Data Dissemination of Emergency Messages in Mobile Multi-Sink Wireless Sensor Networks

Ayşegül Tüysüz Erman and Paul Havinga
Pervasive Systems Research Group
University of Twente
Enschede, The Netherlands
{*tuysuza, havinga*}*@cs.utwente.nl*

***Abstract*—In wireless sensor networks (WSNs), data dissemination is generally performed from sensor nodes to a static sink. If the data under consideration is an emergency message such as a fire alarm, it must be transmitted as fast and reliably as possible towards the sink of WSN. In such mission critical applications, it may not be enough to have one static sink but there will be multiple and mobile sinks. Sinks can be associated to first responders such as firefighters, but also to unmanned aerial vehicles (UAVs). The existing approaches have a high communication cost and a high delivery latency, which makes them less suitable for emergency situations. As an alternative to existing protocols, we present Honeycomb Architecture and Hexagonal Tiling-Based Data Dissemination (HexDD) protocol for emergency message transmission in mobile multi-sink WSNs. Simulation results show that HexDD has a high data delivery ratio and a very low data delivery latency, which are major requirements for disaster management scenarios.**

I. INTRODUCTION

In wireless sensor networks (WSN), each sensor individually senses the environment, but collaboratively achieves complex information gathering and dissemination tasks. Typically wireless sensor network follows the communication pattern of convergecast, where sensors collect data about a phenomenon and relay streams of data to a common static sink node. Depending on the application requirements, we can mention about three basic data delivery models [1]: (i) *Periodic sensing:* sensors transmit the collected data continuously at periodic intervals, (ii) *event-driven:* sensor nodes report data only if an event of interest occurs, and (iii) *query-driven:* sensors only report data in response to an explicit request from the sink.

Our work presented in this paper is mainly motivated by disaster management scenarios where the deployment of the sensors is performed in a random fashion (e.g., dropping sensors from unmanned aerial vehicles (UAVs) flying above the field). In such scenarios, UAVs, personal forces (e.g. firefighters in a fire detection scenario), or vehicles (e.g. firetrucks in a fire detection scenario) carry sink nodes on-board. These mobile sinks are used to collect more reliable data about the event in the dangerous/inaccessible regions. In this scenario, which is under the consideration of the European research project AWARE (EU IST-2006-33579) [2], we mainly focus on data dissemination of emergency messages towards multiple mobile sinks. A mobile multi-sink WSN scenario requires for sinks to send queries which help to track sinks' location and for sensor nodes to send emergency messages when they sense an event of interest. The WSN application under consideration reveals two main questions: (i) how to achieve reliable data delivery to moving sinks, and (ii) how to keep data delivery latency low, while having a low communication cost. The mission critical WSN described above is required to solve these problems and to satisfy a high delivery ratio and a low latency criterion for critical data packets.

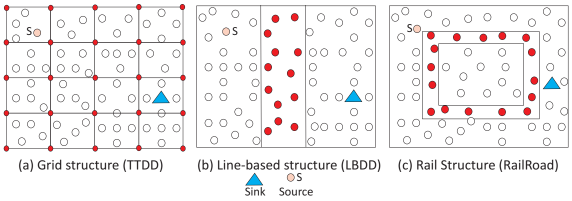
In this context, it is needed to achieve two main goals: (i) To accommodate the dynamics of the WSN due to stimulus and sink mobility, in such a way that avoids excessive updates caused by frequently changing environment, (ii) To collect as much emergency data as possible in a short time. For these purposes, in this paper, we present the *Honeycomb Architecture* and the data dissemination protocol *Hexagonal Tiling-Based Data Dissemination (HexDD)* for WSNs with mobile sinks. Our proposed scheme is based on a virtual infrastructure proposing rendezvous regions for events (data caching) and queries (lookup). It supports both mobility of the sinks and the sources. We compare HexDD to existing approaches and show that HexDD performs better than other approaches in mobile scenarios resulting in low data delivery latency and high data delivery ratio which are the main targets of our application domain.

The rest of this paper is organized as follows: Several related works are introduced with their strengths and weaknesses in Section 2. In Section 3, we present the honeycomb architecture and basic operations of HexDD. Section 4 gives the simulation results to evaluate the performance of the proposed protocol. Section 5 discusses our data dissemination protocol and its possible extensions. Finally, Section 6 concludes the paper.

II. RELATED WORK

Several data dissemination protocols have been proposed for mobile wireless sensor networks. Basically, proposed protocols fall in two major categories: (i) Flooding-based and (ii) Virtual infrastructure-based data dissemination protocols. In general, virtual infrastructure-based protocols can be divided into (i) rendezvous-based approaches, and (ii) backbone-based approaches (e.g. [3]) depending on how the virtual infrastructure is formed by set of potential storing nodes. Since we also propose a rendezvous-based protocol, we mainly discuss rendezvous-based approaches in this section.

978-1-4244-8435-5/10/$26.00



(a) Grid structure (TTDD) (b) Line-based structure (LBDD) (c) Rail Structure (RailRoad)

Fig. 1. Virtual infrastructure-based data dissemination protocols.

