TR(M_i)$ is computed according to equation (3), where $\mu = 1$ J/m. In addition, 2 Joules are required to transmit 1 megabit over wireless communication according to (4). For these settings, Figure 2 shows that the twelve segments are grouped into C_i, $i = 1, 2, 3, 4$, in 8 rounds. In the 1st rounds ($r = 0$), M_5 is to be placed at eG where S_3 happens to locate and forms $C_5 = \{S_3\}$ and eleven individual clusters are formed, e.g., $C_i = \{S_i, eG\}$ for $i = 0, 1, 2$ and $C_{i-1} = \{S_i, eG\}$ for $i = 4, \ldots, 11$. Then during $r = 1, 2$, and 3, each pair of two clusters, i.e., (C_4, C_5), (C_7, C_8), and (C_9, C_{10}) is combined and becomes one cluster, $\{S_4, S_5, eG\}$, $\{S_7, S_8, eG\}$, and $\{S_9, S_{10}, eG\}$, respectively since merging two clusters reduces the most energy consumed for travelling in each round. The selection of a pair of two clusters to merge is repeated in the subsequent rounds. Finally in the last round ($r = 7$), four non-central clusters $C_0 = \{S_0, S_1, eG\}$, $C_1 = \{S_2, S_9, S_{10}, S_{11}, eG\}$, $C_2 = \{S_6, S_7, S_8, eG\}$, and $C_3 = \{S_4, S_5, eG\}$ are formed with the least amount of energy $\sum_{i=1}^{4} E(M_i)$, where $E(M_1) = 320$, $E(M_2) = 526$, $E(M_3) = 470$, and $E(M_4) = 330$. The pseudo code of the 1st phase is described in Algorithm I.

Algorithm 1 Pseudocode of the 1st Phase of LOAF

$S \leftarrow$ a set of given segments $S_i, \forall i$;
$Data_{S_i} \leftarrow$ a set of $Data(S_i, S_j)$, $S_i \neq S_j \in S$; // Data volume exchanged between S_i and S_j
// Computing a weighted core of mass of S, $eG(c_x, c_y)$
$c_x = \frac{\sum_{\forall i} x\_(S_i) \cdot Data_{S_i}}{\sum_{\forall i} Data_{S_i}}$, $c_y = \frac{\sum_{\forall i} y\_(S_i) \cdot Data_{S_i}}{\sum_{\forall i} Data_{S_j}}$
// forming $(k-1)$ non-central clusters minimizing the overall energy consumption of MDCs
$C_k = \{S_i |\ EuclideanDist(S_i, eG) \leq R\}$;
$S - = C_k$;
$i = 0$
for each $S_j \in (S - C_k)$\{
  $C_i = \{S_j, eG\}$; $i++$;
\} **end for**
$r = 1$; $N_{cluster}^{0} = |S| - |C_k|$; $N_{cluster} = N_{cluster}^{0}$;
do \{
  $C_x, C_y =$
  $\min_{x,y}\left(\sum\nolimits_{i=1,i\neq x,y}^{N_{cluster}^{0}-r} E\left(C_i\right)+ME\left(C_x\cup C_y\right)\right)$,
   where $ME\left(C_x, C_y\right) = E_M\left(C_x \cup C_y\right)$
   $+E_C\left(C_x \cup C_y\right)$
  $C_x \cup = C_y$; $C_y = C_{N_{cluster}}$;
  $r = r + 1$; $N_{cluster} - = 1$;
\} **while** $(r < N_{cluster}^{0} - (k-1))$

B. 2nd PHASE: FORMING A CENTRAL-CLUSTER C_k WHILE BALANCING ENERGY AMONG CLUSTERS

While in the 1st phase LOAF focuses on reducing the overall energy overhead experienced by M_i, $i < k$ in touring non-central clustering, the major objective of the 2nd phase is to equalize energy overhead for all MDCs. Since a central cluster C_k is initially formed by placing M_k at eG as a stationary relay in the 1st phase, we adjust the size of C_k for energy balance during the 2nd phase. In other words, the star inter-cluster topology centered at C_k is re-formed by moving P_i outwards or inwards depending on the relative energy consumption of M_i. This phase consists of two steps. In the first step, C_k is grown by shirking the non-central clusters whose MDCs consume more energy than the average among all MDCs, denoted as E_{AVG}. In other words, the rendezvous point, P_j is moved outwards from eG for a non-central cluster C_j whose designated MDC, M_j, consumes more than E_{AVG}. The step is iterative since the tour length of re-formed clusters will change and subsequently E_{AVG}. The 1st step terminates as M_k consumes energy more than E_{AVG}. Then, the second step is geared to adjust the size of non-central clusters whose MDCs consume less than E_{AVG},

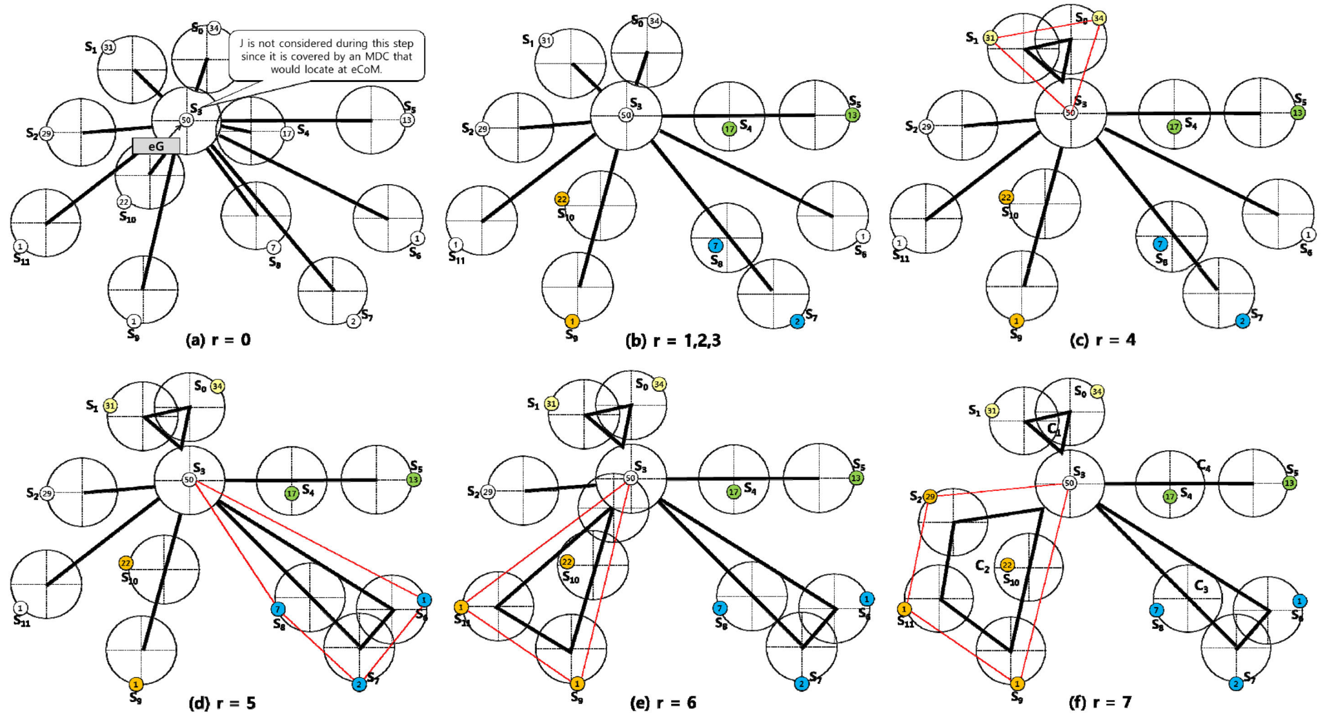

FIGURE 2. In the 2nd step of the 1st phase, LOAF opts to form ($k-1$) non-central clusters, $C_0, \ldots, C_{k-1}$, ($k = 5$) during which the overall energy consumption by MDCs is minimized. The step operates in rounds. In the first round, each segment and *eG* form an individual cluster. Since S_3 is within *R* from *eG*, ($N_{seg}-1$) clusters are formed, where N_{seg} is the number of segments. After that two clusters are combined in each round while minimizing the overall energy experienced by MDCs. The process is repeated until the number of clusters equals ($k-1$) = 4.

where the rendezvous points for these clusters are moved inwards toward *eG*. Both steps are iterative in nature and are performed in rounds. The following notation used in the discussion.

- $\boldsymbol{C_i^r}$: reflects the segments grouped into a cluster C_i in round r. C_k^0 includes segments that are reachable from M_k which is placed at eG i.e., $\{S_i | EuclideanDist(S_i, eCoM) \leq R\}$ and C_i^0's, i < k are the same as the 1st phase.
- $\boldsymbol{P_i^r}$: It represents a rendezvous point where $M_i i < k$ meets M_k in round r. P_i^r's $\forall i$ are always on the straight line between eG and CoM_i^{r-1} which will be explained later.
- $\boldsymbol{TR_i^r}$: The travel path that M_i takes in round r while visiting the segments in C_i is denoted as TR_i^r. It is equivalent to the shortest path where M_i departs from P_i^r and visits every segment $S_i \in C_i$ once and returns to P_i^r. Since finding TR_i^r is mapped into the Travelling Salesman Problem, the heuristic solution of [24], [35], [12] will be used to find TR_i^r.
- $\boldsymbol{CoM_i^r}$: It represents a core of mass of the polygon formed by the segments of C_i and P_i^r in round r. In round r of the 2nd phase of LOAF, C_k may be expanded by moving P_i^r towards CoM_i^{r-1} of C_i or C_j may be extended by relocating its P_j^r towards CoM_k^{r-1}.
- $\boldsymbol{E_i^r}$: denotes the energy experienced by M_i in round r for up/downloading intra-cluster data in C_i and inter-cluster data at P_i^r 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 equation (3) and (4), respectively.
- $\boldsymbol{E_{AVG}^r}$: equals an average value of $E_i^r \forall i$.

With the provided notations, the 2nd phase of LOAF opts to balance the energy overhead by adjusting the MDC tours. Basically, LOAF tries to adjust rendezvous points P_i's of C_i's outwards or inwards based on the average energy overhead. In other words, LOAF relocates P_m outwards towards C_m if $E_m > E_{AVG}$ in order to reduce the heavy energy overhead of C_m or reposition P_l inwards towards *eG*, where C_k is initially formed if $E_l < E_{AVG}$ for expanding C_l. Since the computed *eG* during the 1st phase lies inside the convex hull of the segments, as will be proven in Lemma 1, and the initial energy overhead E_k^0 of forming C_k at the *eG*, is less than E_{AVG}, as will be proven in Lemma 2, (*i*) the rendezvous points P_m's for C_m's whose $E_m^r > E_{AVG}^r$ are moved outward while $E_k^r < E_{AVG}^r$ and then (*ii*) P_l's for C_l's whose $E_l^r < E_{AVG}^r$ are also handled while $E_k^r > E_{AVG}^r$. Each of the steps (*i*) and (*ii*) is repeated in rounds until there is no more improvement of the standard deviation (*SD*) of $E_i \forall i$ as explained below.

