

Tuft: Tree Based Heuristic Data Dissemination for Mobile Sink Wireless Sensor Networks

Omar Busaileh, Ammar Hawbani, Wang Xingfu, Ping Liu, Liang Zhao, and Ahmed Al-Dubai

Abstract—Wireless sensor networks (WSNs) with a static sink suffer from concentrated data traffic in the vicinity of the sink, which increases the burden on the nodes surrounding the sink, and impels them to deplete their batteries faster than other nodes in the network. Mobile sinks solve this corollary by providing a more balanced traffic dispersion, by shifting the traffic concentration with the mobility of the sink. However, it brings about a new expenditure to the network, where prior to delivering data, nodes are obligated to procure the sink's current position. This paper proposes Tuft, a novel hierarchical tree structure that is able to avert the overhead cost from delivering the fresh sink's position while maintaining a uniform dispersion of data traffic concentration. Tuft appropriates the mobility of the sink to its advantage, to increase the uniformity of energy consumption throughout the network. Moreover, we propose Tuft-Cells, a distributed dissemination protocol that models data routing as a Multi-Criteria Decision Making (MCDM) in three steps. To begin with, each criterion constitutes a random variable defined by a mass function. Each of these criterion serves a proportionately distinguishable alternative, and hence, may conflict. Therefore, the Analytic Hierarchy Process (AHP) quantifies the relationship between criteria. Finally, the final forwarding decision is derived by a weighted aggregation. Tuft is compared with state-of-the-art protocols, and the performance evaluation illustrates that our protocol adheres to the requirements of WSNs, in terms of energy consumption, and success ratio, considering the additional overhead cost brought by the mobility of the sink.

Index Terms—Wireless sensor networks, mobile sink, data dissemination, Multi-Criteria Decision Making.



1 INTRODUCTION

Recent technology shifts have enabled the design and deployment of large scale wireless sensor networks, where application scenarios vary from environment monitoring and fire detection systems to battlefield monitoring and surveillance. Wireless Sensor Networks (WSNs) comprise of a single or multiple sink nodes, alongside a vast number of sensor nodes that are randomly or systematically deployed in an area of interest. Each sensor node is designated to gather information from its vicinity and relay streams of data by either one or multiple hops to the sink for consequent processing [1]. These networks are burdened by constrained resources, as sensor nodes are battery-driven. On each detection of an event, sensor nodes consume energy by gathering the sensory data and delivering it to the sink [2]. New approaches such as energy harvesting are emerging for WSNs, where energy can be further supplied from deployed environments. However, harvesting energy to recharge the nodes' batteries may be hindered by the environment of the target area, where sensor nodes may be deployed in harsh environments such as oceans, near an active volcano or on a battlefield. Consequently, energy efficiency is classified as the most crucial issue for WSNs [3].

In WSNs with static sinks, the converge cast (many-to-one) traffic patterns of the network brings about an absence

of uniform dispersion for data traffic concentration, causing disruptions in the network topology and attenuates the sensing field coverage [4]. This is due to the high concentration of data traffic in the sink's vicinity, also known as the hotspot problem, compelling sensor nodes near the sink to relay additional data from distant nodes, leading to the depletion of their batteries faster than other nodes in the network. Inducing what is known as an *energy hole* around the sink, rendering it inaccessible [5].

Mobile sinks have been proposed to mitigate the hotspot problem by dealing with the inadequate network resources and attaining a more balanced traffic dispersion. Continuously shifting the traffic concentration in the sink's proximity with each consecutive move. Therefore, dispersing the energy consumption throughout the network and prolonging the network lifetime [6][7]. The use of a mobile sink can also be unavoidable in some application scenarios. For example, WSNs in battlefields are composed of sensor nodes deployed over an area of interest, sensor nodes are used to report the movement of enemy troops. Mobilized soldiers carry mobile sinks that constantly receive sensory data acquired by sensor nodes detecting enemy troops movement [8]. Moreover, mobile sinks can also help connect network nodes that are deployed in isolated environments [9].

Nevertheless, mobile sinks introduce a new expenditure to the network, due to the mobility of the sink; sensor nodes are obligated to procure the recent sink's position for each delivery of sensory data [10]. The continuous propagation resulting from the advertisement of the fresh sink's position exposes the network to a high overhead cost. Thus, a competent routing protocol for mobile sinks must adapt to the frequent change of the sink's location in an energy-efficient and scalable manner [11]. Essentially, traffic

- Omar B. (busaileh@mail.ustc.edu.cn), Ammar H., Xingfu W. { anmande, wangxfu } @ustc.edu.cn, and Ping L. (iacmy@mail.ustc.edu.cn), are with School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China; Ammar Hawbani and Xingfu Wang are the Corresponding authors.
- Liang Z. (lzhao@sau.edu.cn); Shenyang aerospace University; Shenyang, 110136, China;
- A.Al-Dubai is with School of Computing, Edinburgh Napier University; (a.al-dubai@napier.ac.uk);

introduced by advertising the fresh sink's position should be controlled in exchange for saving energy throughout the network. Moreover, an extreme increase in data delivery latency resulting from the constraints on the advertisement of the sink's position should be circumvented especially for real-time WSN applications.

Several routing protocols have been proposed to address the challenges brought by the mobility of the sink. The most efficient and widely adopted are the hierarchical mobile sink routing protocols [12]. Hierarchical approaches establish a virtual structure and impose varying roles on sensor nodes by incorporating one or multiple tiers. High-tier nodes procure the recent sink's position, while low-tier nodes query high-tier nodes to retrieve the sink's position. Hierarchical approaches are further classified into source-oriented protocols [13], and unique-structure protocols [14][15]. The former approaches propose that each source node constructs a virtual structure surrounding itself, and becomes the crossing point of that structure. However, the separate construction for each source node results in a high overhead cost that degrades the network's resources. The latter approaches propose a perpetual communal virtual structure. Despite reducing the overhead cost from building separate virtual structures, unique-structure protocols suffer from concentrated data traffic towards the high-tier nodes, exposing them to the hotspot problem.

To overcome the restrictions of the aforementioned protocols, we propose Tuft, Tree-Based Query Path for Data Dissemination, that exploits advantageous of hierarchical protocols and mitigates both the high overhead cost of separate constructions and the hotspot problem on high-tier nodes, by implementing a communal virtual spanning tree, and liberating the high-tier nodes to a more dynamic role within the structure. The spanning tree covers the whole network, each tree-node is composed of a *Cell*, where each *Cell* is a group of sensor nodes belonging to a certain area. The advertising and procuring of the fresh sink's position is achieved by traveling through the tree. Furthermore, high-tier nodes change intermittently with respect to the position of the sink, achieving a load balance of both data traffic concentration, and energy consumption on high-tier nodes, with low overhead cost. Tuft proposes *Tuft-Cells* protocol, a heuristic data dissemination protocol for routing. Based on our previous works [16][17], heuristic data dissemination protocols achieve efficiency in terms of energy consumption, energy balance, and waiting times. *Tuft-Cells* models data routing as a *Multi-Criteria Decision Making process*, where each alternative choice hinges on criteria that are modeled as random variables characterized by mass functions. Each criterion's mass function is extrapolated by applying curve fitting to data attained from extensive simulation experiments. Furthermore, these criteria may conflict with one other, therefore, are assigned a weight based on the Analytic Hierarchy Process in terms of various conditions that correspond with the requirements and impacts of the network layer. Tuft aims at solving the challenges brought by the mobility of the sink. Mainly, mitigating the hotspot on the high-tier nodes, lowering the overhead cost of delivering the sink's position and reducing & balancing the energy consumption for routing a packet. In turn, enhancing the energy consumed throughout the network and prolonging

the network lifetime. Tuft's novelty and major contributions are as follows:

1) Tuft exploits the mobility of the sink to its advantage, releasing the high-tier nodes in the spanning tree to a dynamic role that is able to procure the fresh sink's position with a stable low overhead cost, and mitigate the hotspot on the high-tier nodes. To further alleviate the premature depletion of batteries for some nodes, the mobile sink elects its *Access node* based on three metrics, that not only provides a balanced energy consumption, but also ensures the advertisement of the fresh sink's position remains reduced.

2) Tuft implements, *Tuft-Cells*, a heuristic data dissemination protocol that reinforces Tuft to provide a higher degree of uniformity for energy consumed throughout the network. *Tuft-Cells* attributes conditions in the network that comply with the requirements of the network layer, reducing and balancing the energy consumption for routing a packet.

3) Tuft is implemented using a self built on-purpose simulator that provides significant pieces of information visually, available online at: <https://github.com/howbani/tuft>.

The rest of this paper is organized as follows. Section 2 depicts the related works. Section 3 presents preliminaries and assumptions. Section 4 provides details on the proposed protocol. Section 5 explains the data dissemination protocol *Tuft-Cells*. The performance analysis, and performance evaluations are depicted in Section 6, and Section 7, respectively. Finally, Section 8 concludes this work.

2 RELATED WORKS

Mobile sinks in WSNs emanate frequent variations to the topology, which leaves the challenge for routing protocols to adapt to these variations in a balanced form without exhausting the network resources. Many hierarchical protocols have been proposed to address these challenges. Hierarchical protocols can be classified with respect to the established virtual structure formed by the high-tier nodes to, grid based [12][14], cluster based [15][18], tree based [19][13][20], and area based [21][22][23]. Extensive surveys are presented in [11] and [6]. These protocols employ two mechanisms to deliver data packets, first, the direct data dissemination, where low-tier nodes send the data packet to the high-tier nodes that relay it to the sink. Second, the Request-Response mechanism, where low-tier nodes query the high-tier nodes to obtain the sink's position. In the former cases, traffic concentration towards the high-tier nodes are increased, where the latter cases introduce an extra delivery delay.

To begin with, TTDD [12] proposes a virtual grid structure, where each source node constructs a grid around itself and becomes the crossing point of this grid. Whenever the sink requires data, local flooding within a grid-cell is employed to query the network. The query is then relayed towards the source node and data packets from source nodes are guided by the reverse path of the query. TTDD attributes to minimize the number of hops to reach the sink, however, suffers from a high overhead resulting from the separate grid construction for each source node. To avert the separate grid constructions, GBEER [14] imposed a communal grid structure for the whole network. Data announcements, and data requests are guaranteed to intersect at a crossing point in the grid. While GBEER managed to reduce the overhead

cost from the separate grid construction per source node, nevertheless, due to the data traffic concentration towards the nodes that compose the grid, these nodes will become hotspots and deplete their batteries faster than other nodes in the network.

HCDD [15] uses a clustering hierarchy, the virtual structure is formed by a Max-Min-D Cluster Formation algorithm. The highest level cluster headers called *Routing Agent* procure the sink's position by inter-cluster registration, and share it to the other *Routing Agents*. A source node will send data to the *Routing Agent* to be relayed to the sink by intra-cluster path. HCDD utilizes a distributed clustering algorithm where sensor nodes have no need of GPS devices, however the advertisement of the fresh sink's position, mainly the inter-cluster registration method imposes an increasing relation between the number of clusters, the speed of the sink and the number of advertisement packets, consuming network resources for large and sparse networks. CBDD [18] also utilizes a distributed clustering algorithm to construct the hierarchical structure. Like HCDD, CBDD uses inter and intra-cluster communication. Where the fresh sink's position is forwarded from one cluster head to all the other cluster heads in the network. Consequently, CBDD also suffers from an increase of advertisement packets from the inter-cluster communication rendering its problems towards scalability.