PEG (Pursuit-Evasion Games) [4] is a sensor network system that detects an uncooperative mobile agent, *evader*, and assists an autonomous mobile robot called the *pursuer* in capturing the evader. The routing mechanism used in PEG, namely *landmark routing*, uses the node at the center of the network as landmark (i.e. only one rendezvous point) to route packets from many sources to a few sinks. It constructs a spanning tree having the landmark node as the root of the tree. For a node in the spanning tree to route an event to a pursuer, it first sends the data up to the root, the landmark. The landmark, then, forwards the data to the pursuer. The pursuer periodically informs the network of its position by picking a node in its proximity to route a query to the landmark. Since data dissemination used in PEG is a combination of directed diffusion [5] towards the landmark and central re-dissemination, in order to build the gradients from sensors to landmark node (i.e. spanning tree), it uses flooding-based approach (i.e. each node sends a beacon packet which is further re-broadcasted by all the neighbors of the node) which results in broadcast storm problem increasing the congestion.

As the flat architectures and flooding-based protocols do not scale, overlaying a virtual infrastructure over physical network often has been investigated as an efficient strategy for an efficient data dissemination in mobile WSNs [6]. This strategy uses the concept of virtual infrastructure, which acts as a rendezvous area for storing and retrieving the collected measurements. After the mobile sink crosses the network, the designated nodes are queried to report the sensory input.

The geographic hash table (GHT) [7] is a rendezvous-based approach in which the data report type is hashed into geographic coordinates, and the corresponding data reports are stored in the sensor node, called home-node, which is the closest to these coordinates. The main drawback of this approach is the hot spot problem because all data reports and queries for the same meta-data are concentrated on the same home node. This may restrict the scalability and the network lifetime.

In TTDD [8], each source node proactively builds a uniform virtual grid structure throughout the sensor field, as shown is Fig. 1(a). A sink floods a query within its local grid cell. The query packet then propagates along the grid to reach the source node. While the query is disseminated over the grid, a reverse path is established towards sink and data is sent to the sink via this reverse path. If the stimulus is mobile, number of sources and grids increase. This situation can lead to excessive energy drain, and therefore, limit the network lifetime.

LBDD [9], which is proposed for mobility of sink and source nodes, defines a vertical line that divides the sensor field into two equal sized parts, as shown in Fig. 1(b). This line acts as a rendezvous area for data storage and look up. When a sensor detects a new event, it transmits a data report towards the nodes in the virtual line. This data is stored on the first node encountered in the virtual line. The sink's query is flooded along the virtual line until it arrives to the *inline node* that owns the requested data. Thus, data reports are sent directly to the sink. However, using a line as rendezvous area at the middle of the network can results in high latency for the nodes near the boundary of the network.

Railroad [10] places a virtual *rail* in the middle of the deployment area, as shown in Fig. 1(c). When the source node generates data, the corresponding data is still stored locally, but corresponding meta-data is also forwarded to the nearest node inside the rail. When a sink node wants to collect the generated data, a query message is sent into the rail region.

Although there are many proposals for data dissemination in WSNs with mobile sinks, most of them suffer from high communication cost resulting in high energy consumption as the number of queries and data increases. Generally, their data dissemination paths are not optimal; thus, they have also high data delivery latency.

III. HONEYCOMB ARCHITECTURE

Our main goal is to implement an efficient data dissemination protocol that supports sink mobility, by exploiting hexagonal-cell based network space partitioning. In the honeycomb architecture, the whole network space is divided into hexagonal cells of edge length r. The architecture also defines three principle diagonal virtual lines (called *border lines*) of width w which divides the sensor field into six parts, as shown in Fig. 4. The lines, which intersect at the center of the network, are used as rendezvous regions for the queries and the generated data.

In our architecture design, we also prefer to use the concept of overlaying a virtual infrastructure over the physical network because this concept has several advantages. The infrastructure acts as a rendezvous region for the queries and the generated data. Therefore, it enables the gathering of all of the generated data in the network and permits the performing of certain data optimizations (e.g. data aggregation) before sending the data to the destination sink [6]. Secondly, in WSNs deployed in harsh environments, source nodes can be affected by several environmental conditions (e.g., wildfire, seism, etc.), and therefore, the risk of losing important data is high. To ensure the persistence of the generated data, the source node can disseminate the data towards the rendezvous area instead of storing it locally. Thus, the virtual infrastructure enables data persistence against node failures. Main disadvantage of using a virtual infrastructure is creating some hotspot regions in the network. However, it is possible to solve this problem by adjusting size of rendezvous regions. The parameters r and w are used to address the hotspot problem and the scalability issue.

While having the advantages in mind, we consider that the sensing area is divided into virtual hexagonal tilings which are overlaid over the physical network. Hexagonal cells[1] were used in literature for various applications. Cellular phone station placement is one of the very well known applications of hexagonal tessellation. It is important to point out that in this paper, we don't use the concept of cellular networks. Here, we propose to use hexagonal-cell architecture only for the purpose of geographical routing towards a region (i.e. a hexagonal virtual tiling). Using hexagonal cells in order to divide the network into sub-regions is a simple and useful method for geographical routing since a node in a given hexagonal-cell knows all its neighbors with their associated-cell addresses that are the indications of which direction a neighboring node lies on. Therefore, a node can choose a neighboring node according to its data forwarding direction.