(*i*) *moving* P_i'*s outwards*: During this step, LOAF iteratively shrinks the boundary of clusters, C_m's, of heavily loaded MDCs, and accordingly expands C_k. In the first round ($r = 0$), for each cluster C_i, $i = 1, \ldots, k-1$, P_i^0, TR_i^0, and CoM_i^0 are set up based on the intra-cluster topology formed in the 1st phase. Then the initial energy consumption, E_i^0 is computed as a sum of $E_C(M_i)$ and $E_M(M_i)$, e.g., $E_1^0 = 320$,

$E_2^0 = 526$, $E_3^0 = 470$, $E_4^0 = 330$, and $E_5^0 = 194$ in the example seen in Figure 2-(f), and which corresponds to the initial star-shaped cluster arrangement. In particular, E_5^0 reflects only $E_C(M_5)$ which includes energy consumption for transporting the inter-cluster data traffic volume represented using double dotted lines in Figure 3. Therefore, $E_C(M_k)$ increases when high volume of data traffic is delivered between clusters.

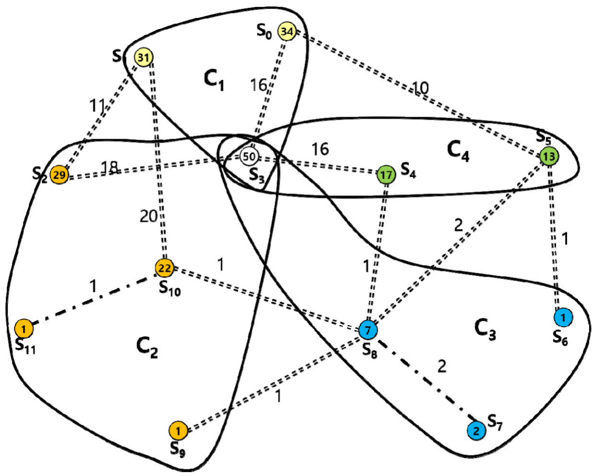

FIGURE 3. In the 1st phase of LOAF, the non-central clusters C_1, C_2, C_3 and C_4 are formed centered at *eG* where M_5 is placed. C_5 that includes segments reachable from M_5 is formed. Thus E_5^0 contains only energy for up/downloading inter-cluster data traffic represented using double dotted lines.

LOAF relocates P_i's away from *eG* starting with clusters whose MDCs consume energy more than average and adjusts their tour accordingly. For the relocation of P_i's, LOAF first determines the relative movement rate φ_i of each P_i based on the energy imbalance among the MDCs, i.e., $\varphi_i = \frac{E_i}{E_{least}}$, where $E_{least} = \min_{\forall j} E_j^0$. In other words, φ_i factors in the relative excessive energy overhead of each M_i to the least energy consuming MDC. In order to adjust the actual moving distance in meters, a constant parameter, ε (≥ 1) is multiplied to φ_i. With excessively large values of ε, the convergence to form the energy-balanced clusters may be fluctuated while small ε slows the convergence of energy balancing. Thus, an appropriate value of ε, should be selected. Since the formed clusters are served by MDCs, the communication range of an MDC, i.e., R is the recommended setting for ε in practice.

Based on φ_i and ε, LOAF expands a size of C_k and accordingly reduces that of C_i by moving P_i away ($\varphi_i \times \varepsilon$) meters along a line from P_i to a core of mass of a polygon formed by segments grouped to C_i in round $r = 0$ denoted as CoM_i^0. After that tour paths of M_k and M_i, i.e., TR_k and TR_i, are recomputed using an updated rendezvous point P_i. Accordingly, E_k, E_i and CoM_i are also updated. In addition, re-grouping of segments may occur, e.g., $S_x \in C_i$ may be included to C_k if S_x becomes reachable from M_k after expanding C_k. Grouping all segments to k clusters is maintained during this process as will be proven in Theorem 1. The same process of C_k expansion is repeated in successive rounds and terminates in round f where M_k consumes energy more than average, i.e., $E_k \geq E_{AVG}$. Therefore, the process is converged in O($\frac{Dist(eG,L)}{\varepsilon}$), where L is the smallest polygon that includes all segments, and *Dist(eG, L)* represents a distance from *eG* to the furthest point on L.

(*ii*) *moving P_l inwards*: This step adjusts P_l's of C_l's, whose serving M_l requires less energy than average, i.e., $E_l < E_{AVG}$. Thus LOAF tries to extend the size of C_l's by moving the P_l's towards C_k, exactly towards *eG* or a core of mass of C_k, i.e., CoM_k in case C_k includes more than one segment. Accordingly, TR_k is shortened. For repositioning of P_l's, LOAF also computes φ'_l which reflects the relative energy shortage of M_l to the least energy consuming MDC and is found in the opposite way of computing φ_i, i.e., $\varphi'_l = \frac{E_{least}}{E_l}$ and then P_l moves inwards ($\varphi'_l \times \varepsilon$) meters along a line from P_l to CoM_k. Using φ'_l, the less energy overhead M_l incurs, the more distance P_l moves inwards towards CoM_k. Thus C_l is accordingly extended and E_l also increases. After that, TR_l, CoM_k, and E_l, E_k and E_{AVG} are updated. The same procedure is repeated until $E_k \leq E_{AVG}$.

Figure 4 shows the procedure of the 2nd phase of LOAF as ε is equal to R. In the first round ($r = 0$) of the C_k extension step, the initial rendezvous points P_i^0's $\forall i < k$ are located at *eG*. Then each C_i relocates ($\frac{E_i^0}{E_5^0} \times R$) meters, $i = 2,3,4$ respectively towards $CoM_i^0 i = 2, 3, 4$ as seen in Figure 4-(b). This is because M_2^0, M_3^0, and M_4^0 consume more energy than E_{AVG}^0. Accordingly, C_5^1 is extended towards C_2^0, C_3^0, and C_4^0 as seen in Figure 4-(c). In addition, the same process is applied to C_2^0, and C_3^0 in a subsequent round as shown in Figure 4-(d). After that S_{10} becomes unreachable to M_2, and accessible to M_5; therefore, S_{10} changes association to C_5. Figure 4-(e) to (h) show the 2nd step. The first round of that step, C_1^0, C_3^0, and C_4^0 grow towards C_5 by adjusting their rendezvous points. Similarly, C_1^0, and C_4^0 are extended in the second round. Finally, the resulting intermittent topology of k ($= 5$) MDCs formed in an energy-balanced manner for federating twelve segments after the first round of the 2nd phase is presented in Figure 4-(i). The pseudo code of the 2nd phase of LOAF is described in Algorithm II. Lines 9-12 describe the first step and lines 23-32 explain the second step. As a result, LOAF returns k sets of segments each of which is served by an individual MDC and their tour paths, i.e., $\{C_i, TR_i \forall i\}$ in line 33.

C. ALGORITHM ANALYSIS

LOAF is analyzed in this subsection. We mainly focus on proving that the energy imbalance among MDCs is minimized by LOAF while forming a star inter-cluster topology. In addition, the complexity of LOAF is analyzed to show that the resulting cluster topology is formed within an execution time bound. We introduce the following theorems and lemmas:

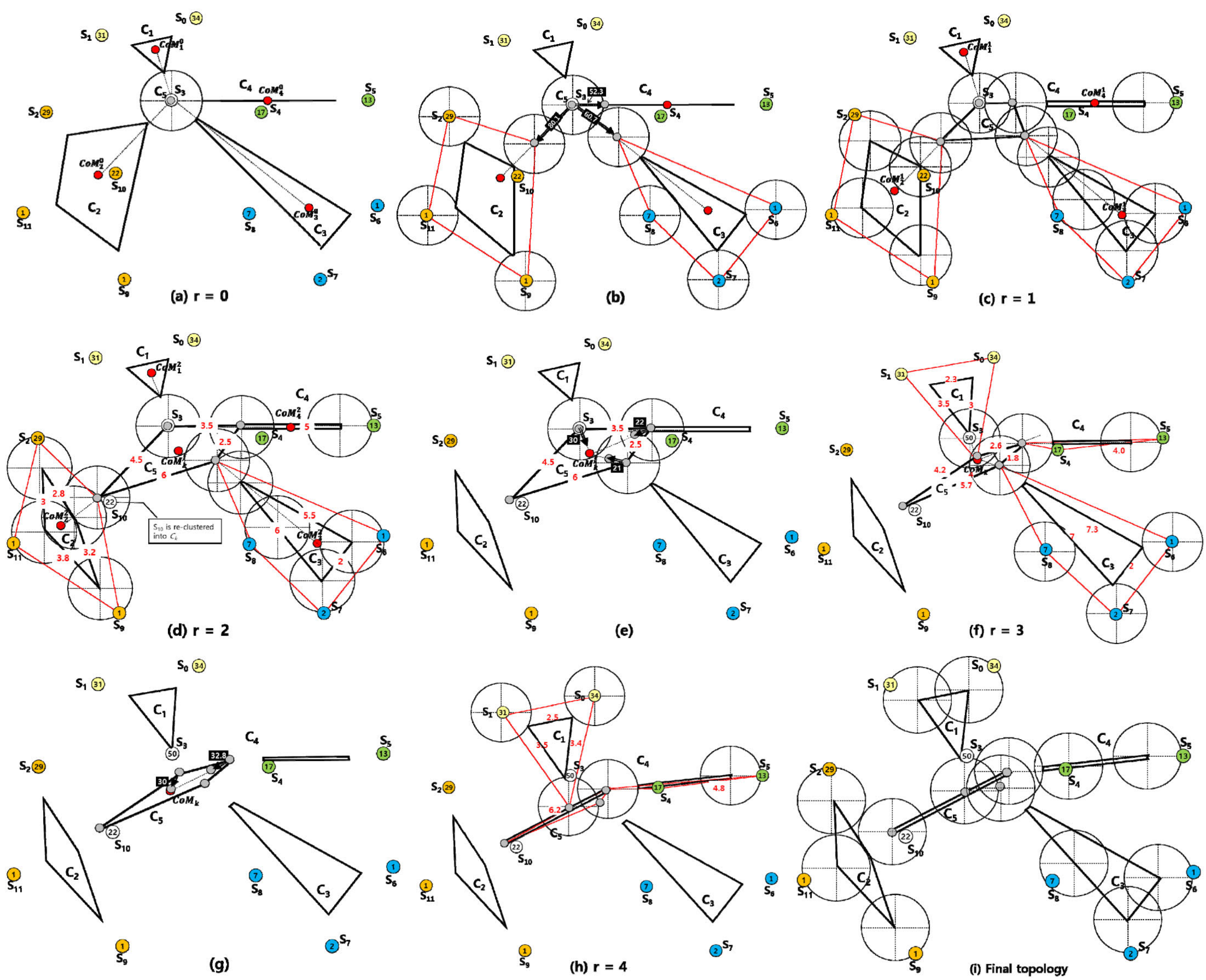

FIGURE 4. The 2nd phase of LOAF is dedicated to balance energy among MDCs by adjusting the rendezvous points P_i, $i < k$. Balancing is performed in an iterative manner during which LOAF expands C_k by moving P_i outwards towards C_i if $E_i > E_{AVG}$ in order to reduce more-than-energy consuming clusters C_i's,(a)-(e) and then P_l's of C_l's whose $E_l < E_{AVG}$ are moved inwards towards C_k, (f)-(h). (i) shows the result of the two steps.