SEAD [13] employs a minimum-cost weighted Steiner tree rooted at the source node. Each source node constructs a virtual structure and the tree cost is reduced by selecting replicas at intermediate points that form the high-tier nodes. The mobile sink selects an *Access node* to act as its delegate. While the virtual structure results in a minimized number of hops to reach the sink, however, like TTDD the separate construction of trees for each source node incur a high overhead cost. QDD [20] proposed that the virtual hierarchical structure is constructed by a *Quad tree*. The network is recursively partitioned into four equally sized quadrants each considered as a separate parent. The rendezvous points of each quadrant forms the high-tier nodes of the structure. Compared to SEAD the overhead cost of QDD is lower, nevertheless, the data traffic concentration towards the high-tier nodes will result in the hotspot problem. To enhance on both SEAD, and QDD, STDD [19] proposes a data dissemination strategy that controls the advertisement of the fresh sink's position by establishing a dissemination tree that encapsulates all sensor nodes in the network. The *Access node* is designated as the root of the dissemination tree. Whenever the sink is out of the *Access node's* range only the first level of the tree is updated and maintained, therefore, overcoming the frequent topology changes introduced by the mobility of the sink.

Area based protocol LBDD [22] divides the sensor field into equivalent partitions by allocating a vertical line or strip. Sensor nodes within the boundaries of the vertical line act as rendezvous points and form the high-tier nodes. Sensory data is then sent towards the rendezvous area. The mobile sink queries the vertical line for data, where the sink's query is flooded inside the vertical line until it is received by the node that has the specified data. LBDD provides a simple virtual structure, however, the direct data dissemination employed by LBDD drives the low-tier nodes

to send all data packet on the high-tier nodes, this would increase the traffic concentration on the rendezvous area, and will cause high-tier nodes to become hotspots, and hence, decrease the network lifetime.

To mitigate the hotspot problem and delivery delay in LBDD, Ring Routing [21] proposed a virtual one node-width ring-structure that confine the network center where ring nodes form the high-tier nodes. Moreover, Ring Routing appropriates an *Anchor node* to the mobile sink, delivery of the fresh sink's position is achieved by sending an information packet from the sink's *Anchor node* to the high-tier nodes, the query is then shared along the ring nodes. Source nodes can then query the high-tier nodes to procure the sink's position. While Ring Routing proposes a method for mitigating the hotspot problem on the high-tier nodes by changing Ring nodes roles with regular nodes, however, this would lead to either expanding or collapsing the ring. In the former case, the expansion of the ring increases the number of ring nodes, thus, incrementing the hops of the path for delivering the fresh sink's position and consuming more energy, especially when the mobile sink's speed is high. In the latter case, the collapsing of the ring decreases the number of ring nodes, thus, increasing the concentration of data traffic to a smaller number of nodes in the center of the network, which leads to the hotspot problem. Other protocols [8][24] utilize a controllable mobile sink, which is out of scope of this paper. We focus on random mobile sink patterns since it is more applicable to WSNs.

As opposed to the aforementioned protocols, we propose Tuft that is capable of mitigating the problems induced by the tree based and area based protocols, by utilizing the mobility of the sink to its advantage. Tuft employs a communal spanning tree, where each tree-node is a *Cell* that encapsulates a group of sensor nodes. The spanning tree bounds the advertisement of the fresh sink's position, ensuring the overhead cost remains reduced and balanced. The mobile sink elects an *Access node* to act as its delegate and source nodes procure the *Access node's* position by simply accessing the tree. The roles designated for nodes are dynamic and change with low overhead cost. Finally, Tuft implements *Tuft-Cells*, a heuristic data dissemination protocol that conforms with the requirements of the network layer, and aims to lower the energy consumed to route a packet, while maintaining a higher degree of uniformity for the consumed energy.

3 PRELIMINARIES

3.1 Assumptions

We define the network by the set \mathbb{N} that consists of m sensors deployed randomly using a uniform distribution in the sensing field A , where $\mathbb{N} = \{n_0, n_1, \dots, n_m\}$. Each sensor is aware of its geographic location by means of a localization scheme such as [25] or is equipped with a GPS device. The sink's mobility is modeled after the *Gauss-Markov Mobility Model*, while other nodes in the network are static. Sensor nodes belonging to the network \mathbb{N} are homogeneous and have limited communication range δ & initial battery power. The *Euclidean Distance* between two nodes n_i , and n_j is denoted by $d_{i,j}$. A node $n_i \in \mathbb{N}$ has coordinates (x_i, y_i) , and is aware of the position of its neighbors using any

Neighbor Discovery strategy such as the one in [26]. The set of neighbors for a node $n_i \in \mathbb{N}$ is the subset $\mathbb{N}_i \subseteq \mathbb{N}$ such that $\mathbb{N}_i = \{n_j | n_j \in \mathbb{N} \text{ } d_{i,j} \leq \delta\}$, and m_i is the number of neighboring nodes for n_i . The most used notations are further listed in Table 1.

At the initialization level, the sink gathers the information of all the nodes in the network to perform the construction of the virtual structure by distributing different designated roles for sensor nodes. The density of the network \mathcal{D} is defined in (1), where it consists of two terms, the first being the expected average number of neighboring nodes for each node (average number of nodes located within the communication range of each node), the second reflects the degree of connection for each node (the connection that a node has to other nodes, i.e. it's neighboring nodes).

Finally, we define the maximum number of hops for a packet in (2), where $d_{s,d}$ is the *Euclidean distance* between the source and destination, and δ is the nodes' communication range.

$$\mathcal{D} = \frac{1}{2} \left(\frac{\pi m \delta^2}{A} + \frac{\sum_{i=0}^m m_i}{m} \right) \quad (1)$$

$$Mx = \frac{\mathcal{D}}{2} \times \frac{d_{s,d}}{\delta} \quad (2)$$

TABLE 1: Notations

Notation	Description
A	The area of the sensing field, $a \times b$.
\mathcal{D}	The network density.
\mathbb{N}	The network consisting of m nodes.
\mathbb{N}_i	The set of neighboring nodes of node n_i , where m_i is the size of \mathbb{N}_i .
n_b	The sink node
n_s	The source node
\mathcal{C}	The actual network center.
$\hat{\mathcal{C}}$	The computed network center.
$d_{i,j}$	The Euclidean distance between n_i and n_j .
ℓ_i	Cell i in the virtual structure.
\mathcal{R}	The Cell's radius.
\hat{c}_i	Cell ℓ_i 's center.
$D_{i,j}$	The Euclidean distance between the two Cells' center \hat{c}_i , and \hat{c}_j .
$depth_i$	The depth of Cell ℓ_i in the tree structure.
\mathcal{H}	The number of hops taken for a packet traveling from source to destination.

3.2 Mobility Model

Used in ad hoc network protocol simulations, the Gauss-Markov mobility model acclimates the degree of randomness by using a tuning parameter that varies the degree of randomness in the mobility pattern [27]. This mobility model offers a more realistic movement behavior, where it solves the problems of the sudden stop and the sharp turn [28]. The mobile sink is assigned an initial mean value of speed s and direction r . The motion of the mobile sink is then achieved by updating the values of both the speed and the direction. The freshly acquired values s_t, r_t at a certain instance t depend upon: the previous instance's values (i.e. s_{t-1}, r_{t-1}), the tuning parameter (to acclimate the degree of randomness), the mean value of both speed & direction, and a random variable that follows a Gaussian distribution.

s_t, r_t are obtained using (3) and (4) respectively, where α is the tuning parameter such that ($0 \leq \alpha \leq 1$), \bar{s}, \bar{r} are constants that represent the mean values, and s_{xt-1}, r_{xt-1} are random variables that follow a Gaussian distribution. Then the new location (x_t, y_t) of the mobile sink at the instance t is obtained using (5).

At the simulation level, the movement of the sink is bounded within the sensing field. Whenever the sink is within a threshold distance to one of the edges of the network, it is forced away from the edge. This is mainly achieved by adjusting the mean value of direction \bar{r} . Incorporating Gauss-Markov mobility model will introduce uncertainty in the simulation environment, making the location of the sink hard to predict and controlled.

$$s_t = \alpha s_{t-1} + (1 - \alpha) \bar{s} + \sqrt{(1 - \alpha^2)} s_{xt-1} \quad (3)$$

$$r_t = \alpha r_{t-1} + (1 - \alpha) \bar{r} + \sqrt{(1 - \alpha^2)} r_{xt-1} \quad (4)$$

$$\begin{aligned} x_t &= x_{t-1} + s_{t-1} \cos r_{t-1} \\ y_t &= y_{t-1} + s_{t-1} \sin r_{t-1} \end{aligned} \quad (5)$$

3.3 Energy Consumption Model

In WSNs, sensory data are collected in the sink. Sensor nodes can send their data by one or multiple hops. The energy consumed to deliver the data from a source node n_s to the sink n_b is considered a crucial problem. We consider the path $\mathbb{P}_{n_s}^{n_b}$ as the path taken to deliver a data packet from the source node to the sink. $\mathbb{P}_{n_s}^{n_b}$ can be a one hop path from the source to the destination, or a multiple hop path from the source node going through intermediary nodes to reach the destination. In conformity with the *First Order Radio Model* [29], the energy consumed to transmit and receive a data packet of size k from node n_i to node n_j are expressed in (7), and (8), respectively. Where E_{elec} (the energy dissipated to run the node's transmitter or receiver circuitry), ε_{fs} & ε_{mp} (amplifier energy) are constants, and d_* is the *Threshold Distance* acquired from (6). Hence, the energy consumed to transmit and receive a data packet of size k in each hop $hop(i, j)$ of the path $\mathbb{P}_{n_s}^{n_b}$ can be expressed in (9).

$$d_* = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \quad (6)$$

$$T_x(i, j, k) = \begin{cases} (k \cdot E_{elec}) + (k \cdot \varepsilon_{fs} \cdot d_{i,j}^2) & d_{i,j} < d_* \\ (k \cdot E_{elec}) + (k \cdot \varepsilon_{mp} \cdot d_{i,j}^4) & d_{i,j} \geq d_* \end{cases} \quad (7)$$

$$R_x(j, k) = k \cdot E_{elec} \quad (8)$$

$$e_{hop}(i, j) = \begin{cases} (2k \cdot E_{elec}) + (k \cdot \varepsilon_{fs} \cdot d_{i,j}^2) & d_{i,j} < d_* \\ (2k \cdot E_{elec}) + (k \cdot \varepsilon_{mp} \cdot d_{i,j}^4) & d_{i,j} \geq d_* \end{cases} \quad (9)$$

The overall energy consumed to transmit and receive the data packet that traveled along the path $\mathbb{P}_{n_s}^{n_b}$ is defined in (10) where \mathcal{H} is the number of hops needed to reach the destination.

$$e_{path} = \sum_{\forall i, j \in \mathbb{P}_{n_s}^{n_b}} e_{hop}(i, j) \quad (10)$$

4 PROPOSED PROTOCOL

We propose Tuft, a novel hierarchical protocol based on a tree structure for mobile sink WSNs. The virtual structure composes of *Cells*. Each *Cell* encapsulates a group of sensor nodes contingent to their relative position in the network as shown in Fig. 1a (nodes confined within the grey area). The tree structure is then formed by virtually coupling the *Cells* together starting from the root node of the tree defined as the *Root-Cell* branching out to reach the leaf nodes of the tree defined as *Leaf-Cells*.

The protocol designates each sensor node with an appointed role in the network, varying from *regular nodes*, *Cell nodes*, to *Cell-Leader nodes*, and *Access node*. *Cell nodes* are sensor nodes encapsulated in a *Cell*. The *Cell-Leader node* is a node amongst the *Cell nodes* appropriated as a gateway between *Cells*. The mobile sink appoints the nearest *Cell* to itself as the *Root-Cell*. Hence, the *Root-Cell* is dynamic pertaining the mobile sink's position. The *Root-Cell Leader* acquires and stores the fresh sink's position. Where source nodes can simply access the structure to procure the position information of the sink. Finally, the *Access node* acts as a delegate for the mobile sink connecting it to the rest of the network.

Tuft's structure contributes to advertise, and procure the fresh sink's position with low overhead cost, while maintaining a degree of uniform dispersion for energy consumed throughout the network, this is achieved by the following: the virtual structure is built centrally in the sink, nodes are designated with their prospective roles to realize the structure, and the maintenance mechanism ensures a balanced, and stable low overhead cost. Finally, to attribute for the endurance of the high-tier nodes, a dynamic role within the *Cell* is implemented. Moreover, the election of the *Access node* involves three metrics to provide a higher degree of uniformity for the consumed energy. .