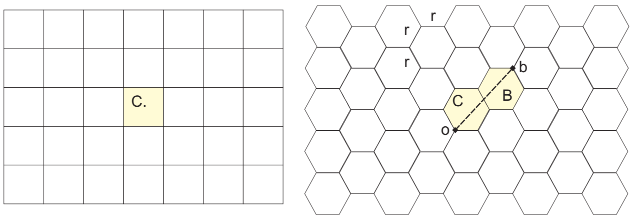
To begin with, we indicate some assumptions for virtual honeycomb architecture:

- To bring off geographic routing, every node has its location information as also assumed in [8]-[11]. This position information can be obtained either by GPS-free localization mechanisms [12], [13], [14] or by means of a virtual coordinate system [15] during the network initialization phase.
- Through periodic interactions (beacon packets), a sensor node can learn the location and cell of its neighbors.
- All of the nodes know the coordinates of the center of the network. It is required to form rendezvous region at the setup stage. A simple method to get the coordinates of the center of the network at the setup phase is given in [10].
- There are multiple sinks moving randomly in the sensor field. It is assumed that the sinks have a reliable backbone. In our actual deployment in AWARE project experiments, we use a IEEE 802.11 network as backbone in ad hoc mode [2], [16]. Sinks are equal from the information point of view; it does not matter to which sink a data packet is sent.

This section introduces basic operations of our communication protocol. They are largely divided into three parts: (i) the setup stage to construct virtual hexagonal cells and rendezvous regions, (ii) message forwarding with the support of virtual hexagonal cells, and (iii) mobility management.

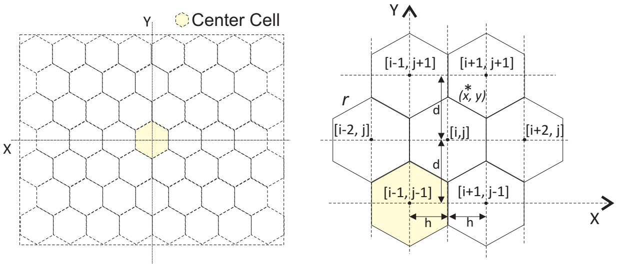
A. Hexagonal Tiling-Based Network Partitioning

In this subsection, we describe how the physical network is partitioned into virtual hexagonal cells in our architecture. Instead of square grids, which are used in many protocols [8], [17], we used a honeycomb architecture that imposes a two dimensional hexagonal grid structure, as shown in Fig. 2. While *Cell C* in rectangular mesh (Fig. 2(a)) is having four neighbors, *Cell C* in Fig. 2(b) now has six neighbors covering destination from all directions in honeycomb architecture. Therefore, one of the improvement, which honeycomb architecture provides, is grid homogeneity in sense that there exists no neighboring grid that shares a corner, rather than an edge.



(a) (b)

Fig. 2. (a) Rectangular grid, (b) Hexagonal grid.



(a) Hexagonal cell based deployment area partitioning (b) Hexagonal cells on Cartesian coordinate system and node association

Fig. 3. (a) Cell placement and (b) Node-cell association.

Definition 1: *A hexagonal cell is defined such that, for two adjacent cells C and B, all the nodes in cell C can communicate with all the nodes in cell B, and vice versa.*

As illustrated in Fig. 2(b), the longest distance between two adjacent cells, e.g. represented by line *(o, b)*, is $l_{(o,b)} = \sqrt{13}r$, where *r* is the edge length of the hexagon. According to *Definition 1*, in order for all the nodes in adjacent cells to be able to communicate with each other, the longest length must satisfy $l_{(o,b)} = \sqrt{13}r \leq R$ where R is the transmission range. Therefore, we choose $r = R/\sqrt{13}$, such that sensors in adjacent cells are within *communicable distance* of each other.

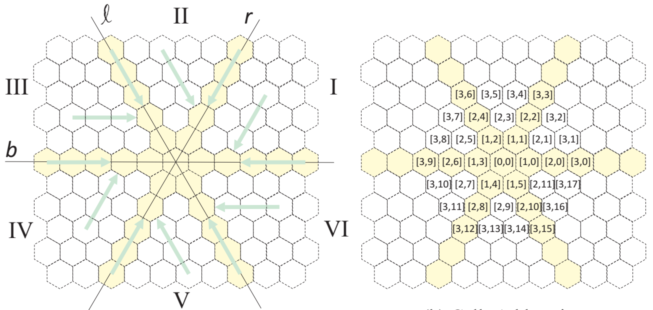
Next, we explain how honeycomb architecture overlays virtual hexagonal cells over physical network space and associates sensors with virtual cells. After cell placement, we mention how honeycomb architecture defines three border lines as rendezvous regions and gives an address to each of hexagonal cells for data dissemination.

*1) **Cell Placement and Node Association:*** In our honeycomb architecture, a hexagonal cell placement and node association scheme needs to be established. In this scheme, hexagonal virtual cells' central points are positioned according to Fig. 3(b). Apparently, $d = \frac{3}{2}r$ and $h = \frac{\sqrt{3}}{2}r$, where r is the edge size of the hexagonal cell.

As shown in Fig. 3(a), we choose the horizontal line as the X axis and vertical line as the Y axis which are crossing perpendicularly at the center of the network. Each virtual cell center is located at (id, jh) where i and j are integers. We assumed that a virtual cell centered at (id, jh) is named as the cell $[i, j]$. Fig. 3(b) show the cell $[i, j]$ and its neighboring cells with their associated names in the XY coordinate system.

---

[1] The words cell and tiling are used interchangeably.



(a) Border Cells and Hextant Formation (b) Cells Addressing

Fig. 4. Virtual Infrastructure in Honeycomb Architecture.