Lemma 1: In case that N_{seg} is more than two, the energy-based center of mass among segments during the 1st phase of LOAF, denoted as eG, always lies inside the convex hull of all segments.

Proof: The computed energy-based center of mass among segments, denoted as *eG* can be one of three cases [40], [41].

(1) If there are only two segments, i.e., $N_{seg} = 2$, then *eG* lies on the line connecting the two segments.
(2) If $N_{seg} \geq 3$ and the segments form a convex polygon as illustrated by the example in Figure 2, then *eG* lies inside the polygon.
(3) If $N_{seg} \geq 4$ and the segments form a concave polygon as an example seen in Figure 5, then *eG* may lie outside the polygon depending on the inter-segment data volume as seen in Figure 5-(a). However, the *eG* lies inside the convex hull which includes all segments, represented as a solid polygon in Figure 5-(b). In addition, Figure 5-(c) shows the resulting topology formed by LOAF.

Therefore, the computed *eG* in the 1st phase of LOAF always lies inside the convex hull of the segments.

Lemma 2: The energy consumption of M_k serving for the central cluster formed in the 1st phase of LOAF is larger than the average energy of all MDCs, i.e., $E_k^o > E_{AVG}^0$.

Proof: Via the 1st phase of LOAF, the initial inter-cluster topology is formed in a star shape, where a central cluster C_k is formed by placing M_k as a stationary relay at the computed *eG*, around which $(k-1)$ non-central clusters are formed. Segments residing within R from M_k are made part of C_k and the rest of segments are grouped into C_i, $i < k$. Therefore, the energy consumed by M_k, E_k^o is completely determined by $E_C(M_i)$, which is primarily affected by the sum of all inter-segment data volume, i.e., $E_k^o \approx \sum_{\forall i,j,i \neq j} Data(S_i, S_j)$. In addition, the average energy consumption of all MDCs,

Algorithm 2 Pseudo Code of the 2nd Phase of LOAF

// Forming a central-cluster while balancing $E(M_i)$, $\forall i$
for $\forall i$ {// based on the initial inter-cluster topology formed during the 1st phase
  $C_i^0 \leftarrow C_i$, in the 1st phase;
  $TR_i^0 \leftarrow$ Tour path of M_i in C_i computed by [24], [35], [12];
  $P_i^0 \leftarrow$ Rendezvous point where M_i meets M_k;
  $E_i^0 \leftarrow$ Energy consumed by M_i in C_i;
  $CoM_i^0 \leftarrow$ Core of mass of a polygon formed by TR_i^0;
} **end for**
$E_{AVG}^0 \leftarrow Average(E_i^0, \forall i)$; // $E_k^o < E_{AVG}^0$
$SD^0 \leftarrow$ Standard Deviation of $E_i^f \forall i$; $rr \leftarrow 0$;
do {
  $r = 1$; //1st *step during which* P_m'*s move outward*
  do{
    $E_{least}^{r-1} = E_i^{r-1}$;
    for $E_i^{r-1} \geq E_{AVG}^{r-1}, \forall i$ {
      $L_i^r =$ Line from P_i^{r-1} towards CoM_i^{r-1};
      P_i^r moves $\left(\varepsilon \times \frac{E_i^{r-1}}{E_{least}^{r-1}}\right)$ meters along L_i^r;
  }**end for**
  Update TR_k^r based on P_i^r;
  if $\exists S_x \in C_i^r$ in a polygon formed by TR_k^r **then** {
    $C_k^r \cup = \{S_x\}$; $C_i^r - = \{S_x\}$;
  } **end if**
  Compute TR_i^r and CoM_i^r, $i < k$ and E_i^r, $\forall i$;
  $E_{AVG}^r = Average(E_i^r, \forall i)$; $r+ = 1$;
  } **while** ($E_k^r < E_{AVG}^r$)
  $f = r = r - 1$; // 2nd *step during which* P_l'*s move inward*
  do {
    $CoM_k^r = CoM_k^f$;
    for each C_i^r whose $E_l^r < E_{AVG}^r$ {
      $L_i^r =$ Line from P_i^r towards CoM_k^r;
      P_i^r moves $\left(\varepsilon \times \frac{E_{least}^{r-1}}{E_i^{r-1}}\right)$ meters along L_i^r;
      Update TR_i^r, E_i^r, and CoM_i^r;
    }**end for**
    $r+ = 1$;
  } **while** ($E_k^r > E_{AVG}^r$)
  $rr+ = 1$; $SD^{rr} \leftarrow$ Standard Deviation of $E_i^r \forall i$;
} **while** ($SD^{rr} < SD^{rr-1}$)
return $\{C_i^{r-1}, TR_i^{r-1} \forall i\}$

E_{AVG}^0 is mainly determined by a total data volume up/downloaded by each M_i i.e., $\sum_{\forall i}\sum_{S_j \in C_i} Data_{S_j}$ and a sum of $TR(M_i)$, $i < k$. Since each $TR(M_i)$ is bound to the tour length of each M_i in the area of $(\frac{L}{(k-1)})$, where L is the smallest polygon that includes all segments, $E_{AVG}^0 \approx \frac{1}{k}\left(\sum_{\forall i}\sum_{S_j \in C_i} Data_{S_j} + TR(\frac{L}{(k-1)})\right)$. Therefore, E_k^o cannot be less than E_{AVG}^0 because $\sum_{\forall i,j,i \neq j} Data\left(S_i, S_j\right) \approx \frac{\sum_{\forall i}\sum_{S_j \in C_i} Data_{S_j}}{k}$ and $TR\left(\frac{L}{(k-1)}\right) > 0$. □

Lemma 3: The energy consumption of M_i *serving a non-central cluster* $C_i, i < k$ *is minimized in* $O((N_{seg} - k) \cdot (N_{seg} - k)^4 log(N_{seg} - k))$.

Proof: The 1st phase of LOAF is devoted to forming $(k-1)$ non-central clusters, during which at most N_{seg} of clusters are grouped into $(k - 1)$ clusters via successive merging two clusters in subsequent rounds. The maximum number of non-central clusters corresponds to the first round ($r = 0$) and equals $(N_{cluster}^0 - k + 1)$, where $N_{cluster}^0$ $\left(\leq N_{seg}\right)$. The main time complexity of each round comes from finding a tour path of each cluster, i.e., $TR(M_i)$, which is computed using the heuristic whose execution time bounds to O($n^4 logn$) [24], where n is the number of segments in a cluster in our case, i.e., 1, 2, 3, ..., $(N_{cluster}^0 - k + 1)$ each round. Thus the 1st phase's time complexity equals $\sum_{n=1}^{(N_{cluster}^0 - k+1)} n^4 logn$ which is O($(N_{seg} - k) \cdot (N_{seg} - k)^4 log(N_{seg} - k)$). □

Theorem 1: LOAF guarantees the convergence to grouping N_{seg} *segments into k-clusters in a star topology.*

Proof: As proven in Lemma 3, LOAF groups N_{seg} segments into k-clusters in its first phase. Thus, for the proof of the convergence of N_{seg} segments into k-clusters, it is sufficient to show that (i) a segment S_i which is not visited by the updated tour of M_i is necessarily visited by M_k during the first step of the 2nd phase, and (ii) vice versa, for S_i that is not visited by TR_k^r ought to be covered by TR_i^r, $\exists i < k$ during the second step.

For proving (i), it is required to show that a candidate S_i for re-clustering is always placed between a convex hull CH_r of $\{S_j \forall j \in C_i, P_i^r\}$ and TR_i^r which is R away from CH_r and will be visited by M_i or M_k in the next round $(r+1)$. It is proven by contradiction. If S_i resides outside CH_r or inside R away from TR_i^r, it self-proves the found CH_r or TR_i^r is incorrect due to the natural features of a convex hull and the way of computing a tour path described in [24] respectively. Therefore, S_i that will be re-grouped into C_k is undoubtedly inside CH_r and in a range R from TR_i^r. In addition, in the cases in which S_i will not be reachable from TR_k^{r+1} and thus not be re-clustered into C_k as seen in Figure 6-(c) and (d), the S_i will be necessarily visited by the updated tour path TR_i^{r+1} as Figure 6-(a) and (b) show. The same proof applies to (ii) and thus k-clustering of the segments formed in a star topology is guaranteed by LOAF. □

Theorem 2: The time complexity of LOAF is proportional to the computation time for TR_i^r *which is determined by a gap between* N_{seg} *and the number of the available MDCs (k), as* $\Omega((N_{seg} + N_r k) \cdot (N_{seg} - k)^4 \cdot log(N_{seg} - k))$*, where* N_r *is the maximum iteration of the 2nd phase and bounds to* $O(\frac{Dist(eG,L)}{\varepsilon})$*, k is the number of available MDCs and* N_{seg} *is the number of segments in the area L.*

Proof: As proven in lemma 3 the 1st phase of LOAF is bound to O($(N_{seg} - k) \cdot (N_{seg} - k)^4 log(N_{seg} - k)$). In addition, the 2nd phase of LOAF is iterative in two steps. The number

