Sleep Scheduling for Geographic Routing in Duty-Cycled Mobile Sensor Networks

Chunsheng Zhu, Student Member, IEEE, Laurence T. Yang, Member, IEEE, Lei Shu, Member, IEEE, Victor C. M. Leung, Fellow, IEEE, Joel J. P. C. Rodrigues, Senior Member, IEEE, and Lei Wang, Member, IEEE

Abstract—Recently, the research focus on geographic routing, a promising routing scheme in wireless sensor networks (WSNs), is shifting toward duty-cycled WSNs in which sensors are sleep scheduled to reduce energy consumption. However, except the connected-$k$ neighborhood (CKN) sleep scheduling algorithm and the geographic routing oriented sleep scheduling (GSS) algorithm, nearly all research work about geographic routing in duty-cycled WSNs has focused on the geographic forwarding mechanism; further, most of the existing work has ignored the fact that sensors can be mobile. In this paper, we focus on sleep scheduling for geographic routing in duty-cycled WSNs with mobile sensors and propose two geographic-distance-based connected-$k$ neighborhood (GCKN) sleep scheduling algorithms. The first one is the geographic-distance-based connected-neighborhood for first path (GCKNF) sleep scheduling algorithm. The second one is the geographic-distance-based connected-neighborhood for all paths (GCKNA) sleep scheduling algorithm. By theoretical analysis and simulations, we show that when there are mobile sensors, geographic routing can achieve much shorter average lengths for the first transmission path explored in WSNs employing GCKNF sleep scheduling and all transmission paths searched in WSNs employing GCKNA sleep scheduling compared with those in WSNs employing CKN and GSS sleep scheduling.

Index Terms—Connected-$k$ neighborhood (CKN), duty-cycle, geographic routing, mobility, wireless sensor networks (WSNs).

I. INTRODUCTION

GEOGRAPHIC routing [1]–[4] is one of the most promising routing schemes in wireless sensor networks (WSNs) [5], due to its simplicity, scalability, and efficiency [6]. In such a scheme, regardless of the network size, the forwarding decision is determined purely based on the location of each node and it can be done even when there are irregular radio ranges and localization errors. Recently, the research focus of geographic routing is centering on WSNs with duty-cycles, since duty-cycled WSNs have a natural advantage of saving energy by dynamically putting nodes to sleep and waking them according to some sleep scheduling algorithms [7]–[10].

However, nearly all these works overlook one important fact that sensors can actually be mobile to gain better energy efficiency, channel capacity, etc., and enable a lot of new application scenarios [11]–[14]. For example, because sensors can move, they can transmit their data from different locations and avoid the problem that sensors near the gateway or sink always exhaust their energy first; thus, energy usage can be more efficient [15]. Also, mobile sensors such as mobile phones or cars can become the interface between the information center and the mobile customers; thus, real-time information (e.g., traffic information) transmitted from the information center to these mobile objects can be provided to nearby customers [16], [17].

Moreover, almost all current works about geographic routing in duty-cycled WSNs [18]–[21] try to change the geographic forwarding mechanism to deal with the dynamic topology caused by some nodes being cycled off or going to sleep mode. For instance, it is suggested in [18] to wait for the appearance of the expected forwarding successor first and select a backup node if the first mechanism fails. In [19], the sensor field is sliced into some $k$-coverage fields, then some always-on cluster heads are selected to collect the data from their nearby sensors and finally transmit all data to the sink. Apart from the connected-$k$ neighborhood (CKN) sleep scheduling algorithm proposed in [22] and the geographic routing oriented sleep scheduling (GSS) algorithm presented in [23], few research works have tackled the node availability uncertainty issue in duty-cycled WSNs from the view of sleep scheduling.