For the node-cell association algorithm (see Algorithm 1), we have used a similar geometrical approach used in [18]. For a node positioned at point (x, y), let $i = \lfloor x/h \rfloor$ and $j = \lfloor y/d \rfloor$. If $i + j$ is even, node positioned at (x, y) is either in cell $[i, j]$ or in cell $[i+1, j+1]$; if $i+j$ is odd, node positioned at (x, y) is either in cell $[i + 1, j]$ or in cell $[i, j + 1]$ depending on which center is closer. Therefore, with this algorithm, each sensor node uses its coordinates to associate itself with a hexagonal cell. This algorithm is lightweight in computing: It has a total of 16 lines of code. A typical node would need to go through only 9 lines of code to find its cell. There is no communication overhead. Each node executes the algorithm locally.

Algorithm 1 Node-Cell Association

```
Input: $r$: edge size of the hexagonal cell, $(x, y)$: coordinates of the node
Output: $[i, j]$ be the cell name assigned to the node at $(x, y)$
I. Calculate distance between (x,y) and candidate cells' centers
$d = 3 * r/2;\ h = sqrt(3) * r/2;$
$i = int(x/h);\ j = int(y/d);$
$a = x - (i * h);\ b = y - (j * d);$
II. Check which center is closer
if $(i + j)\%2 == 0$ then
    if $a^2 + b^2 \le (d - b)^2 + (h - a)^2$ then
        $[i, j] \Leftarrow [i, j]$
    else
        $[i, j] \Leftarrow [i + 1, j + 1]$
    end if
else
    if $b^2 + (h - a)^2 \le (d - b)^2 + a^2$ then
        $[i, j] \Leftarrow [i + 1, j]$
    else
        $[i, j] \Leftarrow [i, j + 1]$
    end if
end if
```

2) ***Hextants Formation and Cell Addressing**:* Our honeycomb architecture defines three principle diagonal lines labeled as l, b, and r which are drawn through the origin of center cell, as illustrated in Fig. 4. These lines divide the sensor field into six regions which are named as *Hextants*. Each of six *hextants* is marked with roman numerals in the figure.

After calculating the cell names $[i, j]$ for each cell, we assign addresses of the form $[H, I]$ to the each sensor nodes in the same cell named as $[i, j]$, where H is the shortest cell-count of the node from the origin cell and I denotes the index of the hop-H hexagonal cell. The index starts at the *line* b and increases in the counter-clockwise direction. Hence, the nodes in the first-hop cells are addressed as $[1, 0]$, $[1, 1]$,..., $[1, 5]$. For simplicity, we use $[H, I]$ to refer to a node's cell address. Observe that nodes of the form $[H, .]$ are all located on the same hexagonal ring at distance H form the center cell. Since the number of cells on H^{th} hop hexagonal ring is $6 \times H$, the cell addresses range from $[H, 0]$ to $[H, 6H - 1]$ (see Fig. 4(b) for an example). We transform the cell names of the form $[i, j]$ into cell addresses of the form $[H, I]$ to make use of cells on the diagonal lines (called *border lines*) as rendezvous regions for data dissemination.

Definition 2: *All hexagonal cells on diagonal lines l, b, and r are borders of hextants so called as* ***border cells****. More formally, all the cells addressed as $[H, I]$ are border cells if one of the following conditions is verified: (i)$I = 0$, (ii)$I = H$, (iii)$I = 2H$, (iv)$I = 3H$, (v)$I = 4H$, or (vi)$I = 5H$. The nodes associated with border cells are called* ***border nodes****.*

To transform $[H, I]$ to $[i, j]$, we use the equations for hextants,

$$Q = \left\lfloor \frac{I}{H} \right\rfloor, K = I - QH,$$

where $Q + 1$ is the hextant number of the given node.

Q	Transformation
0	$[H, I] \Rightarrow (2H - K, I)$
1	$[H, I] \Rightarrow (H - 2K, H)$
2	$[H, I] \Rightarrow (-H - K, H - K)$
3	$[H, I] \Rightarrow (-2H + K, -K)$
4	$[H, I] \Rightarrow (-H + 2K, -H)$
5	$[H, I] \Rightarrow (H + K, -H + K)$

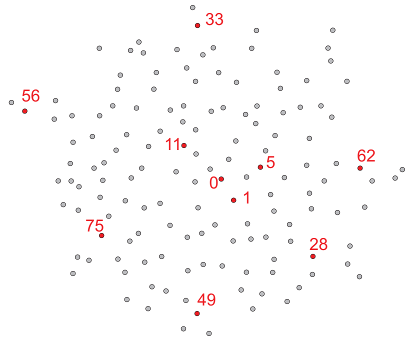
TABLE I
TRANSFORMATION RULES

For simplicity, in Table I, the transformation rules from $[H, I]$ to $[i, j]$ are shown. For the transformation from $[i, j]$ to $[H, I]$, we apply the inverse transformation of the rules in the table.

Fig. 5 shows the all steps of the virtual cell formation and addressing in a 2000x2000 m^2 network having transmission range of 250 m. The construction of this virtual infrastructure is carried out only once at the network setup stage.