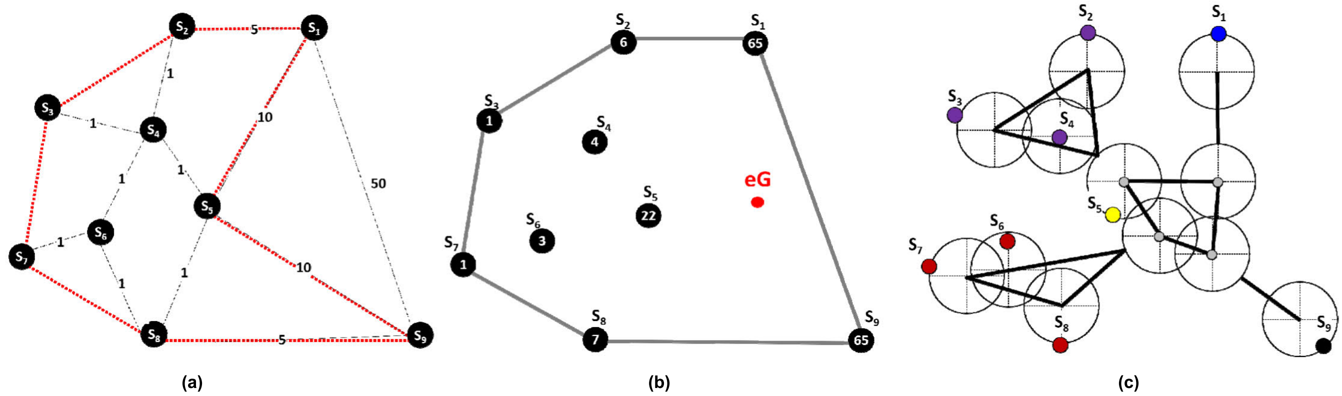

FIGURE 5. (a) An example setup of 9 segments that form a concave polygon. The numbers on each line represent data volume exchanged between the segments at the two ends in megabits; (b) Showing an energy-based center of mass, denoted as *eG* that lies outside the concave polygon due to the heavy data communication load between S_1 and S_9. However, *eG* lies inside the convex hull of segments represented by a solid line; (c) Depicting the formed data collection topology of five clusters, i.e., $C_1 = \{S_1\}$, $C_2 = \{S_2, S_3, S_4\}$, $C_3 = \{S_6, S_7, S_8\}$, $C_4 = \{S_9\}$, $C_5 = \{S_5\}$ and the MDCs' routes, assuming an area of interest = 270m×360m, R = 30m, and MDC count of five.

of rounds N_r for balancing energy in the steps is bound to $O(\frac{Dist(eCoM,L)}{\varepsilon})$, where L is the smallest polygon that includes all segments, and $Dist(eG, L)$ represents a distance from eG to the furthest point on L in meters. In addition, the execution time of each round in both steps is determined by computing $TR(M_i)$ for at most $(k-1)$ clusters and the average number of elements in clusters tentatively formed during the 2nd phase is $\frac{N_{seg}}{k}$. In addition, the two steps are iterative while the standard deviation of E_i, $\forall i$ has been improved. Therefore, the 2nd phase of LOAF requires the execution time complexity which is bounded by $\Omega(N_r k \cdot (\frac{N_{seg}}{k})^4 \cdot log(\frac{N_{seg}}{k}))$. In conclusion, the time complexity of LOAF for grouping N_{seg} segments into k-clusters in an energy-balanced manner equals $\{(N_{seg}-k) \cdot (N_{seg}-k)^4 log(N_{seg}-k) + N_r k \cdot (\frac{N_{seg}}{k})^4 \cdot log(\frac{N_{seg}}{k})\}$ which bounds to $\Omega((N_{seg}+N_r k) \cdot (N_{seg}-k)^4 \cdot log(N_{seg}-k)$, where $N_r = O(\frac{Dist(eCoM,L)}{\varepsilon})$. □

V. PERFORMANCE EVALUATION

In this section, the effectiveness of LOAF is validated through simulation. The simulation experiments study the performance with respect to the number of RNs populated by LOAF in comparison with the best known and recently published algorithms. The quality of the resulting topology of LOAF is also discussed.

A. SIMULATION ENVIRONMENT AND PERFORMANCE METRICS

We have implemented a simulation environment in C. The environment articulates the effect of damage on a single WSN that originally covers an area or multiple autonomous WSNs. Basically, varying numbers of segments are randomly located in an area of interest 1200m×1200m such that the segments are evenly distributed in the quadrants. The following parameters are used to vary the network characteristics:

- Communication range of relays (R): In general, R has the influence on forming clusters which include a set of segments visited by an MDC. The length of tour path is also affected by R, specifically, MDCs need to travel a shorter path with a larger 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. In addition, involving more segments

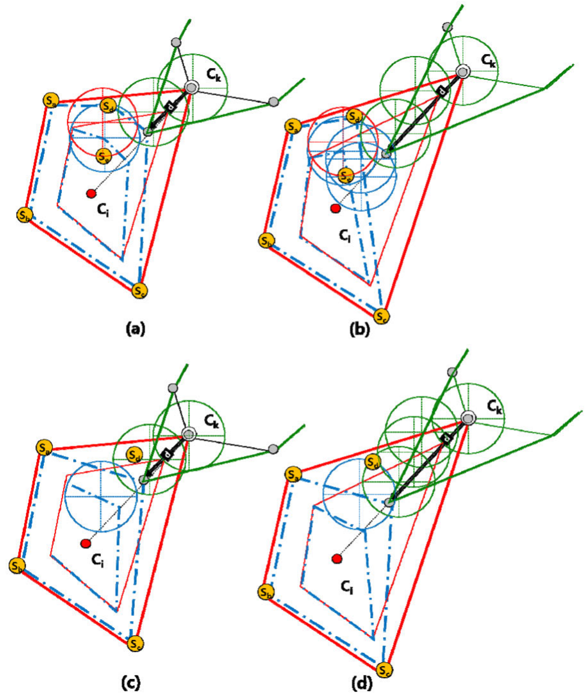

FIGURE 6. During the first step of the 2nd phase, S_d covered by M_i in round r is necessarily visited by M_i (a,b) or M_k (c,d) in the next round $(r+1)$. The inner solid line and inner dotted line represent TR_i^r and TR_i^{r+1} respectively. In (a) and (b), S_d remains in C_i since it is R away from TR_k^{r+1} and the convex hull used in round r is modified for computing TR_i^{r+1} considering S_d since S_d becomes a border segment in C_i^{r+1}.

may increase the traffic volume where more data is exchanged among segments and thus inter-MDC energy balancing becomes more complicated.

- 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 [7] 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 the value of N_{MDC} gets reduced, MDCs may consume more energy for visiting more segments.
- Average Inter-Segment Data Volume ($ISDV$): It represents the data traffic requirements between each pair of segments. The value of $ISDV$ would have influence on energy consumption of MDCs for wireless communication and the inter-MDC energy balancing.

The performance of LOAF is assessed using the following four metrics:

- Maximum inter-segment communication delay (D_{max}): Obviously, LOAF strives to minimize the maximum data delivery latency between segments. D_{max} is mainly determined by rendezvous time along the path between two furthest apart segments, which is affected by a count of clusters and inter-segment data traffic volume.
- Energy balance among MDCs: This is also a main objective of LOAF. Since LOAF tries to balance energy considering motion and communication both, energy balance among MDCs is computed as a standard deviation of total energy consumption of each MDC for motion and communication, i.e., it equals $\sqrt{\frac{\sum_{i=1}^{N_{MDC}} |E(M_i)-E(\bar{M})|^2}{N_{MDC}}}$, where $E(\bar{M}) = \frac{\sum_{i=1}^{N_{MDC}} E(M_i)}{N_{MDC}}$ and $E\left(M_i\right) = E_M\left(M_i\right) + E_C\left(M_i\right)$.
- Average energy consumption of all MDCs (AEC): It measures the average energy consumed by all MDCs before the first MDC dies. It will represent the absolute amount of energy consumption of MDCs for intermittently connecting segments.
- Network lifetime (NL): This indicates how long the formed intermittent topology stays fully operational and is measured as the duration until the first MDC fully depletes its on-board energy supply. In other words, NL represents the length of time during which every MDC operates and serves their clusters. This metric indicates the maximum energy consumed by an MDC. We measure NL as $\frac{(initE-AEC)}{BPC}$, where $initE$ is the initial energy of an MDC, BPC is battery power consumption of an MDC per second.
- Buffer space required at gateway segments: This metric is relevant to only the case in which some MDCs share a segment in their tours and uploads/downloads inter-cluster data from/at the buffer of the common segment. The maximum buffer space required within a certain time window is measured.

B. BASELINE APPROACHES

The performance of LOAF is compared to four competing approaches. The first two solutions focus on reducing inter-segment communication delay. One of them, namely ToCS [32], forms clusters of segments in a star topology like LOAF and balance tours of MDCs. The second is FOCUS [31], which opts to reduce the tour lengths of MDCs and adjusts their speed to minimize the travel distance and time required for transferring inter-clustering data between MDCs. Meanwhile, the focus of the third baseline approach, namely, MINDS [26], is more on reducing and balancing the MDC tours than reducing data delivery delay between segments. Like LOAF, the fourth approach is LEEF [27], opts to achieve MDCs' energy balance and optimize inter-segment delivery latency. These baseline approaches address the same problem tackled by LOAF using different solution strategies, as summarized below.

- ToCS forms clusters around the center of the area G and consists of two phases [32]. In the first phase, each cluster segment initially becomes a cluster C_i, and then combines two C_i's whose merging cost based on G is the least. The merging is repeated until $(k-1)$ clusters are formed. 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 that is on the convex hull of segments of C_i. Then each tour path is determined as done in IDM-kMDC [24] and the average tour length TL_{avg} is found. ToCS then adjusts the size of C_k towards C_i in order to balance the MDC tour lengths in the 2nd phase. During the 2nd phase, 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 $C_i \forall i$ and the 2nd phase terminates when $TL_{avg} \approx TL_i, \forall i$. In ToCS the waiting time at the rendezvous point contributes to the inter-segment communication delay due to the need for MDC synchronization where data uploading/downloading is performed.
- FOCUS operates in three phases [31]. In the 1st phase, segments are grouped into k disjoint clusters based on proximity. Each cluster is to be served by an MDC. In order to minimize the tour path of an MDC, clustering is initiated by selecting k farthest away segments from the centroid and then the remaining segments are joined in a greedy way. 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 $\overrightarrow{C_a, C_b}$ between every pair of clusters. The weight of $\overrightarrow{C_a, C_b}$ is equal to the increase in the tour length of C_a when C_a is extended towards C_b by including a segment from C_b. During the 2nd phase FOCUS opts to overlap the k clusters in order to minimize tour lengths of MDCs for the inter-cluster communication; it does so by selecting an intersection segment (IS) between

two adjacent clusters C_a and C_b. The pair of C_a and C_b 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 MDCs for reducing the time required at *IS* for the 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 the segments along the *mst* edges. Otherwise, MINDS opts to split the longest tour L into two groups until the number of groups equals k in the 2nd phase. Spitting is done by finding a center segment V_c of L from which the path on *mst* to the furthest segment is shortest. Then, the segments in L are divided into two groups g_a and g_b and V_c is designated as a rendezvous point between MDC_a and MDC_b. Forming g_a and g_b is based 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_a 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_b in the same way. If there are still ungrouped segments, each of them belongs to the closest group between g_a and g_b. In the 3rd phase, the tour path of MDCs is computed as done in [24].
- LEEF groups the segments into a set of k clusters in a star topology where a *hub-cluster* C_k facilitates data relaying between clusters. In the first phase, LEEF models an area of interest as a grid of equal-size square-shaped cells, based on which a set CS_T of the fewest cells that cover all segments is identified by evaluating each cell's *reachability* to segments, and *proximity* to the center G of the area. Then 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-step process: *greedy-expansion* and *optimization*. During *greedy-expansion*, 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 the 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 by adding a segment. In the *optimization* step, balancing inter-cluster energy is performed by adjusting cluster membership to enable C_{least} to grow.