This paper addresses the sleep scheduling problem in duty-cycled WSNs with mobile nodes (referred as mobile WSNs in the following) employing geographic routing. We propose two geographic-distance-based connected-$k$ neighborhood (GCKN) sleep scheduling algorithms. The first one...
is the geographic-distance-based connected-k-neighborhood for first path\(^1\) (GCKNF) sleep scheduling algorithm, aiming at geographic routing utilizing only the first transmission path in duty-cycled mobile WSNs. The second one is the geographic-distance-based connected-k-neighborhood for all paths\(^2\) (GCKNA) sleep scheduling algorithm, for geographic routing concerning all paths explored in duty-cycled mobile WSNs. By theoretical analysis and performance evaluations by simulations, we show that when there are mobile sensors, geographic routing can achieve much shorter average lengths for the first transmission paths searched in mobile WSNs employing GCKNF sleep scheduling and all transmission paths explored in mobile WSNs employing GCKNA sleep scheduling compared with those in mobile WSNs employing CKN or GSS sleep scheduling.

The main contributions of this paper are summarized as follows.

1) This paper is a pioneering work proposing and analyzing sleep scheduling algorithms for geographic routing in duty-cycled mobile WSNs, which take full advantages of both duty cycling and sensor mobility.

2) Specifically, this paper proposes two GCKN algorithms, which effectively extend existing geographic routing algorithms designed for static WSNs into duty-cycled mobile WSNs by applying sleep scheduling. The GCKNF sleep scheduling algorithm is designed to explore shorter first transmission paths for geographic routing in duty-cycled mobile WSNs. The GCKNA sleep scheduling algorithm aims at shortening all routing paths for multipath transmissions in duty-cycled mobile WSNs. These GCKN algorithms incorporate the connected-\(k\) neighborhood requirement and geographic routing requirement to change the asleep or awake state of sensor nodes.

For the rest of this paper, Section II reviews the basic idea and related work about geographic routing and sleep scheduling. The detailed designs and theoretical analysis of the GCKN algorithms are presented in Section III. Section IV evaluates the performance of the GCKN algorithms. Section V concludes the paper.

II. RELATED WORK

A. Geographic Routing

The basic idea of geographic routing is greedy routing. Specifically, each packet is tagged with the coordinates of its destination, all nodes know their own coordinates, and a node forward the packet to its neighbor that is geographically closest to the destination.

The earliest proposal for geographic routing is in [24], which has a local minimum problem in that a node may have no closer neighbor to the destination. For this reason, face routing [1] and its variants are proposed to use geometric rules (e.g., right hand rule) to route around voids near the local minimum in case it happens. However, these algorithms require converting the network into a planar graph (e.g., [25]) or removing the problematic cross links from the network (e.g., [3]), which are not very applicable in realistic conditions [26]. Moreover, there is also a hole problem in geographic routing, in that a hole can be formed by a set of dead sensor nodes running out of energy or being damaged. To solve this problem, some research work (e.g., [27]) try to identify the hole boundary nodes first and then use these boundary nodes to avoid the hole. Others (e.g., [28]) try to use geometric modeling to find an optimized hole-bypassing routing path. Recently, by using a step back and mark strategy when it cannot find the next-hop node, a two-phase geographic forwarding (TPGF), which does not have the local minimum or the hole problem, is shown in [29]. With a label-based optimization method, TPGF can optimize the routing paths by finding one with the least number of hops. However, all these works only consider WSNs with static nodes.

Recently, many opportunistic routing protocols [18], [19], [30], [31] have been proposed to extend geographic routing to duty-cycled WSNs. They all try to achieve this goal by dynamically choosing the forwarding node based on the best potential node that can transmit packets. Specially, these protocols typically take into account such factors as link uncertainty to adapt routing accordingly. However, few of these works address the local minimum or hole problem, and nearly all these works do not consider the situation that sensor nodes can be mobile.

B. Sleep Scheduling

The basic mechanism for sleep scheduling is to select a subset of nodes to be awake in a given epoch while the remaining nodes are in the sleep state that minimizes power consumption, so that the overall energy consumption can be reduced.