4.1 Virtual Structure Construction

The structure is a spanning tree, and each tree-node is represented by a *Cell*. After network deployment, information of sensor nodes are gathered in the sink. The computed network center $\bar{\mathcal{C}}$ defined by coordinates (\bar{x}, \bar{y}) is then computed using (11), where p_i is the relative position of node n_i , and \mathcal{C} is the actual center of the network. The computed network center reflects both the relative position of each node, and the actual network center, providing a more dynamic starting point that depends on the actual position of nodes. Then we obtain the *Cell radius* \mathcal{R} based on the communication range δ to initiate the first stage of the structure construction that begins with the prerequisite computed network center, and attains *Cells* covering the x , y & *diagonal* axis of the sensing field. Subsequently, this is followed by a refinement stage to cover the whole network.

$$\bar{\mathcal{C}} = \frac{1}{2} \left(\mathcal{C} + \sum_{i=0}^m p_i \right) \quad (11)$$

4.1.1 Initial Construction

A *Cell* ℓ_i will occupy an area of \mathcal{R}^2 and will be centered at \hat{c}_i . *Cells* are half a *Cell radius* apart, and sensor nodes

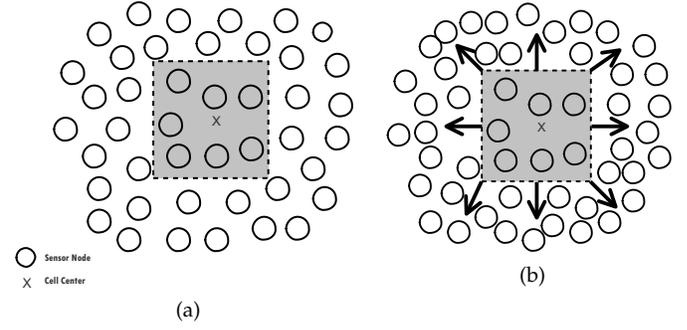


Fig. 1: (a) The *Cell*, and the grey area that covers the *Cell nodes*; (b) The eight directions, represented by the arrows.

encapsulated inside the area of a *Cell* are designated as *Cell nodes*. To begin with, each *Cell's* width will be two hops (i.e. $\mathcal{R} = 2\delta$), and *Cells* are one hop apart from there adjacent *Cells* (i.e. $\frac{1}{2}\mathcal{R} = \delta$). Accordingly, the *Cell radius* is determined by (12).

$$\mathcal{R} = 2\delta \quad (12)$$

Then the starting point is denoted as the computed network center. Hence, the first *Cell* ℓ_1 will be centered at $\bar{\mathcal{C}}$ (i.e. $\hat{c}_1 = \bar{\mathcal{C}}$). Then using \hat{c}_1 and in eight different directions as seen in Fig. 1b the center \hat{c}_j for $j > 1$ of the next *Cell* is determined by an increment or a decrement to the x and y values of \hat{c}_1 by a $\frac{3}{2}\mathcal{R}$ (since adjacent *Cells* are $\frac{1}{2}\mathcal{R}$ apart, their network center's will be $\frac{3}{2}\mathcal{R}$ apart).

This method is repeated until the network's x , y & *diagonal* axis are covered by *Cells* encapsulating a group of sensor nodes surrounded by *regular nodes* as seen in Fig. 2. Algorithm 1 further illustrates the initial construction stage.

Algorithm 1 Initial Cells Construction

Input: Sensor nodes.

Output: *Cells* covering the x, y , and *diagonal* axis of the sensing field.

```

1: procedure INITIALCELLSCONSTRUCTION
2:   Compute the network center  $\bar{\mathcal{C}}$  by (11)
3:   Define the initial radius  $\mathcal{R}$ 
4:   Start from  $\hat{c}_1 = \bar{\mathcal{C}}$ 
5:   Get all sensor nodes  $\mathbb{B}_1$  in  $\ell_1$ 's area and assign the
   role of Cell nodes
6:   Construct the first Cell  $\ell_1$ 
7:   for each direction do // As seen in Fig. 1b.
8:     do
9:       Increment or decrement  $\hat{c}_1$  to get  $\hat{c}_j$  for  $j > 1$ 
10:      Get all sensor nodes  $\mathbb{B}_j$  in  $\ell_j$ 's area and
      assign the role of Cell Nodes
11:      If Cell  $\ell_j$  is not empty
12:        Construct  $\ell_j$ 
13:      while inside sensing field
14:    end for
15: end procedure

```

4.1.2 Network Refinement

After the initial construction, since the *Cell radius* defines the distance between *Cells*, and is independent of both the size

and the density of the network, there may be gaps between *Cells* on the *diagonal* and *x* axis of the sensing field. This is further illustrated in Fig. 2 showing two different cases; Fig. 2a the initial construction covered the whole network with *Cells*, since the network size is smaller than in Fig. 2b, where gaps are clearly seen adjacent to the diagonal *Cells*. Therefore, network refinement will compensate *Cells* to fill the voids.

As the construction in the initial phase goes through eight directions from the starting point, *Cells* on the *x* and *y* axis of the sensing field illustrated in Fig. 2b and denoted by \mathbb{S} will need refinement. The farther the nodes are from the starting point, the more the gaps that need refining. Therefore, the distance $D_{i,1}$ between each *Cell's* center in \mathbb{S} and the starting point is proportional to the number of *Cells* to be added. *Cells* on the *x* axis will compensate *Cells* under or above it, while *Cells* on the *y* axis will compensate *Cells* adjacent to it. Accordingly, the number of *Cells* k that are to be compensated is obtained using (13). Thereafter, the position of each compensated *Cell* will be an increment or decrement of the *x* or *y* coordinate of its adjacent *Cell's* center \hat{c}_i . Finally, the nearest *Cell* node to the *Cell* center is elected as a *Cell-Leader* for each *Cell*. Algorithm 2, further illustrates the network refinement stage, where in the case of Fig. 2a it will proceed to the election of *Cell-Leaders*, and in Fig. 2b, it will fill the void then elect *Cell-Leaders*, the resulting network of Fig. 2b is illustrated in Fig. 2c.

$$k = \frac{D_{i,1}}{\frac{3}{2}\mathcal{R}} - 1 \quad (13)$$

Algorithm 2 Refining the Network.

Input: The group of *Cells* \mathbb{S} that need refining.

Output: *Cells* covering the gaps, and assigns *Cell-Leader*

```

1: procedure NETWORKREFINING
2:   for each  $\ell_j \in \mathbb{S}$  do
3:     Get  $k$  using (13)
4:     Use Algorithm 1 for  $k$  Cells in two directions
5:   for each  $\ell_i$  in the sensing field do
6:     SELECTCELLLEADER( $\ell_i$ )
7:   end for
8: end procedure
9:
10: procedure SELECTCELLLEADER(Cell  $\ell_i$ )
11:   for each sensor node  $n_j \in \mathbb{B}_i$  do
12:     Select  $n_j$  nearest to  $\hat{c}_i$  & assign role of Cell-Leader
13:   end for
14: end procedure

```

4.1.3 Tree Coupling

At the initialization level, each *Cell* keeps a link with its neighboring *Cells* (i.e. *Cells* at a distance less than or equal to $\frac{3}{2}$ of the *Cell* radius from its *Cell* center). At first entry of the mobile sink, the nearest *Cell* to it would be designated as the *Root-Cell* and all its neighboring *Cells* will become its Children, also called *Level-One Cells*. Similarly, *Level-One Cells* define their Children from their neighboring *Cells*. This process is repeated until *Leaf-Cells* are reached and a tree

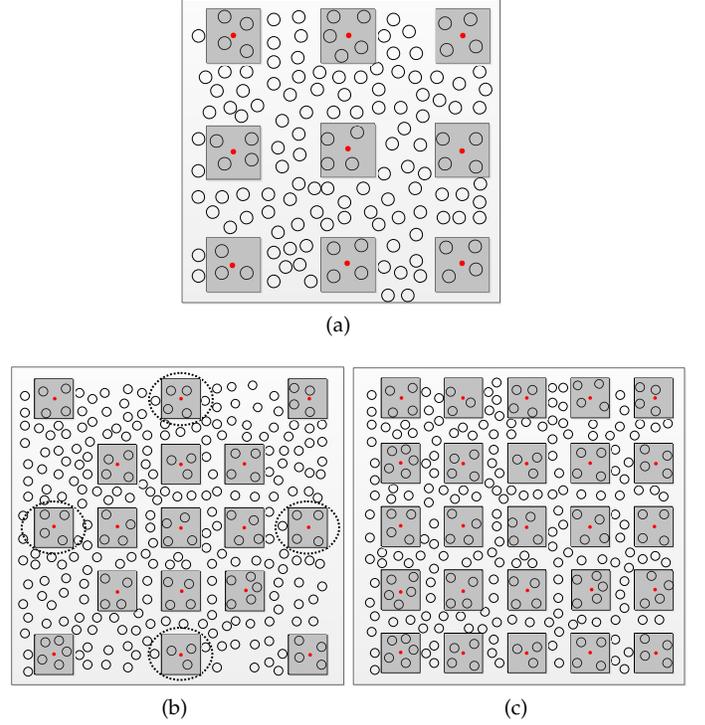


Fig. 2: The results after initial construction. (a) Topology one has no need of network refinement. (b) Topology two shows the gaps between *Cells*. (c) Topology two after network refinement.

structure rooted at the nearest *Cell* to the mobile sink is formed.

Thus, we obtain a dynamic tree structure, where the high-tier nodes are able to change with each movement threshold of the mobile sink, dispersing the data traffic to achieve a better balance in terms of energy consumption and mitigating the hotspot problem on the high-tier nodes of the structure. The construction phase is unambiguous, diversifying each sensor node's role. The mechanism behind the spanning tree's update is further explained in Subsection 4.6.

4.2 Access Node Selection

The mobile sink emanates periodic changes to the topology of the network, to hide these frequent alternations an emissary for the mobile sink is employed. The *Access node* connects the mobile sink to the rest of the network and hides the periodic changes that arise from the mobility of the sink. Therefore, the sink has to stay connected to an *Access node* at all times.

The sink elects an *Access node* from its neighboring nodes. The process of selecting an *Access node* depends on three metrics. To start with, a future sink location is predicted using the current speed and direction of the mobile sink. Accordingly, the distance between each neighboring node of the sink and the sink's future location is obtained. Then, the distance between each neighboring node of the sink and the *Root-Cell* center is computed. Finally, the residual energy of each neighboring node is obtained to assign a priority

to each neighboring node. The priority of each node is the normalized value of the sum of the three metrics.

The mobile sink would then send an *Access-Node Selection (AN)* packet (that encapsulates both the position information of the sink, and the nearest *Cell's* center to it), to the node with the highest priority. Prior to leaving the communication range of the *Access node* the sink would use the same method to elect a new *Access node* and send a packet to both the new and the old *Access node*. Informing the old *Access node* of the new *Access node's* position information. This mechanism is referred to as the follow-up mechanism, and it establishes a link between the mobile sink's last emissary with it's new emissary by progressively advertising the sink's footprints along the path it moves on. This is also known as the progressive footprint chaining [30].

Appropriating the *Access node* and arming it with the follow-up mechanism, packets destined to the mobile sink can be simply delivered to it's intended destination through two cases. First case, if the current *Access node* receives the packet, the packet would be relayed to the mobile sink directly. Second case, if the old *Access node* receives the packet, the packet would be essentially relayed to the new *Access node* to be directly relayed to the sink.

4.3 Sink Position Delivery

The *Access node* is also designated to inform the nearest *Cell-Leader* to the mobile sink to act as the new *Root-Cell*. Hence, after receiving the *AN* packet, the *Access node* sends a *Tree-Update and Advertisement (TUA)* packet destined to the *Cell center* encapsulated in the received *AN* packet. The *TUA* packet informs the *Cell-Leader* of its new role (*Root-Cell Leader*), and the new position information of the *Access node*. Algorithm 3 illustrates the process of *Access node* selection and sink position advertisement.