Overall, the four baseline approaches all try to minimize tour lengths of MDCs. Like LOAF, LEEF and ToCS form a star inter-cluster topology while FOCUS and MINDS establish a cluster-based *mst*. Also, in FOCUS, MINDS, and LEEF, the MDCs rendezvous via a common segment for transferring inter-cluster traffic while in ToCS MDCs meet at a point where no segment is located. In order to minimize the waiting time for rendezvous ToCS strives to even MDC tours. However, ToCS does not schedule MDCs' rendezvous time. FOCUS also strives to reduce buffering space and time consumed at the common segment by adjusting the motion speed of MDCs. However, ToCS, FOCUS, and MINDS do not consider energy issues like LEEF and LOAF.

C. SIMULATION RESULTS

We have simulated multiple configurations, each has different combinations of values R, N_{seg}, N_{MDC}, and $ISDV_a$. The value of R is varied from 50 to 250 with increment of 25 and N_{seg} takes the values between 21 and 39 with increment of 3. In addition, the data volume between segments is randomly picked using a Gaussian distribution with mean, that takes values from {1, 2, 4, 8, 16, 32, and 64} with a standard deviation of 3. The other parameters for simulation setup 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., iRobot PackBot 510 [36] whose maximum speed is 9.6 km/h, i.e., about 2.59 meter/s. 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.

1) MAXIMUM INTER-SEGMENT DATA DELIVERY LATENCY (D_{max})

Figure 7 shows that LOAF experiences less worst-case inter-segment data transfer latency than all baseline approaches in terms of R, N_{seg}, N_{MDC} and $ISDV_a$. D_{max} is mainly affected by rendezvous time along the path between the furthest apart segments; Therefore, D_{max} is affected by a count of clusters in the path and inter-segment data volume and not much dependent on R. However, as seen in Figure 7-(a), MINDS, FOCUS, LEEF and LOAF show less D_{max} with larger R. This is because R has influence on the inter-cluster path length and traffic volume in their algorithms. In other words, in MINDS and FOCUS the inter-cluster traffic is delivered via a segment overlapped by multiple clusters and larger R makes a smaller number of cluster hops between two furthest apart segments.

In addition, both LEEF and LOAF ensure that there are at most two hops between any pair of segments and unlike ToCS which also forms a star topology, they consider energy affected by R during clustering. Nonetheless, LOAF yields the least D_{max} among all approaches, which is attributed to the following four design features: (i) forming a star-shaped inter-cluster topology where the number of clusters between every pair of segments becomes at most three (like ToCS and LEEF, and unlike FOCUS and MINDS); (ii) MDCs in LOAF 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 (like LEEF, unlike ToCS); (iii) LOAF tries to keep a pair of segments which exchange large volumes of data in the same cluster by considering the inter-segment data exchange overhead during cluster formation. This reduces the inter-cluster traffic volume.

TABLE 1. Simulation setup.

Parameter	Description	Value (unit)
Area	Area of interest	1200m × 1200m
R	Radio range of an MDC	[50, 250] (meters) with an increment of 25
N_{seg}	Number of segments	[21, 39] with an increment of 3
N_{MDC}	Number of available MDCs	$[N_{RN}-1, \ldots, y]$ with a decrement of 2, N_{RN} is computed using [7] and, $y = \max([N_{RN}-11], 2)$
$ISDV$	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)
$Speed_{MD}$	Speed of an MDC	2.59 (meter/sec)
BW	Wireless bandwidth of an MDC	1Mbps
$initE$	Initial value of Energy	54400 Joule
BPC	Battery power consumption of an MDC	0.5 Joule/second [37]

While the latter feature is also supported by LEEF, LOAF distinguishes itself by forming a more efficient inter-segment topology, particularly in designating the central cluster. LOAF first locates the central cluster at the energy-balanced center of mass among all segments in terms of communication and motion; then non-central clusters are formed considering segments' individual position. Meanwhile, LEEF identifies the central cluster after forming the non-central ones. Accordingly, the inter-cluster topology formed by LOAF better factors in inter-segment data volume and proximity and thus yields less inter-segment data exchange latency. It is worth noting that ToCS which also forms a star inter-cluster topology has the largest D_{max} with all values of R. This is because ToCS does not reduce rendezvous time by careful scheduling of the MDC tours.

Moreover, as seen in Figure 7-(b) D_{max} grows for all approaches when the number of segments, N_{seg} increases. Such a growth is intuitive since adding more segments not only requires longer trips of MDCs but also involves heavier inter-segment data traffic, and subsequently more stops and longer loading time. Nonetheless, LOAF and LEEF yield lower latency than all baselines regardless of N_{seg}. This is because they strive to include a pair of segments which exchange data in the same cluster, and hence reduces inter-cluster trip for data transfer and diminishes inter-segment data delivery latency. In addition, LOAF outperforms LEEF, which is again attributed to how LOAF places the central cluster that plays the role of a hub for the inter-segment communication. It is worth noting that the performance of ToCS is worse than FOCUS and MIND due to the implicit waiting time for MDC rendezvous to support data up/downloading.

In addition, LOAF yields less D_{max} than all baselines for various numbers of MDCs as seen in Figure 7-(c). Such performance is due to the advantage of forming a star inter-cluster topology, in which data delivery between any pair of two segments requires at most two MDC rendezvous. Although both LEEF and ToCS also form a star topology, ToCS involves waiting time during MDC rendezvous and LEEF pursues inferior clustering procedure as mentioned earlier. It is worth to note that ToCS and LOAF experience little changes in D_{max} when varying N_{MDC}, while D_{max} slightly decreases as N_{MDC} grows. This is because ToCS, LOAF and LEEF form a star topology where D_{max} is mainly affected by a tour length in a central cluster. Unlike ToCS and LOAF which strive to equalize the tour length for all MDCs, LEEF initially forms a central cluster by placing an MDC at a geographical center of an area and $(N_{MDC} - 1)$ of virtual non-central clusters which are merged. Thus, a larger value of N_{MDC} tends to shorten a tour length in a central cluster of LEEF. Meanwhile, increasing N_{MDC} raises D_{max} in MINDS and FOCUS since they form a *mst*-based topology of clusters where the more MDCs are involved, the higher inter-cluster communication traffic along the path between two the furthest apart segments becomes and consequently D_{max} increases.

As seen in Figure 7-(d), growing the inter-segment data volume boosts D_{max}. This is fairly anticipated because increasing the data exchange between segments implies that the MDCs take more time in data up/downloading. However, it is worth noting that D_{max} for LEEF and LOAF is much less than its value in ToCS, FOCUS and MINDS. For instance, with *ISDV* of 64Mb LEEF and LOAF yield 9% of D_{max} of ToCS and 12% of D_{max} of FOCUS. In addition, as *ISDV* grows, D_{max} increases at a much lower rate in LEEF and LOAF than the other three approaches. These results are due to the fact that LEEF and LOAF factor data volume between segments in determining the cluster membership. Meanwhile, ToCS, FOCUS and MINDS do not consider data traffic volume between segments during clustering.

2) ENERGY BALANCE AMONG MDCs

In Figure 8, the effectiveness of LOAF in terms of energy balance among MDCs is validated. LOAF yields better performance than all baselines regardless of R, N_{seg}, N_{MDC} and *ISDV* since it: (i) considers energy consumed by MDCs in up/downloading inter-segment data traffic, and in touring between clusters, and (ii) strives to equalize the

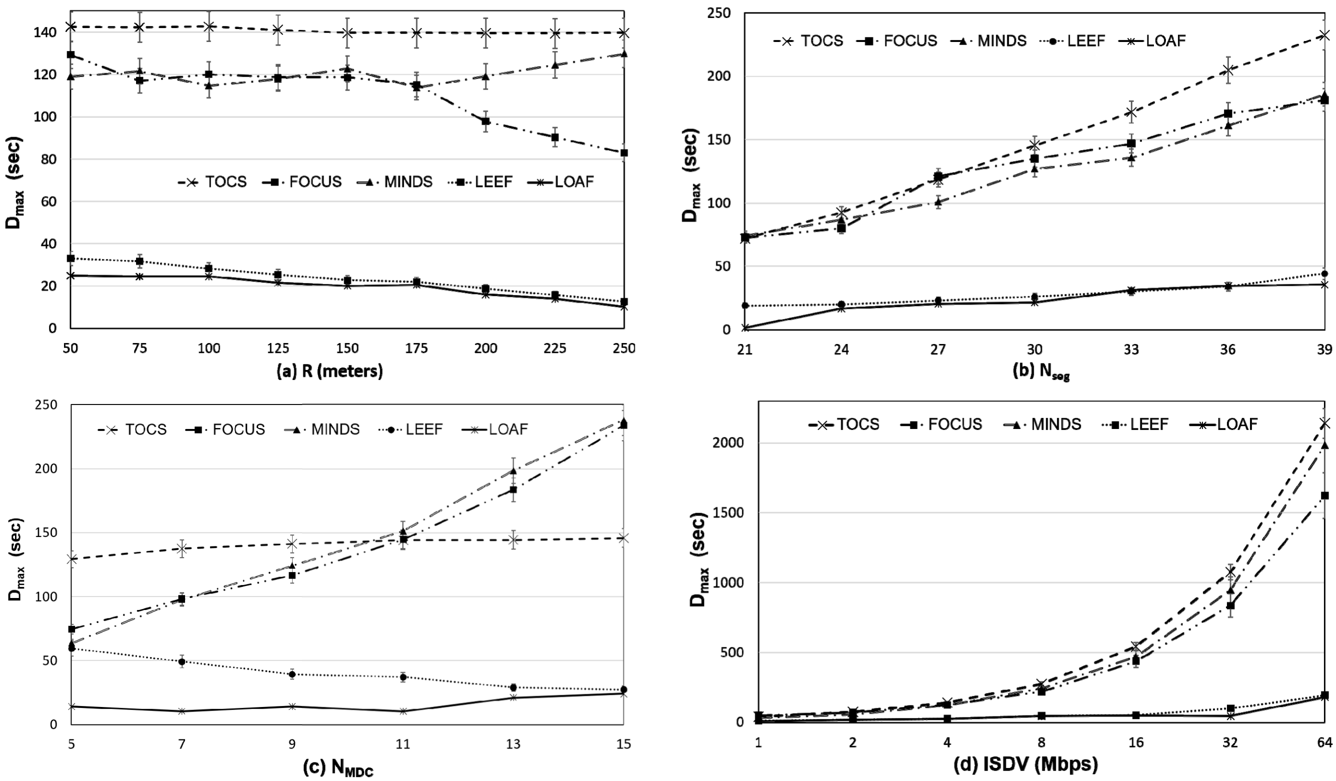

FIGURE 7. The performance comparison of LOAF to LEEF, ToCS, FOCUS, and MINDS in terms of the inter-segment data delivery latency (D_{max}). (a), (b), (c) and (d) show the results with respective to various R, N_{seg}, N_{MDC}, and $ISDV$ respectively with a fixed value of $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, and $ISDV = 4$.

energy overhead while forming the inter-cluster topology. Moreover, LOAF's optimized design enables it to outperform LEEF which also considers energy balancing during cluster formation. In other words, unlike LEEF which locates the central cluster geographically at the center of the deployment area, LOAF forms the central cluster to reduce the energy overhead. In addition, LOAF considers locations of individual segments and thus optimizes the formation of balanced clusters using more precise computation than LEEF.