Existing works on sleep scheduling in WSNs mainly focus on two targets: point coverage and node coverage. For point coverage (also known as spatial coverage), the awake nodes in each epoch are chosen to cover every point of the deployed field. Existing point coverage oriented algorithms differ in their sleep scheduling goals: minimizing energy consumption [7], or minimizing average event detection latency [8]. For node coverage (also called network coverage), awake nodes are selected to construct a globally connected network such that each asleep node is an immediate neighbor of at least one awake node [32], [33]. However, all these works generally focused on the medium access layer of static WSNs with static nodes.

The only recent works addressing sleep scheduling in duty-cycled WSNs employing geographic routing are the CKN scheme proposed in [22] and the GSS method presented in [23]. CKN is a sleep scheduling method providing node coverage and a probabilistic point coverage, which tunes the number of awake nodes in the network by changing the value of \(k\) in CKN. GSS is based on CKN and differs from CKN only by making the potential nearest neighbor nodes to the sink to be awake. However, both CKN and GSS do not consider the scenarios in which sensor nodes can be mobile, and both CKN and GSS determine the awake or asleep state of each node based only on a random rank, which may keep awake many nodes far away from the destination and thus degrade the performance of geographic routing.

\(^1\)This is the first path searched by geographic routing.

\(^2\)These include all the paths found by geographic routing.
III. GCKN Algorithms

A. Network Model

We consider a multihop WSN with \( N \) sensor nodes, which can be modeled by a communication graph \( G = (U, L) \), where \( U = \{u_1, u_2, \ldots, u_N\} \) is the set of normal sensor nodes excluding the source and the sink node and \( L \) is the set of links. The default transmission radius of each sensor is \( tr \) and the maximum transmission radius of each sensor is \( tr_m \). The source and sink are always-on and both assumed to have unlimited energy supplies. The sink or a normal sensor can move to a randomly chosen position with a randomly selected speed within the WSN boundary and it will pause for a time period after it reaches the selected position, according to the random waypoint model in [34], [35]. Normal sensors can dynamically change states between asleep and awake. Two sensors are neighbors if they are within the transmission range of each other and a link \( l_{(u,v)} \in L \) if nodes \( u \) and \( v \) can communicate with each other directly without relaying. Two sensors are two-hop neighbors if \( l_{(u,v)} \not\in L \) and there exists another node \( w \) satisfying \( l_{(u,w)} \in L, l_{(w,v)} \in L, \) or \( l_{(v,w)} \in L, l_{(w,u)} \in L \).

B. Assumptions

We assume that each node knows its own location by using a Global Position System (GPS) receiver or some mobility-based localization algorithm [36]. We further assume that each node also knows the locations of the source and sink nodes by flooding or opportunistic flooding [37]. Specifically, as each sensor knows its own location, if the sink is static and normal sensor nodes are mobile, the sink location information only needs to be flooded once. If the sink is mobile and normal sensor nodes are static, the sink location information needs to be flooded when it moves to a new location.

C. Design Factors

For both GCKNF and GCKNA, we incorporate the connected-\( k \)-neighborhood\(^3\) requirement and geographic routing requirement in their designs.

Specifically, we consider the following six factors for both GCKNF and GCKNA.

1) A node should go to sleep assuming that at least \( k \) of its neighbors will remain awake so as to save energy as well as keep \( k \)-connected.

2) The asleep or awake state of nodes should be allowed to change between epochs so that all nodes can have the opportunity to sleep and avoid staying awake all the time, thus distributing the sensing, processing, and routing tasks across the network to prolong the network lifetime.

3) Although each node decides to sleep or wake up locally, the whole network should be globally connected so that data transmissions can be performed.

4) Each node should have enough initial neighbors [38], in order to make it easier for the node to satisfy the connected-\( k \)-neighborhood requirement; thus, it is more likely to be asleep after sleep scheduling. For GCKNF, which emphasizes the first transmission path of geographic routing, we further take the following factor into account.