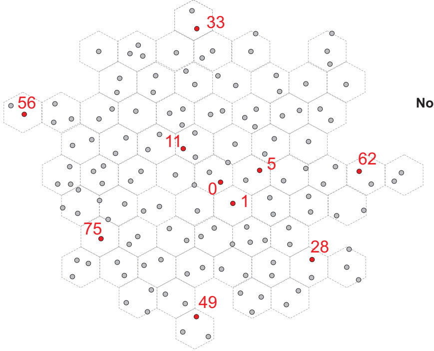
B. Hexagonal Tiling-Based Data Dissemination

In the proposed data dissemination protocol, we use the concept of central re-dissemination in which the packets flow towards the center cells following three different previously selected directions. Instead of sending packets directly to the center cell by using a simple geographic routing, we send data through border lines towards center cell because the aim is to store the generated data reports in the border



Node ID	Location
0	(1007, 1009)
1	(1027, 862)
5	(1067, 1029)
11	(904, 1131)
28	(1322, 647)
33	(925, 1513)
49	(986, 458)
56	(410, 1585)
62	(1514, 998)
75	(638, 837)

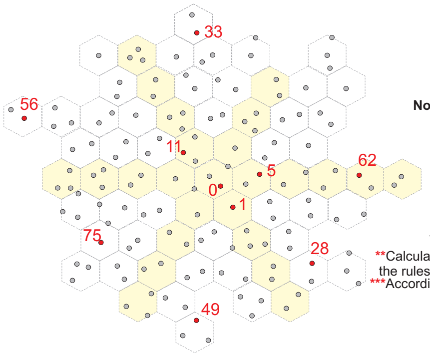
a) Network topology



Node ID	Location	Cell Name*
0	(1007, 1009)	(0, 0)
1	(1027, 862)	(1, -1)
5	(1067, 1029)	(2, 0)
11	(904, 1131)	(-1, 1)
28	(1322, 647)	(5, -3)
33	(925, 1513)	(-1, 5)
49	(986, 458)	(-1, -5)
56	(410, 1585)	(-10, 2)
62	(1514, 998)	(8, 0)
75	(638, 837)	(-6, -2)

*Calculated by Algorithm 1

b) Virtual Hexagonal Cell Formation



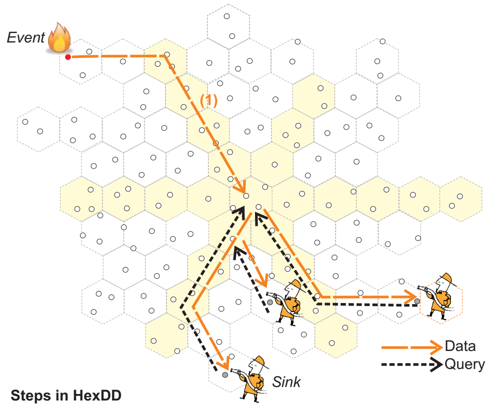
Node ID	Cell Name	Cell** Address	Border Node***
0	(0, 0)	(0, 0)	Yes
1	(1, -1)	(1, 5)	Yes
5	(2, 0)	(1, 0)	Yes
11	(-1, 1)	(1, 2)	Yes
28	(5, -3)	(4, 21)	No
33	(-1, 5)	(5, 8)	No
49	(-1, -5)	(5, 22)	No
56	(-10, 2)	(6, 16)	No
62	(8, 0)	(4, 0)	Yes
75	(-6, -2)	(4, 14)	No

**Calculated by Inverse transformation of the rules in Table 1.

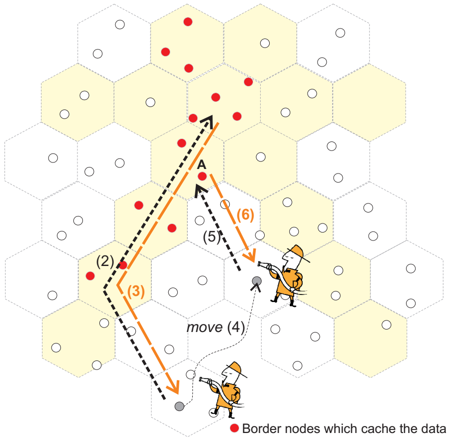
***According to Definition 2.

c) Hextant (Border Line) Formation and Cell Addressing

Fig. 5. Virtual Hexagonal Cell Formation and Addressing.



Steps in HexDD
(1) Source sends the event data towards center *(using Algorithm 2)*



(2) Sink sends a query towards center *(using Algorithm 2)*
(3) Data is forwarded to one of the diagonal line from center cell to sink
(4) Sink moves to a new cell
(5) Sink sends a new query towards center *(using Algorithm 2)*
(6) Border node **A** has the data in its cache so directly sends data to sink

Fig. 6. The HexDD protocol and its support for sink mobility

lines (i.e. rendezvous regions) such that the mobile sinks can easily collect them using a query-based data reporting method. However, our approach is purely geographical which means that we don't use flooding for route setup. The only required information is the node position which is associated with a hexagonal cell in honeycomb architecture. With the given virtual infrastructure, we propose Hexagonal Cell-based Data Dissemination (HexDD) protocol which has the following functions:

1) Data Forwarding. Data forwarding in HexDD is done through border nodes towards center region according to Algorithm 2. All the nodes in the border cells route the data packets along the straight line joining them to the center cell, as shown in Fig. 4(a) with arrows. All the other sensor nodes route the packets as follows: If the node is in hextants I or IV, packets are routed parallel to the *line* r towards the center cell. The sensors in hextants II and V forwards packets parallel to the *line* l towards the center cell. Finally, packets are sent parallel to the *line* b towards the center cell in hextants III and VI. When the data reaches one of the diagonal lines, it is forwarded along the diagonal line towards center cell. Sensors in the border lines acts as rendezvous points for data storage and lookup which means border nodes have a replica of data in their cache. The data can be either stored in all nodes of hexagonal cells or just in the cell-leader of each hexagonal cell.