As seen in Figure 8-(a), balancing the energy overhead among MDCs has nothing to do with R for all approaches. Meanwhile, all approaches show more balanced clustering topology with smaller N_{seg} as shown in Figure 8-(b). This phenomenon is because an MDC consumes its power for up/downloading data traffic volume exchanged between segments as well as touring the segments. Thus, equalizing energy consumption among MDCs gets highly complicated when more segments are involved and subsequently the combinations of inter-segment data volume vary. This is especially evident in FOCUS, MINDS and ToCS, all of which do not consider energy used for the inter-segment data exchanges during clustering and thus the inter-MDC energy balancing rapidly drops with large N_{seg}. Meanwhile, LOAF outperforms all baselines for various values of N_{seg} and its superiority scales with N_{seg}.

Moreover, Figure 8-(c) demonstrates that the star inter-cluster topology formed by LOAF, LEEF and ToCS has more positive impact on load balancing as N_{MDC} grows than that of the *mst*-based topology formed by FOCUS and MINDS. Formation of clusters around a central cluster provides higher flexibility in adjusting proximity and traffic volume between clusters than a *mst*-based topology; such increased flexibility becomes even more influential as N_{MDC} increases. Finally, Figure 8-(d) demonstrates how LOAF efficiently copes with the increased inter-segment data volume and thus it outperforms all baselines in terms of energy consumption balance. This result comes from the fact that LOAF tries to keep a high volume of data exchanging segments in the same clusters and the positive impact of the design principle becomes clearer as the inter-segment data volume, $ISDV$ grows.

3) AVERAGE ENERGY CONSUMPTION OF ALL MDCs

Figure 9 reports the average energy consumed by all MDCs. LOAF outperforms all baselines for varying R, N_{seg}, N_{MDC} and $ISDV$. Figure 9-(a) shows that ToCS, LEEF and LOAF require less energy as R increases, unlike FOCUS and MINDS which pursue a *mst*-based inter-cluster topology. Enlarging the communication range of an MDC, R, helps to shorten the tour length in a star inter-cluster topology and in turn results in reducing the average energy consumption of MDCs. Meanwhile, the tour length is hardly

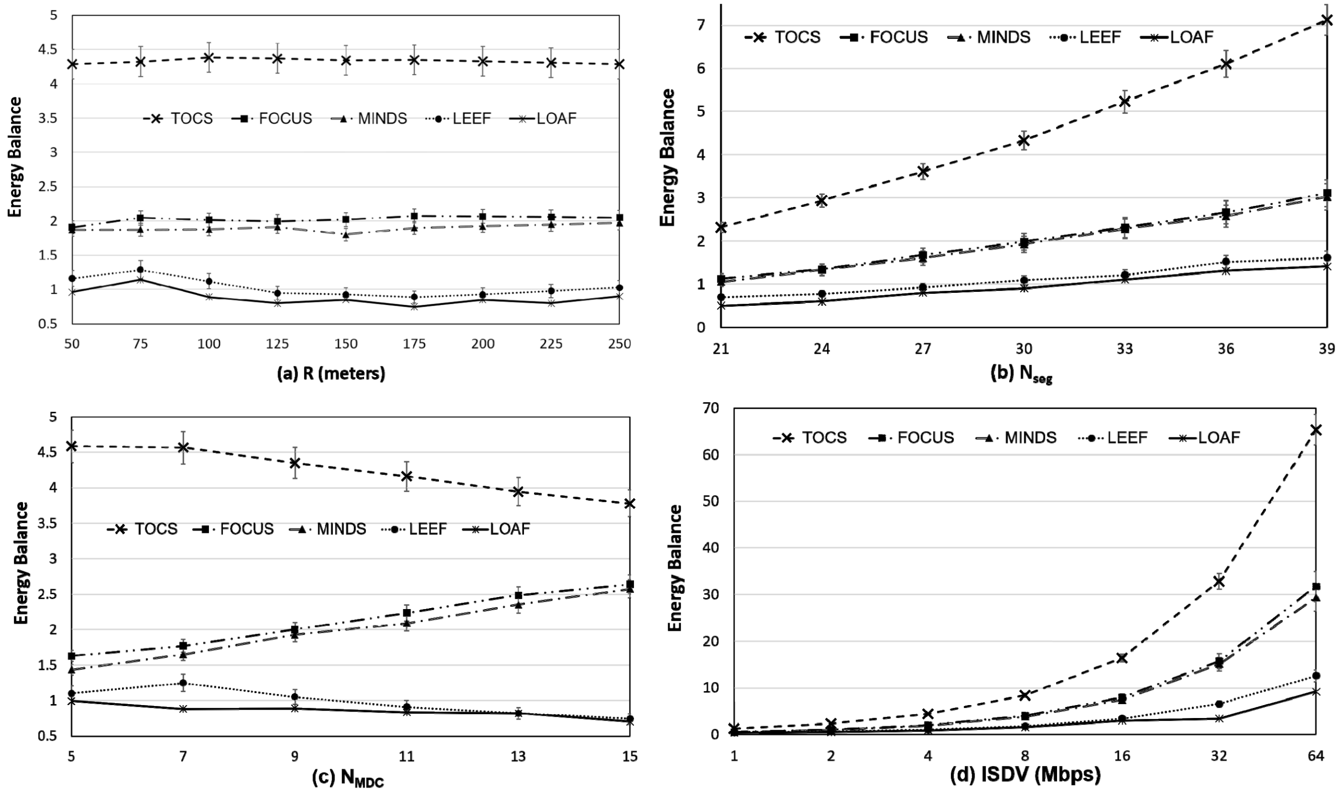

FIGURE 8. (a), (b), (c) and (d) show the performance comparison of LOAF to LEEF, ToCS, FOCUS, and MINDS in terms of energy balance with respect to various R, N_{seg}, N_{MDC}, and $ISDV$ respectively with a fixed value of $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, and $ISDV = 4$.

affected under a *mst*-based inter-cluster topology. Thus, Figures 9-(a) and 10-(a) evidently show that LOAF groups segments in an energy-efficient and balanced way. For instance, LOAF requires about 300% less energy and achieves approximately 480% more energy-balanced clustering in comparison to ToCS.

In addition, larger values of N_{seg} or $ISDV$ require more MDC energy for all approaches as seen in Figure 9-(b) and 9-(d), respectively. This is expected because serving more segments or delivering more inter-segment data traffic would enlarge a total tour length and/or data loading time for MDCs. Figure 9-(c) shows another advantage of a star topology where the energy overhead diminishes dramatically as N_{MDC} grows. We note that for small values of N_{MDC}, LOAF achieves outstanding reduction in the energy overhead as compared to ToCS and LEEF. Such superiority is attributed to the energy-centric design principle of LOAF which strives to place a central cluster at the energy balanced center of all segments and based on which the formation of non-central clusters is optimized.

4) NETWORK LIFETIME (NL)

Figure 10 demonstrates the network lifetime during which the formed topology by each algorithm normally operates. In other words, it shows how long the MDCs tour as planned after federation and thus every pair of inter-segment communication is maintained. We study the effect of R, N_{seg}, N_{MDC}, and $ISDV$ on NL, measured as $\frac{(initE - AEC)}{BPC}$, where $initE$ and BPC are set as 54400 Joule and 0.5 Joule/sec respectively. The results shown in Figure 10 are very much expected. Basically, LOAF yields the longest network lifetime in all cases and the results are consistent with Figure 9 where LOAF outperforms all baselines in terms of the average energy consumption.

5) BUFFER SPACE REQUIRED AT GATEWAY SEGMENTS

Figure 11 presents the maximum buffer requirement of rendezvous segments while delivering the inter-cluster data traffic. Therefore, this performance metric is applicable to only FOCUS, MINDS, LEEF and LOAF in which an MDC uploads/downloads the inter-cluster data from/at a buffer of a segment that is to be visited by another MDC serving a different cluster. As seen in Figure 11, LOAF requires the least buffer space regardless of R, N_{seg}, N_{MDC} and $ISDV$. This is because LOAF opts to not only reduce inter-cluster data traffic by keeping a pair of segments that exchange large data volume in the same cluster but also place a central cluster at the energy-balanced location considering the inter-segment traffic volume. Such design diminishes the buffer requirement of segments at rendezvous points.

Figure 11-(a) shows how the buffer space requirement is affected by the communication range of an MDC, i.e., R. For FOCUS, MINDS and LOAF little change can be observed because R does not have a direct influence on a count of