Algorithm 3 Advertisement of the fresh sink's position.

Input: The sink's neighboring nodes.

```

1: procedure ACCESSNODESELECTION
2:   Obtain the future sink location
3:   for each  $n_j \in \mathbb{N}_b$  do
4:     Get final priority according to the
5:     three metrics. //Explained in Subsection 4.2
6:   end for
7:   Select Access node  $n_j$  where  $n_j \in \mathbb{N}_b$  and has the
   highest priority
8:   INFORMACCESSNODE(AN)
9: end procedure
10:
11: procedure INFORMACCESSNODE(AN PACKET)
12:   Assign role of Access node.
13:   Inform Cell's Leader of the position information of the
   Access node and new role
14: end procedure

```

The Sink Position Delivery mechanism is utilized to both ensure the mobile sink is connected to the *Access node* at all times, and deliver the fresh sink's position to the *Root Cell-Leader*. Moreover, the dynamic change of the *Root-Cell* is achieved by receiving the *TUA* packet from the *Access node* which updates the tree structure's hierarchy.

4.4 Obtaining the Sink's Position

The advertisement of the fresh sink's position is constrained in the virtual tree structure. Algorithm 4 and Algorithm 5 illustrate the sending and the receiving of the *Query-Request* packet, respectively. The varying roles of sensor nodes in the network allow the *Query-Request* packet to traverse from the source node n_s through the tree structure to reach its intended destination (*Root-Cell Leader*).

Algorithm 4 Procuring the fresh sink's position.

Input: Source node n_s .

```

1: procedure SEND QUERY-REQUEST PACKET
2:   if  $n_s$  is a Cell node then
3:     if  $n_s$  is a Cell-Leader then
4:       if  $n_s$  is not a Root-Cell Leader then
5:         Send Query-Req packet to Parent Cell
6:       else
7:         Send Query-Req packet to Cell-Leader
8:     else
9:       Send Query-Req packet to nearest Cell
10: end procedure

```

Algorithm 5 Receive Query Request.

Input: Recieve Query-Request packet by node n_i .

```

1: procedure RECIEVE QUERY-REQUEST PACKET
2:   if  $n_i$  is a Cell node then
3:     if  $n_i$  is a Cell-Leader then
4:       if  $n_i$  is a Root-Cell Leader then
5:         Generate Query-Resp to source.
6:       else
7:         Relay Query-Req to Parent-Cell center
8:     else
9:       Relay Query-Req to my Cell-Leader
10:   else
11:     Relay Query-Req to destination
12: end procedure

```

4.5 Data Dissemination

Once a source node receives the *Query-Response* packet. The data packet is sent towards the *Access node*. If the data packet is received by the old *Access node*; it is relayed towards the new *Access node* using the follow-up mechanism. Finally, the current *Access node* can send the data packet to the mobile sink. An example for the data dissemination steps is illustrated in Fig. 3. The network's spanning tree hierarchy is illustrated in Fig. 4b.

4.6 Maintaining the Tree

The consistent movement of the mobile sink coincides with a proportional update to the tree structure. Hence, dispersing the data traffic on the hotspots of the network. Whenever, the sink moves away from its current *Root-Cell's* center by more than a $\frac{3}{4}\mathcal{R}$, or is about to exit the current *Access node's* transmission range. The sink would check the nearest *Cell* to it and anoints it as the new *Root-Cell*. This

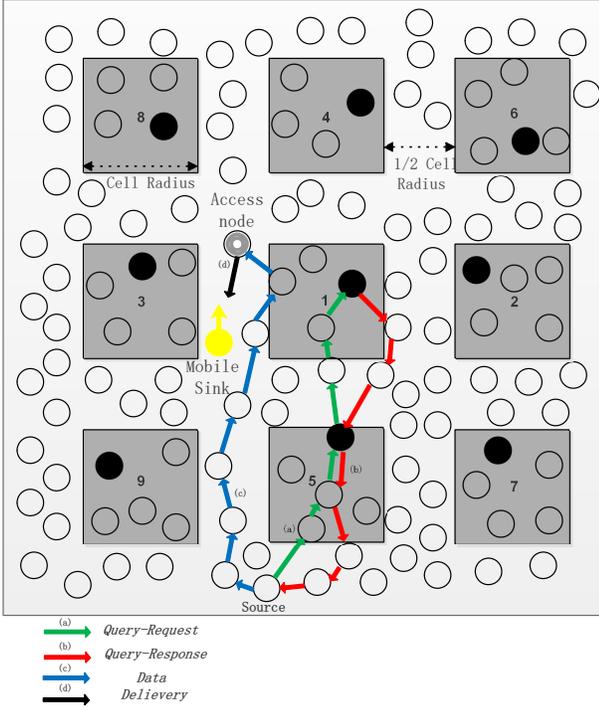


Fig. 3: Data dissemination steps in Tuft. The *Cell-Leaders* are represented by the sensor nodes filled with a black color. (a) The source node sends a *Query-Request* packet towards its nearest *Cell* ℓ_5 , which is then relayed towards the *Root-Cell* ℓ_1 . (b) The *Root-Cell Leader* would reply back with a *Query-Response* packet. (c) The data packet is sent towards the *Access Node*, (d) that relays it to the sink.

is achieved in two cases: (1) If the sink remains in the transmission range of the current *Access node*, the sink sends a new *AN* packet informing the current *Access node* of the newly appointed *Root-Cell*. (2) If the sink is about to exit the current *Access node's* transmission range, then a new *Access node* is chosen and is informed of the newly pointed *Root-Cell*. Subsequently, the new or current *Access node* informs the new *Root-Cell* of its designated role. This triggers an update to the tree structure's *Root-Cell* and *Level-One Cells*. The mechanism behind maintaining the tree resembles the maintenance phase used in our previous work [19].

Initially, the newly appointed *Root-Cell* will become the new root of the spanning tree, equivalently, the old *Root-Cell* becomes its direct *Child*. Moreover, the shared links between the diagonal *Children* of the old *Root-Cell* and the neighboring *Cells* of the new *Root-Cell* become the new *Root-Cells's Children*. Fig. 4 gives an example to further illustrate the spanning tree's maintenance process. In Fig. 4b the old *Root-Cell* is ℓ_1 , and the newly appointed *Root-Cell* is ℓ_2 . *Cells* ℓ_6 and ℓ_7 are the old *Root-Cell's* diagonal *Children* and neighbor *Cells* of the new *Root-Cell*. Hence, the tree structure becomes as the one seen in Fig. 4c Every *Level* change of the tree structure is achieved by sending a *Maintenance* packet to update the hierarchy of *Cells*.

4.7 Cell Leader Election

In each *Cell* the *Cell-Leader* is designated to handle *Query Requests* and maintain the tree structure and hence, is bound

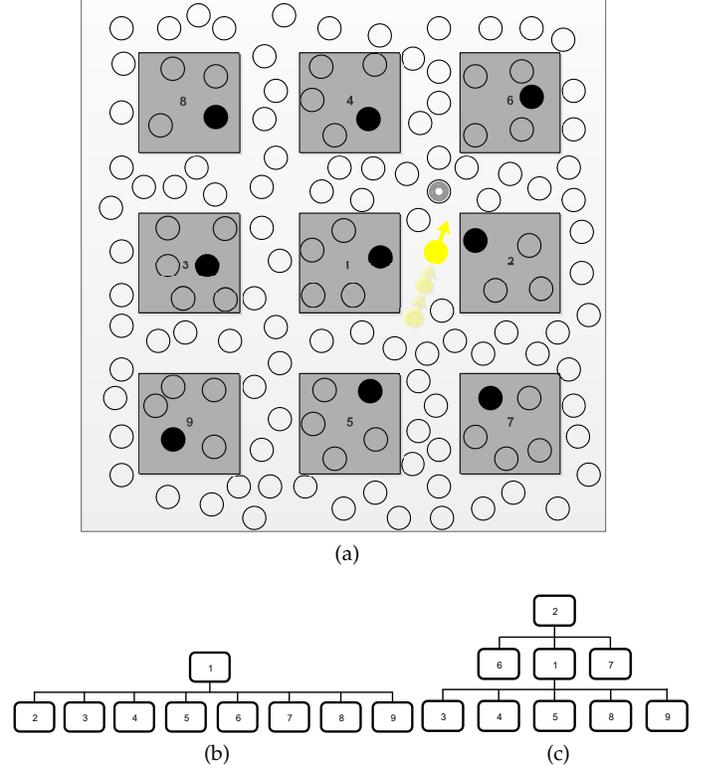


Fig. 4: (a) The movement of the mobile sink. (b) The tree structure with the sink close to *Cell* ℓ_1 . (c) The tree structure with the sink close to *Cell* ℓ_2 .

to consume more energy than other nodes in the *Cell*. Therefore, *Cell Leader Election* guarantees the change of the *Cell-Leader* after a certain threshold of battery level. In turn, a more balanced energy consumption in the *Cell* is achieved.

Whenever a *Cell-Leader* reaches a certain battery level threshold, it sends beacon-packet to its neighboring nodes within the *Cell*. The *Cell node* with the highest remaining energy is elected as the new *Cell-Leader*. This ensures that the residual energy of the newly appointed *Cell-Leader* isn't already depleted. The new *Cell-Leader* will, in turn, send beacon-packet to its neighboring nodes within the *Cell* informing them of its new role. Hence, whenever a packet intended to the *Cell-Leader* and destined for the *Cell* center enters the *Cell*, it is relayed to the current *Cell-Leader*.

5 TUFT-CELLS

Tuft-Cells' data routing is modeled as a *Multi-Criteria Decision Making process* (MCDM). In MCDM, alternative decision making hinges on criteria or attributes. Each criteria serves a proportionately distinguishable alternative and may conflict with each other. Based on our previous work [16][17] there are three primary factors that are directly related to the energy consumed for routing a packet, transmission distance, direction angle, and perpendicular distance. These three factors are all directly proportional to the energy consumed for routing a packet. Moreover, energy balance is considered a crucial factor for an adequate routing protocol. Consequently, the energy balancing factor that measures the residual energy of the nodes alleviates the premature

energy depletion of some nodes. To attain efficient routing in terms of energy consumption and energy balance, *Tuft-Cell's* MCDM process is subject to four criteria, transmission distance, direction angle, perpendicular distance, and residual energy. Each criterion is modeled as a random variable characterized by a mass function defined by applying interpolation (curve fitting) to data attained from extensive simulation experiments. Individually, each criterion introduces conflicting alternatives to the decision making, therefore, a weighted aggregation identifies the relationship between criteria by assigning an evaluation weight to each criterion based on the Analytic Hierarchy Process (AHP).

Various conditions define different weights for each criterion corresponding to the requirements and impacts of the network layer. Whenever a source node has a packet to send, it defines its choices (forwarder set) as all its neighboring nodes. Subject to the specific condition, the weight of each criterion is defined to derive the weighted aggregation. The source node would then prioritize its choices in conformity with their weighted aggregation. Finally, the source node designates the highest priority choices as alternatives (candidates). The rest of the section explains the criteria and the decision making.

5.1 Criteria

To provide *Tuft-Cells* with tolerability; four criteria that adhere to the reduction and balance of the energy consumed for routing a packet are used. Each criterion is a random variable characterized by a mass function obtained from previous extensive simulation experiments conducted in [16][17].