Algorithm 2 Hexagonal Tiling-based Data Dissemination

```
Input: $[H, I]$: address of the current cell
Output: $[H, I]$ be the address of next hop cell
I. Find next hop cell
$k = \lceil I/H \rceil$
$[H, I] \Leftarrow [H - 1, I - k]$
```

The HexDD algorithm given in Algorithm 2 keeps the traffic flow in all regions of the network nearly balanced because honeycomb architecture divides the network space into six partitions and each partition uses a different border line segment for data dissemination; therefore, the traffic is spread among the different border line regions. After the reception of data in the center cell, the data is directed to one of the diagonal lines according to sink's location which is specified in the sink's query packet.

2) Query Forwarding. In order to retrieve a specific data, a sink sends a query towards center by using same Algorithm 2. The first border node which receives the query forwards it towards center cell. Each node in the border cells checks its cache when it receives a query. If the data requested is in the cache of a border node, it sends data back to sink through the reverse path. Replicating data on the border cells can decrease the cost of data look up and the data delivery latency. Fig. 6 shows the data and query dissemination in HexDD.

3) Mobility Support. The mobility of WSN, where most of the sensor nodes are stationary, can be divided into the stimulus mobility and sink mobility.

Stimulus mobility: The impact of stimulus mobility on the network and dissemination scheme is very little because when stimulus moves to another cell, a sensor node that captures the stimulus just sends the data towards center.

Sink mobility: If the sink moves inside its current cell, there is no need for another process since the data will be forwarded to the same neighboring cell until sink leaves its cell. When the sink moves to another cell, it needs to send a new query message towards center to inform the center nodes about its new cell. If any border node has the requested data in its cache, it directly sends data to the new cell of the sink (see Fig. 6).

IV. PERFORMANCE EVALUATION

For the purpose of performance evaluation, we have compared our protocol HexDD with other rendezvous-based approaches LBDD and grid-based TTDD model. We choose TTDD and LBDD as the base comparison since we would like to investigate the effect of using hexagonal cells instead of rectangular grids and using three diagonal lines acting as rendezvous area instead of only one line-based region. Firstly, we analyze HexDD, LBDD and TTDD with varying total number of sink-source pairs. Secondly, we explore the impact of sink mobility (i.e. sink's maximum speed) on the performance of these protocols.

A. Simulation Environment

We have implemented and tested our protocol in the ns2 [19] environment. In order to guarantee a fair comparison between TTDD and HexDD, we set simulation parameters comparable to those used in [8]. This includes simulation of IEEE 802.11 DCF as the underlying MAC and an energy model in which a sensor node's transmitting, receiving and idling power consumption rates are set to 0.66W, 0.395W and 0.035W, respectively. The cell size α in TTDD model is set to 600 meters. The LBDD virtual infrastructure parameter, g, width of the line, is set to 250 m. In our simulation setting, each node has a transmission range of 250 m and 250 sensor nodes are randomly distributed on a 2000×2000 m^2 field. Each simulation run lasts for 200 seconds, and each result is averaged over six random network topologies. A source generates one data packet per second, so there are total 200 data packets/source sent. Sinks' mobility follows the standard *Linear Mobility* model. For different set of simulations, speed and pause times of sink are varied.

We used the following metrics to evaluate the performance of HexDD and TTDD: (i) *Average Data Delivery Ratio:* defined as the ratio between the total number of data packets received by the sinks from a specific source and the total number of data generated by its corresponding source; (ii) *Average Delay:* defined as the total time elapsed between the data generation by a source and its reception by a sink, also averaged over all source-sink pairs; and (iii) *Average Energy Consumption:* defined as the communication (transmitting and receiving) energy the network consumes; the idle energy is not counted since it depends largely on the data generation interval and does not indicate the efficiency of data delivery.

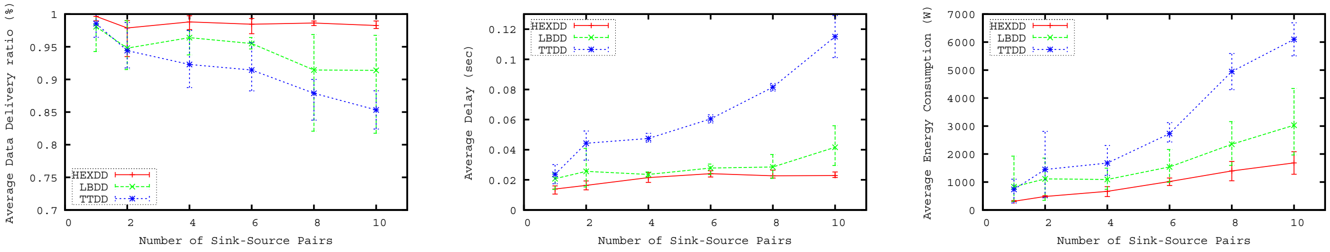
B. Simulation Results

1) ***Impact of the number of sink-source pairs****:* For the first set of simulations, number of sink-source pairs is varied. Mobile sinks could attain a maximum speed up to 10 m/s with 5 seconds pause time. The stimuli remain static during the simulation time.

Fig. 7(a) shows the average data delivery ratio. We observe that the success rate slightly decreases as the number of sink-source pairs increases because of the congestion in the network. Although the results of HexDD, LBDD and TTDD are close, HexDD has the highest delivery ratio benefiting from the use of a virtual infrastructure of three border lines which allows to better distribute the load among the nodes inside the rendezvous area. The delivery ratio of TTDD scheme falls more consistently as the number of sink-source pairs grows.