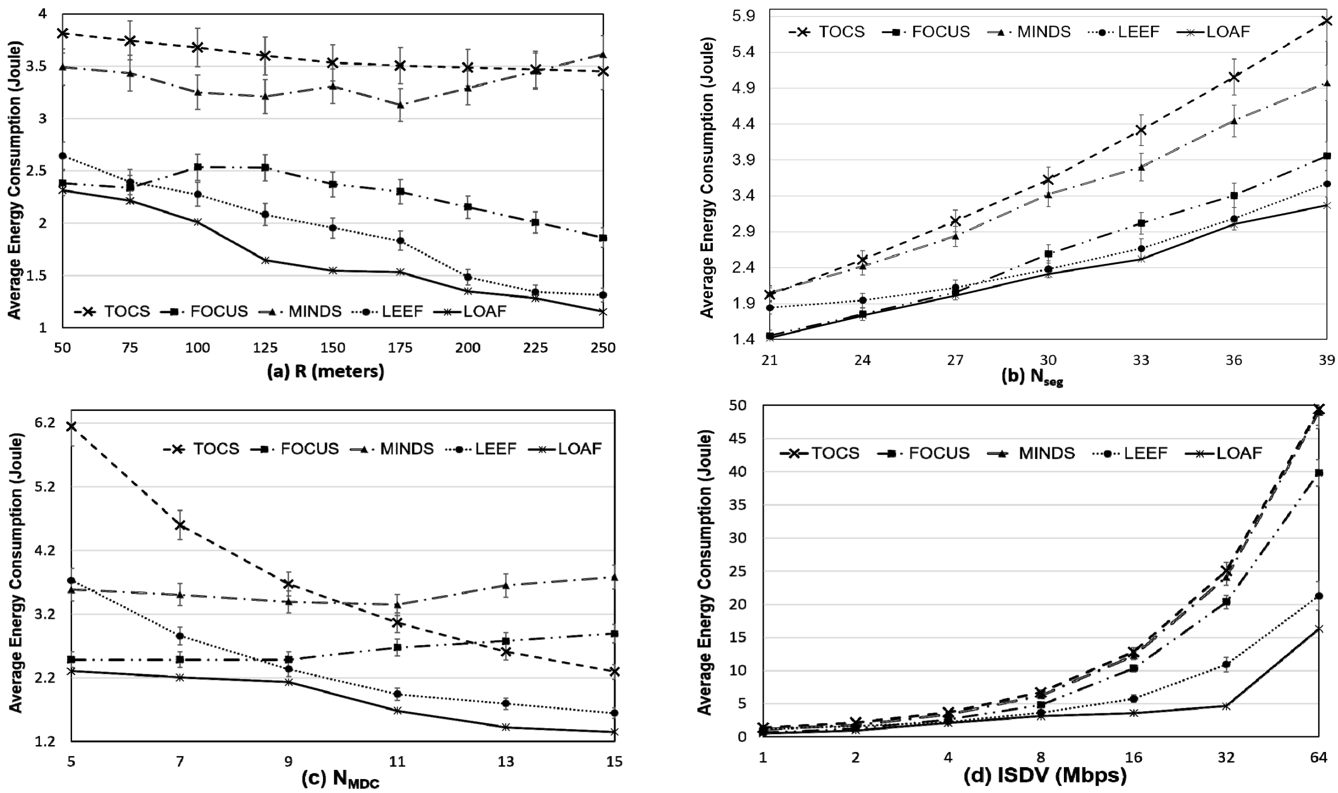

FIGURE 9. The performance comparison of LOAF in terms of the average energy consumption of MDCs with a fixed value of $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, and $ISDV = 4$. (a), (b), (c) and (d) show the results with respective to various R, N_{seg}, N_{MDC}, and $ISDV$.

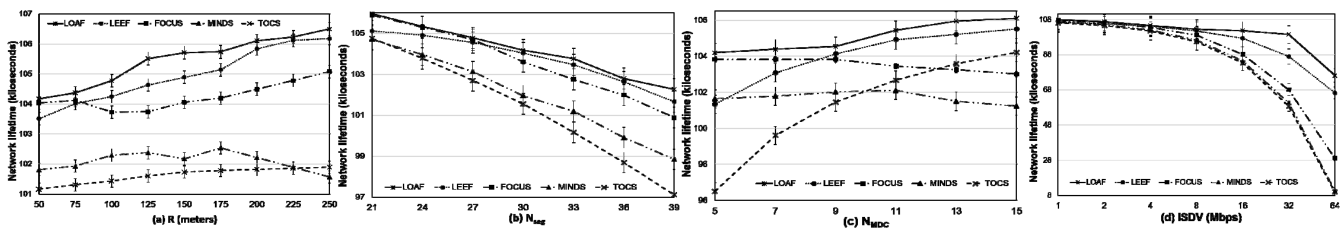

FIGURE 10. The performance comparison of LOAF with respect to network lifetime with a fixed value of $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, and $ISDV = 4$. (a), (b), (c) and (d) show the results with respective to various R, N_{seg}, N_{MDC}, and $ISDV$.

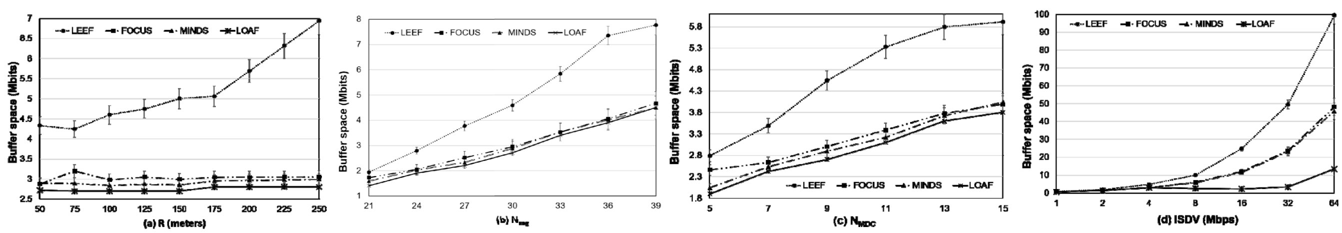

FIGURE 11. The performance comparison of LOAF in terms of the required buffer space for MDCs with a fixed value of $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, and $ISDV = 4$. (a), (b), (c) and (d) show the results with respective to various R, N_{seg}, N_{MDC}, and $ISDV$.

segments belonging to a cluster or the inter-cluster data traffic volume; particularly, the latter affects the maximum buffer space of segments at rendezvous points. In addition, LOAF yields better performance than FOCUS, MINDS and LEEF since it places a central cluster at a mass of inter-segment traffic demands and thus reduces burdens on the shared segment in the central cluster. It is worth noting that LEEF requires more buffers as R grows. This is because LEEF exploits a cell-based grid architecture where the size of a cell is determined by R and identifies the smallest set CS_T of the selected cells that covers all segments, based on which a star-shaped cluster topology is formed. Thus, as R gets larger one cell in CS_T

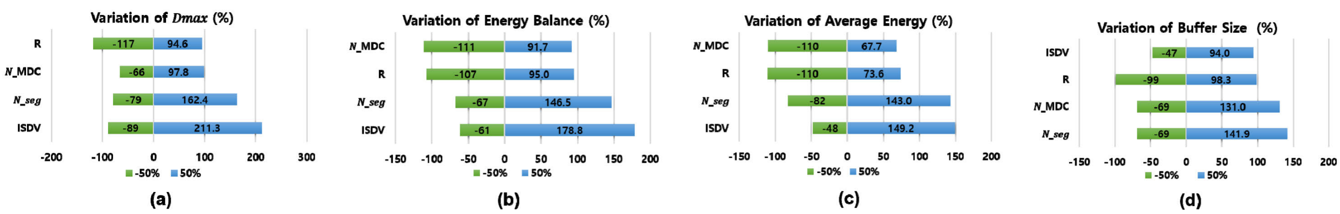

FIGURE 12. Comparison of the effect of various parameters R, N_{seg}, N_{MDC} and $ISDV$ on performance of LOAF in terms of (a) the inter-segment data delivery latency (D_{max}), (b) the energy balance among mobile data collators, (c) the average energy consumption and (d) buffer size. In each chart, the left and right bars represent the results of minus and plus 50% variations from a set of the base values for input parameters which is {R = 100, N_{seg} = 30, N_{MDC} = 9, $ISDV$ = 4}.

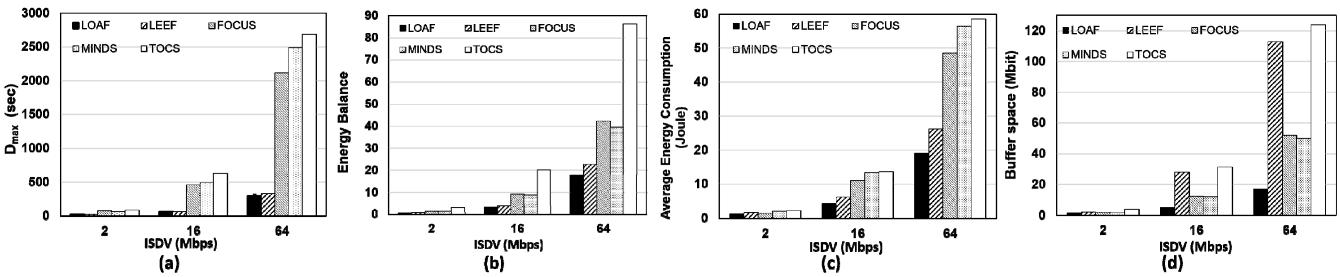

FIGURE 13. Comparison of the effect of change of the inter-segment data volume. ISDV is initially set at 8 Mbps and later changes to 2, 16, or 64 Mbps and the other input parameters are set as R = 100, N_{seg} = 30, N_{MDC} = 9. Each of (a), (b), (c), and (d) shows LOAF maintains a performance advantage in term of inter-segment data delivery latency, energy balance among MDCs, energy consumption of MDCs and buffer requirement for MDC respectively.

probably covers more segments which in turns requires more buffer space for storing the inter-cluster data if the cell is included in a central cluster.

Moreover, as much expected and seen in Figure 11-(b) and 11-(d), larger buffer space is required for all approaches as N_{seg} or $ISDV$ rises due to the increased inter-segment data volume. Figure 11-(c) captures the effect of the number of MDCs. Growing N_{MDC} implies more clusters, which in turn introduces more inter-cluster data traffic. Therefore, the shared segments need more buffer space as N_{MDC} increases.

6) COMPARING THE IMPACT OF VARIOUS PARAMETERS ON LOAF

Figure 12 captures the sensitivity of the performance to configuration parameters, R, N_{seg}, N_{MDC} and $ISDV$. In each chart, the left and right bars represent the results of minus and plus 50% variations from the base values for input parameters, i.e., $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$, $ISDV = 4$. As seen in Figure 12-(a), (b) and (c), an increment in the $ISDV$ value seems to influence LOAF more than that of R, N_{seg}, and N_{MDC} for D_{max}, energy balance between MDCs and the average energy consumption. This is because D_{max} is determined by rendezvous time between MDCs which is primarily determined by the inter-segment data volume. The energy consumption and balance are also more affected by the data communication load between segments than the travel path. Thus, a 50% increase of $ISDV$ from the baseline setting causes larger performance variations than other parameters. These results further confirm the importance of factoring in the data upload/offload overhead in the solutions, something that sets LOAF apart from competing approaches.

Meanwhile, Figure 12-(d) shows that N_{seg} and N_{MDC} have more effect on the maximum buffer size required at gateway segments than $ISDV$. This is because such a requirement is mostly determined by the formed topology, i.e., MDC travel paths to serve a group of segments, rather than the inter-segment data volume.

7) COMPARING THE EFFECT OF VARIATION IN THE DATA VOLUME

Figure 13 shows how LOAF and the baseline approaches respond to changes in the inter-segment data volume (ISDV) during operation. In the experiment, ISDV is initially set at 8 Mbps and later changes to 2, 16, or 64 Mbps. The other input parameters are fixed as $R = 100$, $N_{seg} = 30$, $N_{MDC} = 9$. With this setup, the initial energy-balanced cluster is formed in the first phase of LOAF considering 8 Mbps inter-segment data volume and maintained during the second phase where data volume between segments changes to 2, 16, or 64 Mbps. The same steps are applied to LEEF which also works based on clustering. As seen in Figure 13, LOAF maintains its superiority and outperforms LEEF as well as FOCUS, MINDS and ToCS even when the value of ISDV changes in terms of inter-segment data delivery latency, energy balance between MDCs and energy consumption of MDCs and required buffer space for MDCs.