5.1.1 Transmission Distance

The transmission distance denoted by $d_{i,j}$, is defined as the *Euclidean distance* between the sender node n_i and its neighboring node n_j (for all $n_j \in \mathbb{N}_i$). The mass function for the transmission distance $\tilde{d}_{i,j}$ is defined in (14). $\tilde{d}_{i,j}$ gives higher values for shorter transmission distances. This succumbs to the fact that energy consumed for farther nodes is higher than nearer nodes. However, only considering this criterion, the path taken by a packet from the source node to the destination will undertake more hops and hence, consume more energy as explained in the Subsection 3.3.

$$\tilde{d}_{i,j} = \frac{e^{-(d_{i,j})}}{\sum_{k=0}^{m_i} e^{-(d_{i,k})}} \quad n_j \in \mathbb{N}_i \quad (14)$$

5.1.2 Direction Angle

The direction angle denoted by $\theta_{i,j}$, is the angle formed between the destination, the sender node n_i and its neighboring node n_j , where the destination in *Tuft-Cells* can simply be a (x, y) coordinate. The mass function for the direction angle $\tilde{\theta}_{i,j}$ is defined in (15). $\tilde{\theta}_{i,j}$ gives higher values to lower direction angles since lower direction angles imply a closer node to the destination and the sender node. Considering this criterion alone will lack a dispersion of the energy consumed in the network, therefore, will conflict with the energy balancing factor.

$$\tilde{\theta}_{i,j} = \frac{1}{e^{\theta_{i,j}} \sum_{k=0}^{m_i} \frac{1}{e^{\theta_{i,k}}}} \quad n_j \in \mathbb{N}_i \quad (15)$$

5.1.3 Perpendicular Distance

The perpendicular distance denoted by $\psi_{i,j}$, is determined as the distance between the neighboring node n_j and the virtual line that connects the source node n_s and the destination. The mass function for the perpendicular distance $\tilde{\psi}_{i,j}$ is obtained in (16). $\tilde{\psi}_{i,j}$ gives higher priority to shorter perpendicular distances. However, considering this criterion alone, will also conflict with the efficiency and balance of the performance of the routing protocol.

$$\tilde{\psi}_{i,j} = \frac{e^{-(\psi_{i,j})}}{\sum_{k=0}^{m_i} e^{-(\psi_{i,k})}} \quad n_j \in \mathbb{N}_i \quad (16)$$

5.1.4 Residual Energy

The residual energy denoted by $\Phi_{i,j}$, is defined as the remaining energy for a node. The mass function for the residual energy $\tilde{\Phi}_{i,j}$ for the sender node n_i 's neighboring node n_j is expressed in (17). $\tilde{\Phi}_{i,j}$ gives higher priority to the nodes with higher residual energies. Using this criterion alone, conflicts with the the energy consumption factor of the network.

$$\tilde{\Phi}_{i,j} = \frac{e^{(\Phi_{i,j})}}{\sum_{k=0}^{m_i} e^{(\Phi_{i,k})}} \quad n_j \in \mathbb{N}_i \quad (17)$$

5.2 Decision Making

5.2.1 Weighted Aggregation

A sender node n_i evaluates all its choices (neighboring nodes) to define its alternatives (candidate set). Each choice is evaluated by a weighted aggregation obtained in (18) where w_{cr} is the weight of each criterion cr explained in Subsection 5.2.2. Then $\bar{R}_{i,j}$ is normalized to $\tilde{R}_{i,j}$ using (19).

$$\bar{R}_{i,j} = (w_d \cdot \tilde{d}_{i,j}) + (w_\theta \cdot \tilde{\theta}_{i,j}) + (w_\psi \cdot \tilde{\psi}_{i,j}) + (w_\Phi \cdot \tilde{\Phi}_{i,j}) \quad (18)$$

$$\tilde{R}_{i,j} = \frac{\bar{R}_{i,j}}{\sum_{k=0}^{m_i} \bar{R}_{i,k}} \quad (19)$$

5.2.2 Weights

Varying conditions of the network call for alternative decision making. Hierarchical structures impose designated roles on the nodes of the network, which induces some to process more packets than other. Therefore, in order for *Tuft* to obtain a higher degree of uniform energy dispersion, some criteria will be given higher priority than others, resulting in different alternatives with each different condition. AHP defines the relationship between conflicting criteria by incorporating the importance of each criterion under an explicit condition. A relative weight assigned to each criterion that produces a different alternative is the core of AHP.

$$M = \begin{matrix} & d & \theta & \psi & \Phi \\ \begin{matrix} d \\ \theta \\ \psi \\ \Phi \end{matrix} & \begin{bmatrix} a_{d,d} & a_{d,\theta} & a_{d,\psi} & a_{d,\Phi} \\ a_{\theta,d} & a_{\theta,\theta} & a_{\theta,\psi} & a_{\theta,\Phi} \\ a_{\psi,d} & a_{\psi,\theta} & a_{\psi,\psi} & a_{\psi,\Phi} \\ a_{\Phi,d} & a_{\Phi,\theta} & a_{\Phi,\psi} & a_{\Phi,\Phi} \end{bmatrix} \end{matrix} \quad (20)$$

The relative weights assigned to each criterion are computed based on evaluation matrices that compose of pairwise comparison between criteria, higher weights imply higher importance. Each entry $a_{j,k}$ in the evaluation matrix

illustrated in (20) represents the degree of importance for criterion j to that of criterion k . The implications for the degree of $a_{j,k}$ are explained in Table 2. The steps of derivation for the relative weights of each criterion are derived the same as in [31].

TABLE 2: Implications for the degree of $a_{j,k}$

Degree of $a_{j,k}$	Implication
1	j is equally important to k
3	j is lightly more important than k
5	j is more important than k
7	j is highly more important than k

To define the basis of varying conditions, we look at a *Query-Request* and a data packet in Tuft. A data packet that travels from the source node to the destination is always destined to the *Access node* to be later relayed to the sink. Based on the position of the *Access node* (closer to the *Root-Cell* and the *Root-Cell Leader*), a data packet that is close to its destination is preferred to avoid the *Cell nodes* near the *Access node* to prevent these nodes' premature depletion of batteries. Whereas a *Query-Request* packet will travel through the tree structure, and hence, the perpendicular distance criterion will not be efficient in this case.

Consequently, for a data packet three conditions are appropriated to Tuft in terms of the distance between the sender node and the destination node to the *Cell* radius. The first condition is applied whenever the distance succumbs closer to \mathcal{R} , where the residual energy criterion is given the highest importance to avoid nodes with lower residual energy (i.e. *Cell nodes*). The evaluation matrix of condition one is shown in (21). Condition two is applied only when two cases are met, the distance is between \mathcal{R} & $2\mathcal{R}$, and the perpendicular distance of the sender node with respect to the source node is more than 2δ . In this condition, the packet strayed away from the virtual line between the source node and the destination, therefore, the perpendicular distance criterion is given a higher importance, (22) illustrates the evaluation matrix for condition two. Finally, condition three is applied when none of the above conditions are met. Condition three implies that the sender node is far from the destination, therefore, the direction criterion is given the highest importance. Condition three is further illustrated in (23).

For a *Query-Request* packet, the evaluation matrix in (24) is employed, where it applies the lowest importance to the perpendicular distance distribution. Finally, other packet types (*Query-Response*, *AN*, *TUA*) in Tuft will employ the evaluation matrix (23).

$$M_1 = \begin{matrix} & d & \theta & \psi & \Phi \\ \begin{matrix} d \\ \theta \\ \psi \\ \Phi \end{matrix} & \begin{bmatrix} 1 & 1/7 & 1/5 & 1/7 \\ 7 & 1 & 7 & 1/5 \\ 5 & 1/7 & 1 & 1/7 \\ 7 & 5 & 7 & 1 \end{bmatrix} \end{matrix} \quad (21)$$

$$M_2 = \begin{matrix} & d & \theta & \psi & \Phi \\ \begin{matrix} d \\ \theta \\ \psi \\ \Phi \end{matrix} & \begin{bmatrix} 1 & 1/7 & 1/7 & 3 \\ 7 & 1 & 3 & 7 \\ 7 & 1/3 & 1 & 7 \\ 1/3 & 1/7 & 1/7 & 1 \end{bmatrix} \end{matrix} \quad (22)$$

$$M_3 = \begin{matrix} & d & \theta & \psi & \Phi \\ \begin{matrix} d \\ \theta \\ \psi \\ \Phi \end{matrix} & \begin{bmatrix} 1 & 1/7 & 1/7 & 1/7 \\ 7 & 1 & 3 & 5 \\ 7 & 1/3 & 1 & 5 \\ 7 & 1/5 & 1/5 & 1 \end{bmatrix} \end{matrix} \quad (23)$$

$$M_4 = \begin{matrix} & d & \theta & \psi & \Phi \\ \begin{matrix} d \\ \theta \\ \psi \\ \Phi \end{matrix} & \begin{bmatrix} 1 & 1/5 & 7 & 1/3 \\ 5 & 1 & 7 & 7 \\ 1/7 & 1/7 & 1 & 1/7 \\ 3 & 1/7 & 7 & 1 \end{bmatrix} \end{matrix} \quad (24)$$

5.2.3 Final Decision Making

When a sender node has a packet to send, it defines its choices from all its neighboring nodes \mathbb{N}_i . Then by a certain condition, the corresponding weight of each criterion is obtained to derive the weighted aggregation of each choice as in (19). Afterwards, the sender defines the alternative set (candidate set) \mathbb{T}_i according to (25) where m_T is the number of nodes in the set \mathbb{T}_i . The sender will then send a preamble-packet to discover its active alternatives (candidates). Any candidate that receives the preamble will respond back with an ACK-packet. The sender will then choose the alternative with the highest weighted aggregation that responded back with an ACK, and designate it as the final forwarder.

$$m_T \leq \sqrt{\mathcal{D} + 1}; \quad \tilde{R}_{i,j} \geq \frac{1}{\mathcal{D}} \quad \forall n_j \in \mathbb{N}_i \quad (25)$$

6 ANALYSIS

Tuft's performance hinges on the communal hierarchical structure and the implemented data dissemination protocol *Tuft-Cells*. This section assesses the efficiency of Tuft with regard to, the number of hops, the success ratio, and the energy consumption for obtaining the fresh sink's position.

To approximate the capability of Tuft, we only consider the worse case scenarios, where a source node is not a *Cell-Leader*, and the distance between a *Cell node* and its *Cell-Leader* is at most \mathcal{R} . According to Subsection 4.4, there are three cases for a source node n_s . **Case 1**, n_s is a *Root-Cell node*. **Case 2**, n_s is a *Cell node* encapsulated by *Child-Cell* ℓ_i at depth ($depth_i$). **Case 3**, n_s is a regular node, not encapsulated by a *Cell*.

The term data packet in this section will refer to any of these packets adopted in Tuft; *AN*, *TAC*, *Query-Request*, *Query-Response*, *Maintenance*, and *Data* packets.

6.1 Expected Number of Hops

The number of hops \mathcal{H} from source to destination influences the overall energy consumption as seen in (10), a packet that travels on a longer path will increase the energy consumed to reach its destination. Therefore, decreasing the number of hops a packet travels on will greatly decrease the energy consumption. In mobile sink WSNs this includes packets that deliver and obtain the sink's position information. In Tuft the expected number of hops for a *Query-Request* and a *Query-Response* packet remains stable throughout the network's lifetime and hence can be derived.