Fig. 7(b) presents the average delay in seconds. We notice that in all protocols, the delay increases with the increase in number of sink-source pairs. However, the increase in the delay of TTDD is very large since more sources generate more data packets, and more sinks need more local query flooding. Both increase the traffic volume and lead to longer delivery time. In HexDD, on the other hand, since all sources and sinks uses the same common hexagonal architecture for data storage and look up, there is no need a flooding mechanism to track sink mobility. Therefore, the incurred delay slightly increases as the number of sink-source pair increases. The delay of LBDD and HEXDD are very close to each other; however, while the number of sink-source pairs increases the difference between delay values of LBDD and HEXDD gets larger since in LBDD, more sinks mean more flooding in the virtual-line resulting in increase in the traffic volume and lead to longer delivery time.

Fig. 7(c) shows energy consumption of the whole system for all schemes. For all protocols, it is observed that the energy consumption is linear in the number of sink-source pairs. TTDD presents a rather higher communication cost since there is no global virtual infrastructure in TTDD. In TTDD, every source node sends data packets to four different corners to construct its own grid structure. As the number of source node increases, a separate grid construction and maintenance on per

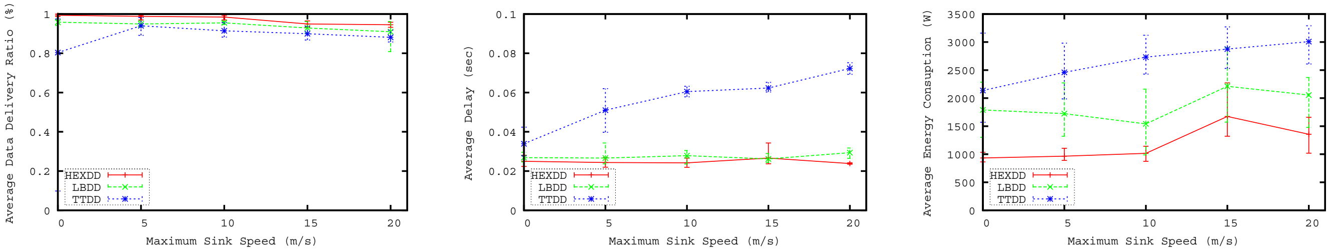


(a) Average Data Delivery Ratio (%)

(b) Average Delay (sec)

(c) Average Energy Consumption

Fig. 7. Impact of the number of sink-source pairs



(a) Average Data Delivery Ratio (%)

(b) Average Delay (sec)

(c) Average Energy Consumption

Fig. 8. Impact of sink speed

source basis results in higher cost, both in terms of packets and energy overhead. Also, the local query flooding mechanism in TTDD contributes significantly to overall energy consumption. Therefore, there is a big gap between plots of HexDD and TTDD in the figure, especially for higher number of sink-source pairs. However, the global virtual hexagonal infrastructure of HexDD results in lower energy consumptions since data packets are required to send to only rendezvous nodes (i.e. nodes in border cells). Although LBDD also uses a virtual infrastructure, it suffers from congestion and retransmissions during the flooding of queries through the line-region.

2) ***Impact of sink mobility:*** In this section, we present a comparison between HexDD, LBDD and TTDD with varying sink speeds. We have tested the performance of given protocols under both low mobility (i.e. 4-5 km/h for walking humans) and high mobility (e.g., 50-60 km/h for UAVs) scenarios. Therefore, the sinks' speeds are set to 0, 5, 10, 15, and 20 m/s (0 to 72km/h) with a pause time of 5 seconds, where the speed of 0 m/s means a static sink apparently. The speed 20 m/s means that a sink crosses the border of a cell approximately every 7 seconds (i.e. $2r = 138.5m$ which is the longest distance in a hexagonal, $138.5/20 \cong 7seconds$) in HexDD. We have 6 sink-source pairs in this scenario. The average data delivery ratios of all protocols with varying speeds of sinks show the same behavior in the previous set of simulations. Data delivery rate (see Fig. 8(a)) decreases as the moving sinks' speeds increase. However, for all protocol, success rate remains within the range of 98% to 87%. Even though there is no explicit mobility tracking scheme in HexDD, it functions well under higher mobility. Fig. 8(c) shows that the average energy consumption of TTDD is higher than HexDD and LBDD, since as the sink moves faster it tends to reconstruct a new path between the sinks and the grid by local query flooding and agent updates. In HexDD, the reason of slightly increasing energy consumption is the frequent change of sinks' cell and border node which forwards the data. Due to the same reason, LBDD also does more flooding of the query of sink for its location updates. Finally, Fig. 8(b) shows the average delay vs sinks' speed. The data delivery latencies in HexDD and LBDD are lower than delay in TTDD. While a sink can access data cached in border nodes in a short time in HexDD even while it is moving around the network, in LBDD, it has to search for data in the inline region. Therefore, LBDD has higher delay than HexDD.

V. DISCUSSION AND OPEN ISSUES

As shown in [6], in virtual infrastructure based data dissemination protocols, the use of a large virtual infrastructure - such as HexDD - reduces the dissemination cost and hot spot problem but it increases the data lookup costs. On the other hand, the use of a small virtual infrastructures - such as GHT and PEG - may reduce the energy cost of data dissemination and collection but it may also reduce the protocol reliability and robustness as it concentrates the traffic over a small group of nodes, inducing congestion and early death of nodes. These tradeoffs show that virtual infrastructures parameters - such as the border-line width and cell edge length (e.g., w or r) in HexDD - must be quided by network requirements, and in particular its traffic pattern. For example, in HexDD, depending on the frequency of data reports and queries, to avoid congestion near the center of the network, storing data to a bigger central region having multiple hexagonal cells may be preferable to a data replication over one central cell. For this purpose, HexDD protocol can modify the size of the central region according to the traffic load in the network. For higher frequencies of data reports and queries, the central region can be extended to the cells at the first and/or second hexagonal rings.