VI. CONCLUSION

In this paper, we have studied the federation problem of disjoint WSN segments using a limited number (k) of mobile relays that provide intermittent inter-segment connectivity. We have proposed LOAF, a novel approach that groups the segments into k clusters and defines energy-efficient and also balanced tour paths for the mobile relays by factoring in both inter-segment data volume and proximity to the formed clusters. LOAF finally forms a star topology where a central cluster is located at the center of the area considering inter-segment data volumes as well as geographical distance between segments. The simulation results demonstrate that LOAF outperforms competing approaches in terms of the maximum inter-segment data delivery latency, energy balance among MDCs, average energy consumption of MDCs and network lifetime with a competitively small amount of buffer space requirement. The effectiveness of LOAF grows with larger MDC communication ranges, and with increased segment and MDC counts.

REFERENCES

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

[2] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Comput. Netw.*, vol. 58, pp. 254–283, Jan. 2014.

[3] M. Younis, S. Lee, I. F. Senturk, and K. Akkaya, "Topology management techniques for tolerating node failure," in *The Art of Wireless Sensor Networks*, H. Ammari, Ed. New York, NY, USA: Springer, Oct. 2013. [Online]. Available: http://www.springer.com/engineering/signals/book/978-3-642-40008-7

[4] G.-H. Lin and G. Xue, "Steiner tree problem with minimum number of Steiner points and bounded edge-length," *Inf. Process. Lett.*, vol. 69, no. 2, pp. 53–57, Jan. 1999.

[5] F. M. Al-Turjman, H. S. Hassanein, and M. Ibnkahla, "Optimized relay placement to federate wireless sensor networks in environmental applications," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, Istanbul, Turkey, Jul. 2011, pp. 2040–2045.

[6] S. Lee and M. Younis, "Recovery from multiple simultaneous failures in wireless sensor networks using minimum Steiner tree," *J. Parallel Distrib. Comput.*, vol. 70, no. 5, pp. 525–536, May 2010.

[7] F. Senel and M. Younis, "Relay node placement in structurally damaged wireless sensor networks via triangular Steiner tree approximation," *Comput. Commun.*, vol. 34, pp. 1932–1941, May 2011.

[8] S. Lee and M. Younis, "Optimized relay node placement for connecting disjoint wireless sensor networks," *Comput. Netw.*, vol. 56, no. 12, pp. 2788–2804, Aug. 2012.

[9] Z. Cheng, M. Perillo, and W. B. Heinzelman, "General network lifetime and cost models for evaluating sensor network deployment strategies," *IEEE Trans. Mobile Comput.*, vol. 7, no. 4, pp. 484–497, Apr. 2008.

[10] S. Misra, N. E. Majd, and H. Huang, "Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks," *IEEE Trans. Comput.*, vol. 63, no. 12, pp. 2933–2947, Dec. 2014.

[11] F. Senel, M. F. Younis, and K. Akkaya, "Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1835–1848, May 2011.

[12] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Netw.*, vol. 24, pp. 1–19, Jan. 2015.

[13] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm for k-connected relay node placement in wireless sensor networks," in *Proc. 2nd Int. Conf. Comput. Commun. Technol.* New Delhi, India: Springer, 2016, pp. 721–729.

[14] S. Lee, M. Younis, and M. Lee, "Optimized bi-connected federation of multiple sensor network segments," *Ad Hoc Netw.*, vol. 38, pp. 1–18, Mar. 2016.

[15] S. Misra, S. Dong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: Formulation and approximations," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 434–447, Apr. 2010.

[16] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Efficient approximations," in *Proc. 7th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Boston, MA, USA, Jun. 2010, pp. 1–9.

[17] S. Lee and M. Younis, "EQAR: Effective QoS-aware relay node placement algorithm for connecting disjoint wireless sensor subnetworks," *IEEE Trans. Comput.*, vol. 60, no. 12, pp. 1772–1787, Dec. 2011.

[18] A. Nigam and Y. K. Agarwal, "Optimal relay node placement in delay constrained wireless sensor network design," *Eur. J. Oper. Res.*, vol. 233, no. 1, pp. 220–233, Feb. 2014.

[19] H. M. Almasaeid and A. E. Kamal, "Data delivery in fragmented wireless sensor networks using mobile agents," in *Proc. 10th ACM Symp. Modeling, Anal., Simulation Wireless Mobile Syst. (MSWiM)*, 2007, p. 96.

[20] H. M. Almasaeid and A. E. Kamal, "Modeling mobility-assisted data collection in wireless sensor networks," in *Proc. IEEE GLOBECOM-IEEE Global Telecommun. Conf.*, Dec. 2008, pp. 1–5.

[21] K. Li and K. A. Hua, "Mobile data collection networks for wireless sensors," in *Proc. 5th Int. Conf. Multimedia Commun., Services Secur. (MCSS)*, Krakow, Poland, May 2012, pp. 200–211.

[22] W. Wang, V. Srinivasan, and K.-C. Chua, "Extending the lifetime of wireless sensor networks through mobile relays," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1108–1120, Oct. 2008.

[23] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1472–1483, May 2013.

[24] F. Senel and M. Younis, "Optimized interconnection of disjoint wireless sensor network segments using K mobile data collectors," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 492–496.

[25] J. L. V. M. Stanislaus and M. Younis, "Delay-conscious federation of multiple wireless sensor network segments using mobile relays," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Québec City, QC, Canada, Sep. 2012, pp. 1–5.

[26] Y. K. Joshi and M. Younis, "Mobility-based Internetworking of disjoint segments," in *Proc. 27th Biennial Symp. Commun. (QBSC)*, Kingston, ON, Canada, Jun. 2014, pp. 193–197.

[27] S. Lee, M. Younis, B. Anglin, and M. Lee, "LEEF: Latency and energy efficient federation of disjoint wireless sensor segments," *Ad Hoc Netw.*, vol. 71, pp. 88–103, Mar. 2018.

[28] A. Abbas and M. Younis, "Establishing connectivity among disjoint terminals using a mix of stationary and mobile relays," *Comput. Commun.*, vol. 36, no. 13, pp. 1411–1421, Jul. 2013.

[29] Y. K. Joshi and M. Younis, "Restoring connectivity in a resource constrained WSN," *J. Netw. Comput. Appl.*, vol. 66, pp. 151–165, May 2016.

[30] I. F. Senturk and K. Akkaya, "Mobile data collector assignment and scheduling for minimizing data delay in partitioned wireless sensor networks," in *Proc. Int. Conf. Ad Hoc Netw.*, Oct. 2013, pp. 15–31.

[31] B. Kalyanasundaram and M. Younis, "Using mobile data collectors to federate clusters of disjoint sensor network segments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 1496–1500.

[32] J. L. V. M. Stanislaus and M. Younis, "Mobile relays based federation of multiple wireless sensor network segments with reduced-latency," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 6407–6411.

[33] D. K. Goldenberg, J. Lin, and A. S. Morse, "Towards mobility as a network control primitive," in *Proc. 5th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2004, pp. 163–174.

[34] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, 1999.

[35] A. Kashyap, S. Khuller, and M. Shayman, "Relay placement for higher order connectivity in wireless sensor networks," in *Proc. IEEE 25TH IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[36] *Robotyka, Roboty, Mechatronika I Automatyka*. Accessed: Dec. 23, 2019. [Online]. Available: http://www.asimo.pl/materialy/download/iRobot_510_PackBot_for_HazMat_Technicians.pdf

[37] *Ah Rechargeable Lithium-Ion Battery*. Accessed: Dec. 23, 2019. [Online]. Available: https://www.bren-tronics.com/bt-70791ck.html

[38] B.-S. Kim, K.-I. Kim, B. Shah, F. Chow, and K. Kim, "Wireless sensor networks for big data systems," *Sensors*, vol. 19, no. 7, p. 1565, Apr. 2019.

[39] M. Kuorilehto, M. Hännikäinen, and T. D. Hämäläinen, "A survey of application distribution in wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 5, Dec. 2005, Art. no. 859712.

[40] W. Beiglböck, "The center-of-mass in Einsteins theory of gravitation," *Commun. Math. Phys.*, vol. 5, no. 2, pp. 106–130, Apr. 1967.

[41] L. Costa, O. Filipe, and J. Natário, "Center of mass, spin supplementary conditions, and the momentum of spinning particles," in *Equations of Motion in Relativistic Gravity*. Cham, Switzerland: Springer, 2015, pp. 215–258.

SOOKYOUNG LEE received the B.S. and M.S. degrees in computer science from Ewha Womans University, South Korea, in 1995 and 1997, respectively, and the Ph.D. degree in computer science from the University of Maryland, Baltimore County, USA, in 2010. She has been with LG Electronics Inc., the Electronics and Telecommunications Research Institute, the Korea Electronics Technology Institute, and Samsung Electronics Company Ltd., South 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. She is currently a Research Professor with the Department of Computer Science and Engineering, Ewha Womans University. Her research interests include 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 (Senior Member, IEEE) received the Ph.D. degree in computer science from the New Jersey Institute of Technology, USA. He is currently an Associate Professor with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County (UMBC). Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems Research and Development 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. He has published over 260 technical articles in refereed conferences and journals. He 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. His technical interests include network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication, and distributed real-time systems. He is also a Senior Member of the IEEE Communications Society.

MOHAMMED ALSOLAMI received the B.S. degree in education and computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2005, the M.A. degree in career and technical education from Western Michigan University, USA, in 2012, and the M.S. degree in computer science from Western Michigan University, USA, in 2013. He is currently pursuing the Ph.D. degree in computer science with the University of Maryland, Baltimore County, Baltimore, MD, USA. Since 2005, he joined the Royal Commission, Yanbu Industrial, Saudi Arabia, where he is also a Faculty Member and a Research Member. His awards and honors include the SACM Fellowship, Monshaat's Entrepreneurship, AI Hackathon Award, Saudi Research Award, ISO 9001 Award, and TAMAKON Award. He has been involved in research in the field of computer science, and STEM education and evaluation. His current research interests include robotics and artificial intelligent of networks, network architectures, protocols, and future Internet of everything.

MEEJEONG LEE (Member, IEEE) received the B.S. degree in computer science from Ewha Womans University, Seoul, South Korea, in 1987, the M.S. degree in computer science from the University of North Carolina, Chapel Hill, in 1989, and the 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, Ewha Womans University, where she is currently a Professor. She has been involved in research in the field of computer communications and networks and performance modeling and evaluation. Her current research interests include protocols and architectures of mobile and wireless networks and future Internet.