We assume that the nodes are uniformly distributed in the sensing field area ($A = a \times b$). Subsequently, the expected number of hop counts is derived by defining the probability

of one or multiple hop counts between the source and the destination. To begin with, the probability of a one-hop connection between the source node n_s and the destination node n_j is considered only when n_j is a neighboring node of n_s (i.e. $d_{s,j} \leq \delta$). Hence, the probability density function for the shortest *Euclidean distance* $d_{s,j}$ as defined in [32] is illustrated in (26), where \mathcal{D} is the network density. Then using (26) the probability of a one hop connection is derived as in (27).

$$f(d_{s,j}) = \frac{4d_{s,j}}{a^2b^2} \left(\frac{\pi}{2} a \times b - a \times d_{s,j} - b \times d_{s,j} + \frac{1}{2} d_{s,j}^2 \right) \quad (26)$$

$$P(\mathcal{H} = 1) = \int_0^\delta f(d_{s,j}) \, dd_{s,j} \quad (27)$$

Equivalently, the probability of a two hop connection occurs when the destination node n_j is not a direct neighbor of the source node n_s and $d_{s,j} \leq 2\delta$. Hence, there must be at least one node that intervenes in the intersection of both transmission ranges of the two nodes. Prior to attaining the probability of a two hop connection, determining the probability of having at least one relay node n_j in the area of the intersection A_{in} is needed. The probability of having $m_{in} \geq 1$ nodes in a certain area A_{in} is obtained using (29), where $P(m_{in} = 0)$ obtained in (28) is the probability of having no nodes in the area A_{in} , and m is the total number of sensor nodes in the sensing field. The probability of a two-hop connection is then defined in (30). Consequently, using (30) the probability of h -hop connection is obtained in (31).

$$P(m_{in} = 0, A_{in}) = \left(1 - \frac{A_{in}}{A} \right)^{m-2} \quad (28)$$

$$P(m_{in} \geq 1, A_{in}) = 1 - P(m_{in} = 0, A_{in}) \quad (29)$$

$$P(\mathcal{H} = 2) = P(\delta < d_{s,j} \leq 2\delta) \times P(m_{in} \geq 1, A_{in})$$

$$P(\mathcal{H} = 2) = \int_\delta^{2\delta} f(d_{s,j}) \times \left(1 - \left(1 - \frac{A_{in}}{A} \right)^{m-2} \right) \, dd_{s,j} \quad (30)$$

$$P(\mathcal{H} = h) = \int_{(h-1)\delta}^{h\delta} f(d_{s,j}) \times \left(1 - \left(1 - \frac{A_{in}}{A} \right)^{m-2} \right) \, dd_{s,j} \quad (31)$$

Finally using (27), (30), and (31), the expected number of hops $E_{\mathcal{H}}$ is concluded in (32), where Mx is the maximum number of hops obtained in (2).

$$E[\mathcal{H}] = \sum_{h=1}^{Mx} h \times P(h) \quad (32)$$

To obtain the expected number of hops for a *Query-Request* path, in the worst case, we assume that the source node n_s is not a *Cell-Leader*, and any *Cell node* is at most one *Cell radius* away from its *Cell-Leader*. Furthermore, the distance between the *Cell-Leader* in *Cell* ℓ_i and the *Root-Cell* is positively correlated with the depth of ℓ_i in the virtual tree structure. Finally, any node that isn't encapsulated by a *Cell* is at most half a *Cell radius* away from its nearest *Cell*.

Consequently, we obtain the Euclidean distance $d_{s,r}$ between the source node n_s and the *Root-Cell Leader* n_r , and then derive the expected number of hops, for each of the above-mentioned cases (*case_i*). Case 1 is obtained by (33). Case 2 is obtained in (34). Case 3 is expressed in (35). Using (2), (33), (34), and (35). The expected number of hops for each case is illustrated in (36).

$$d_1 = \mathcal{R} \quad (33)$$

$$d_2 = d_1 + \left(\left(\mathcal{R} + \frac{\mathcal{R}}{2} \right) \times depth_i \right) + d_1 \quad (34)$$

$$d_3 = \frac{\mathcal{R}}{2} + d_2 \quad (35)$$

$$\mathcal{H}_{case_i} = \sum_{h=1}^{Mx_i} h \times P(h) \quad \forall i \in \{1, 2, 3\} \quad (36)$$

6.2 Success Ratio

In asynchronous duty-cycled WSNs, each node periodically switches from a sleep state to an active state to check for potential traffic in the channel. If the node finds the channel idle, it will switch back to the sleep state. Otherwise it will send back an ACK notifying its availability to receive the data packet. The periodical switch from sleep to active state is randomly assigned to each node, and hence, may vary for different nodes.

Whenever a sender node has a packet to send, it will switch to the active state, and define its alternatives (candidates) according to subsection 5.2.3, to discover the active candidates, the sender sends a preamble-packet. Active candidates that successfully receive the preamble-packet will respond back with an ACK-packet, thereupon the sender will choose the final forwarder from the active candidates (i.e. candidates that responded back with an ACK-packet). However, if none of the candidates are active, the sender is induced to remain in a waiting-time state. This instance is defined as a transmission attempt, each transmission attempt is independent and has two results; (1) the sender exits its current state if it has at least one active candidate, and (2) the sender remains in its waiting-time state if it has no active candidates.

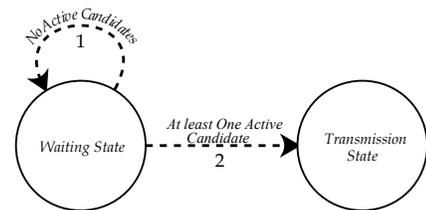


Fig. 5: The transmission attempt. A sender either: (1) remains in a waiting-time state, or (2) exits to the transmission state.

Based on our previous work [16], we assume that all nodes have the same active periods of length t , and each active period is defined with a random starting and ending point. The periodic switch from a sleep to an active state is randomly assigned within $[0, T]$. Then we attain the

probability of having ξ active candidates (alternatives) in (37), where v is the expected number of alternatives E_{alt} , and each action is separately defined in (38), (39), and (40), respectively. Therefore, we model the transmission attempt \mathcal{T}_a as a Bernoulli trial. Each trial can assume two values; success (the sender exits the current state) and failure (the sender remains in the current state), as illustrated in Fig. (5). Furthermore, a trial is characterized by the probability mass function as shown in (41), where ξ is the number of active candidates, and E_{alt} is the expected number of alternatives (candidates), which equals m_T defined in Subsection 5.2.3.

$$P_{4r}(\xi, v) = \begin{cases} P_{4r}(\xi = 0) & \xi = 0 \\ P_{4r}(\xi = 1) & \xi = 1 \\ P_{4r}(\xi \geq 2) & 2 \leq \xi \leq v \end{cases} \quad (37)$$

$$P_{4r}(\xi = 0) = \frac{1}{T^{v-1}} \left[(T - 2t^v) + \frac{2}{v} \left((T - t)^v - (T - 2t)^v \right) \right] \quad (38)$$

$$P_{4r}(\xi = 1) = (v - 1) \left(\int_0^t \frac{x+t}{T} \left(\frac{T - (x+t)}{T} \right)^{v-2} dx + \int_t^T \frac{2t}{T} \left(\frac{T - 2t}{T} \right)^{v-1} dx + \int_{T-t}^T \frac{T - x + t}{T} \left(\frac{x-t}{T} \right)^{v-1} dx \right) \quad (39)$$

$$P_{4r}(\xi \geq 2) = 1 - \left(P_{4r}(\xi = 0) + P_{4r}(\xi = 1) \right) \quad (40)$$

$$P(\mathcal{T}_a) = \begin{cases} q = P_{4r}(\xi = 0) & \xi = 0 \\ p = P_{4r}(\xi \geq 1) & E_{alt} \geq \xi \geq 1 \end{cases} \quad (41)$$

A sender n_i that remains in a waiting-time state for a consecutive $\mathcal{M}_{\hat{k}}$ number of times is forced to drop the packet. Hence, the probability of dropping a packet is designated as a binomial random variable D_h defined by a sequence of independent transmission attempts (independent Bernoulli trials) with probability q to remain in the same state, and probability p to exit the current state. Thus, the probability mass function of the random variable D_h is illustrated in (42).

$$P(D_{h_i}) = \sum_{k=0}^{\mathcal{M}_{\hat{k}}} \binom{\mathcal{M}_{\hat{k}}}{k} q^k p^{\mathcal{M}_{\hat{k}} - k} = q^{\mathcal{M}_{\hat{k}}} \quad (42)$$

The probability mass function (42) is defined for one hop (i.e. one sender). To obtain the probability of dropping a packet for a path from source to destination, we use (43) where \mathcal{H} is the number of hops from source to destination. Then, the probability of dropping a *Query-Request* packet for each of the above-mentioned three cases is illustrated in (44) where \mathcal{H}_{case_i} is obtained by (36).

$$P(D_{\mathcal{H}}) = \sum_{h_i=1}^{\mathcal{H}} P(D_{h_i}) = \mathcal{H} \times \mathcal{M}_k \times q \quad (43)$$

$$P(D_{\mathcal{H}_{case_i}}) = \mathcal{H}_{case_i} \times \mathcal{M}_k \times q \quad \forall i \in \{1, 2, 3\} \quad (44)$$

6.3 Expected Energy Cost

Based on our previous work [33] and Subsection 3.3. The energy cost of sending a packet from node n_i to node n_j depends on the decision making in the underlying routing protocol. As mentioned in Subsection 5.2.3, *Tuft-Cells* undergoes three steps for the final decision making. Mainly, the sender node will discover its active alternatives (candidates) by sending a preamble-packet, candidates that are active will receive the packet and respond back with an ACK-packet to the sender, the sender will then determine the final forwarder and send the data packet to it.

Subsequently, the transmission cost $T_C(i, j, k, \xi)$ when there are ξ active candidates is formalized in (45) and comprises the three steps. The energy cost of sending and receiving the preamble-packet $T_{Pre}(i, j, k_p, \xi)$ is illustrated in (46), where k_p is the size of the preamble-packet. The energy cost for sending and receiving the Ack-packet $T_{Ack}(i, j, k_a, \xi)$ is illustrated in (47), where k_a is the size of the Ack-packet. Finally, the energy cost for sending and receiving the data packet $T_{Data}(i, j, k)$ is illustrated in (48) where k is the size of the data packet.

$$T_C(i, j, k, \xi) = T_{Pre}(i, j, k_p, \xi) + T_{Ack}(i, j, k_a, \xi) + T_{Data}(i, j, k) \quad (45)$$

$$T_{Pre}(i, j, k_p, \xi) = T_x(i, j, k_p) + \sum_{j=0}^{\xi} R_x(j, k_p) \quad (46)$$

$$T_{Ack}(i, j, k_a, \xi) = \sum_{j=0}^{\xi} T_x(j, i, k_a) + R_x(i, k_a) \quad (47)$$

$$T_{Data}(i, j, k) = T_x(i, j, k) + R_x(j, k) \quad (48)$$

According to (45) and the probability for a node to have E_{alt} active candidates. The expected one hop energy cost $T_{hop}(i, j)$ to send a packet from node n_i to n_j is formalized in (49).

The expected energy cost for a data packet, going through the path $\mathbb{P} = \{n_1, n_2, \dots, n_{\mathcal{H}}\}$ is then obtained in (50). Finally, we obtain the energy cost for a *Query-Request* packet based on the above-mentioned cases and (36) in (51).

$$T_{hop}(i, j) = \sum_{\xi=1}^{E_{alt}} P_{4r}(\xi, E_{alt}) \times T_C(i, j, k, \xi) \quad (49)$$

$$T_{path} = \sum_y^{\mathcal{H}} T_{hop}(y, y+1) \quad (50)$$

$$T_{path} = \sum_y^{\mathcal{H}_{case_i}} T_{hop}(y, y+1) \quad \forall i \in \{1, 2, 3\} \quad (51)$$

7 PERFORMANCE EVALUATION

Tuft is evaluated using an on-purpose simulator written in visual studio 2017 (c# WPF). This simulator has been built with a graphical user interface to provide significant pieces of information visually & with ease, and has been used in our previous work [16][17][33]. The simulator was updated

to support a mobile sink. The source code and documents of the simulator are available online at: <https://github.com/howbani/tuft>.

7.1 Simulation Setup

The mobility of the sink is modeled after the Gauss-Markov Mobility Model. Nodes are randomly deployed within the sensing field. Each node runs BOX-MAC [34] with equivalent sleep and active periods. To conform with the requirements of asynchronous BOX-MAC, each node appoints a random starting point from 1s to 5s to initiate its duty cycle. The default parameters used in the simulation are listed in Table 3. Tuft is compared with Ring Routing, LBDD and GCRP. Each simulation scenario is run 10 times. The results are then averaged and plotted.