In-network data aggregation of event messages is also a useful method to avoid sending too many individual packets carrying similar data. Since the forwarding paths along the diagonals of sensor fields are shared among all source-sink pairs, it provides an opportunity for similar data to meet at some common border nodes. Data from multiple sources can be aggregated and replaced by a single data packet and forwarded towards the destined sink. Our proposed scheme can achieve further performance gain by in-network data aggregation.

In our forwarding scheme, we assume that there is at least one node which will perform multi-hop routing within each cell. However, this may not be always the case and sometimes an area of network can be lost for some reasons (e.g. environmental reasons such as fire, earthquake, etc.). A routing holes detection and bypassing mechanism can be added to our protocol. Honeycomb architecture helps to discover holes and find alternative paths because the architecture gives a node an idea about in which directions its neighbors spread. It is also possible to integrate HexDD protocol with another geographical routing which is designed to recover holes.

Due to paper length limitations, we have chosen to let the effects of these issues as future work for a more complete paper.

VI. CONCLUSIONS

In this paper, our goal was designing an data dissemination protocol which supports mobility of sink and source by keeping the data delivery ratio as high as possible and data delivery delay as low as possible. Therefore, we proposed a new virtual infrastructure called *Honeycomb Architecture* which allows an efficient data dissemination, namely *HexDD*. The HexDD uses the concept of rendezvous region for events and queries. The idea is based on storing the event messages in three principle diagonal linear regions called border lines crossing at the center of the network. The sink then retrieves the relevant data by scanning the nodes in border line which is nearest to sink's current location. The border lines, which lie on the every direction of the network, make it faster for sinks to access data. The simulation results show that our architecture helps in minimizing data delivery latency, maximizing data delivery ratio and significant energy saving. To avoid the hot region problem which may be observed in the diagonal border lines and the central cells, we discussed possible extensions such as adjusting the size of the border lines and central region according to the size of the network and the network traffic. Deploying more nodes to these regions is also another precaution for hot spot problem.

REFERENCES

[1] W. H. S. Tilak, N. B. Abu-Ghazaleh, "A taxonomy of wireless microsensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, April 2002.

[2] A. Ollero, M. Bernard, M. L. Civita, L. van Hoesel, P. Marron, J. Lepley, and E. de Andres, "Aware: Platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with unmanned aerial vehicles," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2007)*, Rome, 2007, pp. 1–6.

[3] X. Chen and M. Xu, "A geographical cellular-like architecture for wireless sensor networks," in *Proceedings of the 1st International Conference Mobile Ad-hoc and Sensor Networks (MSN 2005)*, Wuhan, China, December 2005, pp. 249–258.

[4] C. Sharp, S. Schaffert, A. Woo, N. Sastry, C. Karlof, S. Sastry, and D. Culler, "Design and implementation of a sensor network system for vehicle tracking and autonomous interception," in *Proceedings of the Second European Workshop on Wireless Sensor Networks*, January – February 2005, pp. 93–107.

[5] R. G. C. Intanagonwiwat and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the sixth annual international conference on Mobile computing and networking*, Boston, MA USA, 2000, pp. 56–67.

[6] G. C. E.B. Hamida, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications Magazine*, vol. 15, no. 6, pp. 31–37, December 2008.

[7] S. Ratnasamy, B. Karp, and D. Estrin, "Ght: A geographic hash table for data-centric storage," in *Proceedings of 1st ACM International Workshop of Wireless Sensor Net. and Apps.* ACM Press, September 2002, pp. 78–87.

[8] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*. Atlanta, Georgia, USA: ACM, 2002, pp. 148–159.

[9] E. B. Hamida and G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink," in *the IEEE International Conference on Communications (ICC 2008)*. Beijing, China: IEEE, May 2008.

[10] J.-H. Shin, J. Kim, K. Park, and D. Park, "Railroad: virtual infrastructure for data dissemination in wireless sensor networks," in *PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. Montreal, Quebec, Canada: ACM, 2005, pp. 168–174.

[11] Z. H. Mir and Y.-B. Ko, "A quadtree-based data dissemination protocol for wireless sensor networks with mobile sinks," in *Proceedings of 11th International Conference Personal Wireless Communications (PWC 2006)*, Albacete, Spain, September 2006, pp. 447–458.

[12] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[13] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "Tps: A time-based positioning scheme for outdoor wireless sensor networks," in *Proceedings of the 23nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004, pp. 2685–2696.

[14] Y. Shang and W. Ruml, "Improved mds-based localization," in *Proceedings of the 23nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004, pp. 2640–2651.

[15] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*. San Diego, CA, USA: ACM, 2003, pp. 96–108.

[16] P. Gil, I. Maza, A. Ollero, and P. Marron, "Data centric middleware for the integration of wireless sensor networks and mobile robots," in *Proceedings of the seventh conference on mobile robots and competitions (ROBOTICA 2007)*, Portugal, 2007.

[17] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. Rome, Italy: ACM, 2001, pp. 70–84.

[18] R. P. Liu, G. Rogers, and S. Zhou, "Honeycomb architecture for energy conservation in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2006)*, San Francisco, CA, USA, November 2006, pp. 1–5.

[19] "Network simulator ns-2," http://www.isi.edu/nsnam/ns.