TABLE 3: DEFAULT PARAMETERS

Parameter	Value
Sensing Field Area	600m×600m
Network topology	Random
Sink	1 Mobile sink
Total number of received data packets	500 packets
MAC	BoX-MAC
Communication range radius	50m
Node active time	1s
Node sleep time	2s
Buffer size	20 packets
Radio propagation model	Free Space Model
Energy consumption model	First Order Radio Model
Data packet size	1024 bits
Control packet size	512 bits
Re-transmission attempts	7 Times
Tuft's Cell radius	2 δ
Ring Routing: Initial Ring radius	120m
LBDD: Line width	150m
Initial Energy	0.5 joule

7.1.1 Comparison Approach

To solve the advertisement issue of the mobile sink, two types of mechanisms are considered. First, the Request-Response mechanism, where the virtual structure's low-tier nodes query the high-tier nodes for the position information of the sink. Second, direct data dissemination mechanism, where low-tier nodes relay the data packet to the high-tier nodes that relay it to the sink. Hence, the results of Tuft are compared with hierarchical area based protocols Ring Routing [21] (employs the Request-Response mechanism), LBDD [22] (Line Based Data Dissemination), and GCRP [35] (Grid-cycle routing protocol) that both employ the direct data dissemination mechanism.

LBDD delivers the sink position information by broadcasts along the rendezvous area (line) which is susceptible to an increase of the overall energy consumption, especially for high sink speed cases. Besides that, data packets are sent directly to the line, constraining the nodes in the line area to process more data, and hence, depleting their batteries faster than other nodes in the network.

GCRP divides the sensing field into an even number of grid cells. Each cell is elected a grid cell head based on the distance from the node to the cell center and the residual energy of the node. Accordingly, grid cell heads connect to each other to form a cycle, that prevents the extra overhead from advertising the sink's position. Whenever the

mobile sink is nearest to a cell it informs the grid cell head that consequently, informs its other cycle headers. Thus, whenever a sender node has a packet to send, it relays it to its grid cell head that relays it to the sink's nearest grid cell head. While GCRP elects grid cell heads in an efficient way. However, the data packet that is relayed in a cycle is expected to take the same path, since, GCRP didn't employ any routing conditions that take into consideration the residual energy, and hence, is expected to deplete these nodes faster, specially in high sink speed cases. Moreover, the energy consumption from the mechanism of delivering the data packet is bound to be higher than other protocols, and have longer delivery delay.

Ring Routing delivers the sink's position information to the Ring nodes, where it is then shared around the Ring. Subsequently, making the Ring nodes prone to the hotspot problem. Ring Routing solves this by the Ring change method, however, this method requires local flooding, where the old Ring node must inform its Clockwise and Anticlockwise Ring neighbors of the change, and all these nodes are compelled to notify their neighboring nodes of the change. Otherwise, any packet destined to the Ring is unable to enter. This method notably increases the energy consumption from the overhead cost of changing the already-depleted Ring nodes, especially when the new Ring node makes a multi-hop connection with either, the Clockwise and Anti-clockwise Ring neighbors, or both. Whenever the Ring is expanding, the overhead of delivering the fresh sink's position is increasing as well. Rendering this overhead inconsistent throughout the lifetime of the network. Finally, Ring Routing doesn't consider the residual energy of nodes whenever an *Anchor node* or a new Ring node is elected.

Both LBDD and Ring Routing implement a greedy geographic routing that only considers the Euclidean distance to the destination. This, in turn, curbs the degree of energy consumption uniformity throughout the network, which diminishes the network lifetime. Moreover, greedy geographic routing doesn't limit the number of candidates for a sender node, which increases the redundant packets used for neighbor discovery as seen in (45) and (49).

Tuft incorporates the mobility of the sink to its advantage in order to: mitigate the hotspot on the *Cell-Leader* nodes, minimize the overhead of delivering the fresh sink's position, and ensure the overhead remains stable throughout the lifetime of the network. The *Cell-Leader Election* method in Tuft restrains the consumed energy to achieve a dynamic role within the *Cell* efficiently. The change of the *Access node* incorporates the residual energy as one of the metrics, which reduces the burden on the already depleted nodes. *Tuft-Cells* is able to adhere to the network's requirements for each of the previously mentioned case in Section 6, where energy is uniformly consumed throughout the network, and hence, prolongs the network lifetime. Furthermore, the number of candidates in *Tuft-Cells* is restricted to a threshold mainly related to the network density, which in turn decreases the redundant packets.

7.2 Evaluation Metrics

- *Energy consumption* or EC is defined as the total energy consumed to deliver a generated packet from

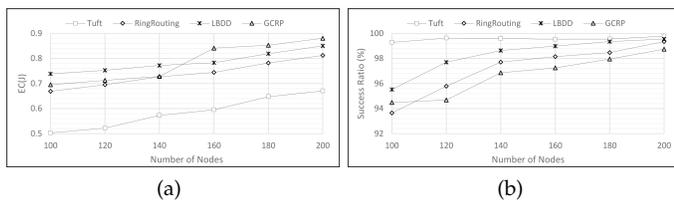


Fig. 6: Experimental results for varying the number of nodes: (a) the average energy consumption, and (b) the average success ratio.

source to destination.

- *Packet success ratio* is defined as the ratio of the number of generated packets to that of the delivered packets.
- *Average delay* is broken down to, 1) *Query-Delay*: defined as the time required for a given source node to obtain the position information of the mobile sink (time required to send a Request packet to the intended destination, along with the time required to receive the Response packet). 2) *Data Dissemination-Delay*: defined as the time required for a given source node to deliver its data packet to the mobile sink.
- *Network lifetime*. The network lifetime in this work is defined as the number of delivered data packets until the first node depletes its battery.

7.3 Experimental Results

We derive the experimental results by alternating two evaluation scenarios, the number of nodes in the sensing field, and the mean sink speed. Each scenario is further explained in its respective subsection. The final obtained results are the average of 10 distinct experiment simulations for each scenario under the same configurations.

7.3.1 Varying the Number of Nodes

The mean sink speed is set to a default 5 m/s while the number of nodes is then varied from 100 to 200 nodes. A source node is selected randomly each second to send one data packet. The simulation ends when the mobile sink successfully receives 500 data packets (that is other packets, such as *Query-Request*, *Query-Response* packets aren't counted). The results are concluded and reported as follows:

Energy Consumption. The results for the average energy consumption varying the number of nodes are plotted in Fig. 6a. The average energy consumption grows with increasing the number of nodes. As the number of nodes grows, the density also grows, which increments the number of connections each node makes. Confining the mechanism of advertising the fresh sink's position alleviates the additional overhead cost, and thus, alleviates the energy consumed. Employing the Request-Response mechanism will alleviate the extra overhead of relaying the data packet on a longer path, (data packets have bigger sizes than control packets and thus consume more energy).

GCRP starts with an average energy consumption compared to both LBDD and Ring Routing, however, as the number of nodes grows GCRP exhibits the worst energy

consumption. This is mainly due to the cell size growing which in turn, increases the number of hops each data packet travels on to reach the sink (data packets are relayed from one cell to the other). Ring Routing and LBDD both employ greedy geographic routing which does not control the number of candidates as explained in Subsection 7.1.1. Hence Tuft outperforms both protocols, due to reducing the number of redundant packets by reducing the number of candidates, and hence, reduces the energy consumed for transmitting a packet. Both LBDD and GCRP have the worst performance since the former utilizes broadcasts to share the sink's position information and the latter doesn't control the path that a data packet takes to reach its destination.

Success Ratio. The success ratio results varying the number of nodes are shown in Fig. 6b. A packet that traverses on a longer path from source to destination has a higher probability of being dropped for two main reasons. First, as seen in (43), the greater the number of hops, the higher the probability of the packet to be dropped for exceeding the number of re-transmission attempts. Second, longer paths might induce the packet to exceed its time to live (TTL) or hop limit, which in turn causes the packet to be dropped. Therefore, Tuft outperforms the three protocols in all cases, especially for small network sizes, since both LBDD and Ring Routing suffer from a lower success ratio, due to the employed greedy geographic routing, where a straight path to the destination is difficult to obtain, while GCRP suffers from a low success ratio due to the path the data packet traverses on to reach its destination, where it is induced to travel on a longer path traversing through the cycle headers. However, Tuft is able to get around the low success ratio through employing *Tuft-Cells*.

Average Delay. The average delay varying the number of nodes is broken down in Fig. 8a. The Request-Response mechanism employed in both Tuft and Ring Routing impacts the average delay but mitigates the hotspot on higher nodes. Hence, both LBDD and GCRP have the lowest overall delay. Moreover, protocols employing greedy routing (Ring Routing, LBDD, and GCRP) enter the waiting-time state when all the neighboring nodes are in a sleep state. As opposed to Tuft, which may enter a waiting-time if none of the active candidates satisfy the condition illustrated in (25), since an active neighbor isn't necessarily the best choice. Therefore, in Tuft packets will travel on near-optimal paths, however, may be induced to enter the waiting-time state more. The compensation of this trade-off lowers the energy consumption and increases the success ratio for Tuft.

7.3.2 Varying the Mean Sink Speed

The number of nodes is set to a default 150 nodes, the mean sink speed is then varied from 0 to 10 m/s . A source node is selected randomly each second to send one data packet. The results are concluded and reported as follows:

Energy Consumption. The results for the average energy consumption varying the mean sink speed are plotted in Fig. 7a. With high sink speeds, the sink's position information advertising increases. High sink speeds may introduce significant consumed energy. Therefore, incorporating the mobility of the sink in Tuft contributes to its lower energy consumption with higher sink speeds. Where the delivery of the sink's position information consumes the least energy

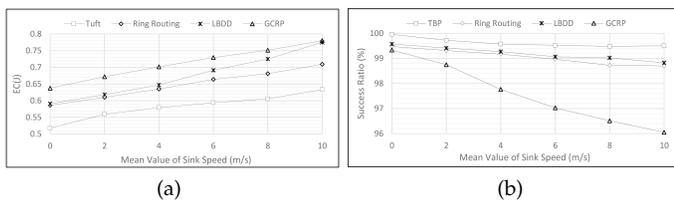


Fig. 7: Experimental results for varying the mean sink speed: (a) the average energy consumption, and (b) the average success ratio.

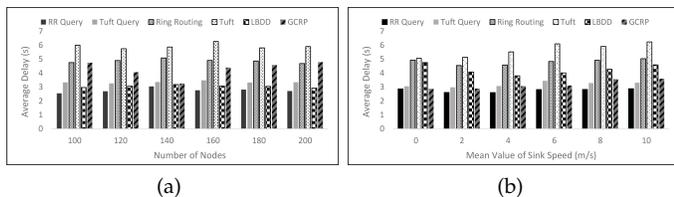


Fig. 8: Experimental results for the average delay for: (a) varying the number of nodes, (b) varying the mean sink speed.

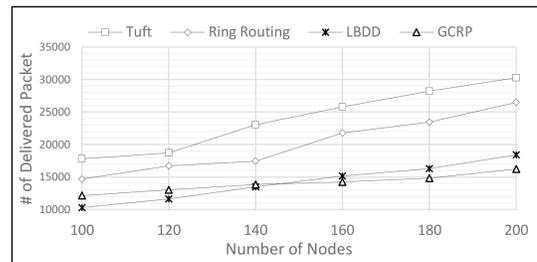
out of the three protocols. LBDD is hindered by broadcasting the position information which positively increases the energy consumption with the sink speed, thus has the highest energy consumption. Ring Routing's delivery mechanism takes a longer path (from the *Anchor node* to a Ring node, and then sharing it throughout the ring), which consumes more energy with high sink speeds. GCRP suffers the highest energy consumption specially in high sink speed due to the mechanism of delivering the data packet.

Success Ratio. The success ratio results varying the mean sink speed are shown in Fig. 7b. Higher sink speeds, increase the burden on the network, and hence, decreases the success ratio. It is clear that higher sink speeds lower the success ratio for all the protocols. However, Tuft still outperforms the three protocols for the same reasons mentioned in the first scenario.

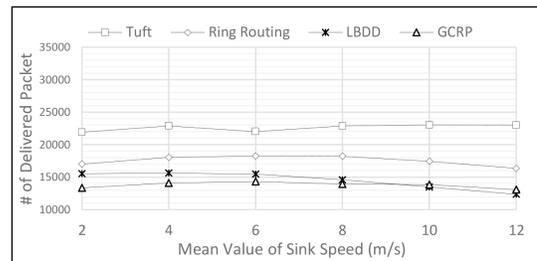
Average Delay. The average delay for varying the mean sink speed is shown in Fig. 8b. For the aforementioned reasons in Subsection 7.1.1, LBDD has the lowest overall delay. Tuft is outperformed by the other three protocols, due to the implemented routing protocol, as previously mentioned. However, the high delivery delay is compensated by lowering the energy consumption, increasing the success ratio and prolonging the network lifetime in all scenarios.

7.3.3 Network Lifetime

We define the network lifetime as the number of received data packets until the first node dies. We further include the two scenarios. For the first scenario, we vary the number of nodes and set the default sink speed to 10 m/s . For the second scenario we vary the mean sink speed from 2 to 12 m/s , while setting the number of nodes to a default 140 nodes. In the second scenario, we do not consider a static sink (at sink speed 0 m/s), since both Tuft and Ring Routing are unable to change the sink's *Access node* without exiting



(a)



(b)

Fig. 9: Experimental results for the network lifetime for: (a) varying the number of nodes, and (b) varying the mean sink speed.

its communication range, and hence, isn't an accurate result for the network lifetime of a mobile sink protocol.

Varying the number of nodes. The average network lifetime results for varying the number of nodes are plotted in Fig. 9a. The results show that Tuft outperforms LBDD, Ring Routing, and GCRP in all cases. This is mainly due to the uniform dispersion of energy consumption and the success ratio of Tuft explained in Subsection 7.1.1. GCRP and LBDD both exhibit the lowest network lifetime, for the lack of energy dispersion around the network, since higher nodes are bound to have concentrated traffic.

Varying the mean sink speed. The average network lifetime results for varying the mean sink speed are shown in Fig. 9b. The results show that Tuft outperforms the other three protocols in all cases. In the case where the sink speed is between 2 m/s and 6 m/s , the network lifetime is prolonged, since the mobility introduces a higher degree of uniform energy dispersion. Whereas, when the sink speed is between 6 m/s and 12 m/s , Tuft's network lifetime remains stable. While the network lifetime of LBDD, Ring Routing and GCRP being decreasing. The main reason behind this is because the three protocols weren't able to take the mobility of the sink to their advantage as explained in Subsection 7.1.1, therefore, the network is burdened by high sink speeds.

7.3.4 Experimental results and analysis results comparison

Due to the randomness in mobile sink WSNs there is a slight difference between the analysis results and experimental results. Accordingly, utilizing the outcome of 20 individual simulation runs. It shows that the average of the analysis results is bigger than experimental results by no more than 7%. However, as the number of simulation runs increases the gap between the average of the two results decreases.

8 CONCLUSION

This paper proposed Tuft, a novel hierarchical tree structure that adopts the mobility of the sink to its advantage. Tuft's communal tree structure contributes to minimize the energy consumption to both delivering and procuring the fresh sink's position. While implementing a dynamic role on the sensor nodes with low overhead cost. A distributed routing protocol Tuft-Cells is further proposed to adhere to the constraints of the resource scarce WSN, by taking into consideration various conditions of the network. Tuft-Cells, along with the proposed tree structure, aim at increasing the uniformity of both data traffic concentration and energy consumption throughout the network. The performance evaluations show that Tuft prolongs the network lifetime and reduces the energy consumption compared to the state-of-the-art solutions.

ACKNOWLEDGMENTS

This paper is supported by the "Fundamental Research Fund for the Central Universities NO. WK2150110007 and WK2150110012", and by the National Natural Science Foundation of China (NO. 61772490, 61472382, 61472381, 61701322 and 61572454).

REFERENCES

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [2] V. Potdar, A. Sharif, and E. Chang, "Wireless multimedia sensor networks: a survey," in *Proceedings of the international workshop on security in RFID and its industrial applications with IEEE 23rd international conference on advanced information networking and applications (AINA 2009)*. IEEE Computer Society, 2009, pp. 636–641.
- [3] A. Hawbani, X. Wang, S. Karmoshi, H. Kuhlani, A. Ghannami, A. Abudukelimu, and R. Ghoul, "Glt: Grouping based location tracking for object tracking sensor networks," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.
- [4] Y. Yun and Y. Xia, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications," *IEEE Transactions on mobile computing*, vol. 9, no. 9, pp. 1308–1318, 2010.
- [5] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 710–720, 2008.
- [6] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2015.
- [7] Y. Yun, Y. Xia, B. Behdani, and J. C. Smith, "Distributed algorithm for lifetime maximization in a delay-tolerant wireless sensor network with a mobile sink," *IEEE Transactions on mobile computing*, vol. 12, no. 10, pp. 1920–1930, 2012.
- [8] X.-X. Yuan and R.-H. Zhang, "An energy-efficient mobile sink routing algorithm for wireless sensor networks," in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2011, pp. 1–4.
- [9] M. Kang, I. Yoon, and D. K. Noh, "Efficient location service for a mobile sink in solar-powered wireless sensor networks," *Sensors*, vol. 19, no. 2, p. 272, 2019.
- [10] A. T. Erman, A. Dilo, and P. Havinga, "A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 17, 2012.
- [11] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 2, pp. 877–897, 2013.
- [12] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "Ttdd: Two-tier data dissemination in large-scale wireless sensor networks," *Wireless networks*, vol. 11, no. 1-2, pp. 161–175, 2005.
- [13] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 193–204.
- [14] K. Kweon, H. Ghim, J. Hong, and H. Yoon, "Grid-based energy-efficient routing from multiple sources to multiple mobile sinks in wireless sensor networks," in *2009 4th International Symposium on Wireless Pervasive Computing*. IEEE, 2009, pp. 1–5.
- [15] C.-J. Lin, P.-L. Chou, and C.-F. Chou, "Hcdd: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. ACM, 2006, pp. 1189–1194.
- [16] A. Hawbani, X. Wang, Y. A. AL-SHARABI, A. Ghannami, H. Kuhlani, and S. Karmoshi, "Load-balanced opportunistic routing for asynchronous duty-cycled wsn," *IEEE Transactions on Mobile Computing*, 2018.
- [17] A. Hawbani, X. Wang, A. Abudukelimu, H. Kuhlani, Y. Al-sharabi, A. Qarariyah, and A. Ghannami, "Zone probabilistic routing for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 728–741, 2018.
- [18] C. Chen, J. Ma, and J. Salomaa, "Simulation study of cluster based data dissemination for wireless sensor networks with mobile sinks," in *2008 10th International Conference on Advanced Communication Technology*, vol. 1. IEEE, 2008, pp. 231–236.
- [19] A. Hawbani, X. Wang, H. Kuhlani, S. Karmoshi, R. Ghoul, Y. Sharabi, and E. Torbosh, "Sink-oriented tree based data dissemination protocol for mobile sinks wireless sensor networks," *Wireless Networks*, pp. 1–12, 2017.
- [20] E. B. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 31–37, 2008.
- [21] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Ring routing: An energy-efficient routing protocol for wireless sensor networks with a mobile sink," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1947–1960, 2014.

- [22] E. B. Hamida and G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink," in *2008 IEEE international conference on communications*. IEEE, 2008, pp. 2201–2205.
- [23] J.-H. Shin, J. Kim, K. Park, and D. Park, "Railroad: virtual infrastructure for data dissemination in wireless sensor networks," in *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM, 2005, pp. 168–174.
- [24] B. Nazir and H. Hasbullah, "Mobile sink based routing protocol (msrp) for prolonging network lifetime in clustered wireless sensor network," in *2010 International Conference on Computer Applications and Industrial Electronics*. IEEE, 2010, pp. 624–629.
- [25] S. Arora and S. Singh, "Node localization in wireless sensor networks using butterfly optimization algorithm," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3325–3335, 2017.
- [26] L. Wei, W. Sun, H. Chen, P. Yuan, F. Yin, Q. Luo, Y. Chen, and L. Chen, "A fast neighbor discovery algorithm in wsns," *Sensors*, vol. 18, no. 10, p. 3319, 2018.
- [27] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [28] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriya-wong, "A comparative study of random waypoint and gauss-markov mobility models in the performance evaluation of manet," in *2006 International Symposium on Communications and Information Technologies*. IEEE, 2006, pp. 894–899.
- [29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE, 2000, pp. 10–pp.
- [30] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personal communications*, vol. 8, no. 4, pp. 52–59, 2001.
- [31] Y. Yin, J. Shi, Y. Li, and P. Zhang, "Cluster head selection using analytical hierarchy process for wireless sensor networks," in *2006 IEEE 17th international symposium on personal, indoor and mobile radio communications*. IEEE, 2006, pp. 1–5.
- [32] C. Bettstetter and J. Eberspacher, "Hop distances in homogeneous ad hoc networks," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, vol. 4. IEEE, 2003, pp. 2286–2290.
- [33] P. Liu, X. Wang, A. Hawbani, O. Busaileh, L. Zhao, and A. Y. Al-Dubai, "Frca: A novel flexible routing computing approach for wireless sensor networks," *IEEE Transactions on Mobile Computing*, 2019.
- [34] D. Moss and P. Levis, "Box-macs: Exploiting physical and link layer boundaries in low-power networking," *Computer Systems Laboratory Stanford University*, vol. 64, no. 66, p. 120, 2008.
- [35] A. Agrawal, V. Singh, S. Jain, and R. K. Gupta, "Gcrp: Grid-cycle routing protocol for wireless sensor network

with mobile sink," *AEU-International Journal of Electronics and Communications*, vol. 94, pp. 1–11, 2018.



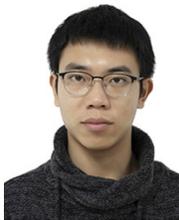
Omar Busaileh received his B.S. degree in Electronic and Information Engineering from Hefei University of Technology. Currently a Master student in the School of Computer Science and Technology at USTC. His research interests mainly include WSNs, IoTs and SDN.



Ammar Hawbani is an associate professor of networking and communication algorithms in the School of Computer Science and Technology at USTC. He received the B.S., M.S. and Ph.D. degrees in Computer Software and Theory from USTC, in 2009, 2012 and 2016, respectively. From 2016 to 2019, he worked as Postdoctoral Researcher in the School of Computer Science and Technology at USTC. His research interests include IoT, WSNs, WBANs, WMNs, VANETs, and SDN.



Xingfu Wang received the B.S. degree in electronic and information engineering from Beijing Normal University of China in 1988, and the M.S. degree in computer science from the University of Science and Technology of China in 1997. He is an associate professor in the School of Compute Science and Technology, University of Science and Technology of China. His current research interests include Information Security, Data Management and WSN.



Ping Liu received his B.S. degree in Computer Science and Technology from Guangdong University of Technology. Currently a Master student in the School of Computer Science and Technology at USTC. His research interests mainly include 5G, IoT, WSNs, VANETs, WMNs, and SDN.



Liang Zhao is a lecturer at Shenyang Aerospace University, China. He received his PhD degree from the School of Computing at Edinburgh Napier University in 2011. Before joining Shenyang Aerospace University, he worked as associate senior researcher in Hitachi (China) Research and Development Corporation from 2012 to 2014. His research interests include WMNs, VANETs and SDN.



Ahmed ADubai [SM] is a Professor of Networking and Communication Algorithms in the School of Computing at Edinburgh Napier University, UK. He received the PhD degree in Computing from the University of Glasgow in 2004. His research interests include Communication Algorithms, Mobile Communication, Internet of Things, and Future Internet.