5) The neighbor of each node, which is closest to sink, should be awake so that geographic routing can utilize these nearest neighbor nodes to make the first transmission path as short as possible. For GCKNA, which considers all transmission paths, we further take the following factor into consideration.

6) For each node, as many as possible of its neighbor nodes that are closer to the sink should be awake so that geographic routing can make all transmission paths as short as possible.

In contrast with CKN and GSS, the fourth design factor of both GCKNF and GCKNA is the extra consideration that makes it easier for each node to satisfy the connected-\( k \) neighborhood requirement during sleep scheduling. In addition, the fifth design factors for both GCKNF and GCKNA to meet the geographic routing requirement in case they encounter mobile sensor nodes or mobile sinks are ignored by the CKN and GSS schemes.

D. GCKN Algorithms

The pseudocode of GCKNF and GCKNA is shown below. Specifically, in GCKNF, each node sends probe packets to its neighbor nodes and receives the ACK packet from its neighbor nodes (Step 1 of the first part of GCKNF). With that, each node calculates whether it currently satisfies the connected-\( k \) neighborhood requirement or not (Step 2 of the first part of GCKNF). If it already belongs to a connected-\( k \) neighborhood or its transmission radius is the maximum, the node maintains its transmission radius. Otherwise, the node increases its transmission radius until the connected-\( k \) neighborhood appears (Step 3 of the first part of GCKNF) [38]. In the second part of GCKNF, the geographic locations (e.g., \( g_u \)) of each node \( u \) and the sink are obtained (Step 1 of the second part of GCKNF) and the each node’s neighbor that is nearest to sink is identified (Step 3 of the second part of GCKNF). In the third part of GCKNF, a random rank \( rank_u \)\(^4\) of each node \( u \) is picked (Step 1 of the third part of GCKNF) and the subset \( C_u \) of \( u \’s \) currently awake neighbors having rank > \( rank_u \) is computed (Step 5 of the third part of GCKNF). Before \( u \) can go to sleep, it needs to ensure that 1) all nodes in \( C_u \) are connected by nodes with rank < \( rank_u \), 2) each of its neighbors has at least \( k \) neighbors from \( C_u \), and 3) it is not the neighbor node closest to the sink for any other node (Step 6 of the third part of GCKNF).

In GCKNA, each node \( u \) also sends a probe packet to each neighbor node and receives the corresponding ACK packet (Step 1 of the first part of GCKNA). Then, whether it currently belongs to a connected-\( k \) neighborhood is also checked (Step 2 of the first part of GCKNA). The transmission radius of the node is increased if the connected-\( k \) neighborhood requirement

\(^{3}\text{Given a constant } k \text{ and an undirected graph } G = (U, L), \text{ a connected-} k \text{-neighborhood is a subset } C \subseteq U \text{ with the following two characteristics: 1) each node } u \in U \text{ has at least } m = \min(k, d_u) \text{ neighbors from } C, \text{ where } d_u \text{ is the degree of } u \text{ in } G; \text{ 2) the nodes in } C \text{ are connected.}\)

\(^{4}\text{The } rank_u \text{ here is the same as the } rank_u \text{ in CKN. The value of both } rank_u \text{ are random in each epoch and they are without real meaning.}\)
is not satisfied and the transmission radius is maintained if the nodes form a connected-
 5 The grank here is a little different from the rank in CKN, grank represents the geographic distance 
 6 This grank here is a little different from the rank in CKN, grank represents the geographic distance 

Second: Run the following at each node \( u \).
1) Get its geographic location \( g_u \) and sink location \( g_s \).
2) Broadcast \( g_u \) and receive the geographic locations of its all neighbors \( A_u \). Let \( G_u \) be the set of these geographic locations.
3) Unicast a flag to \( w, w \in A_u \) and \( g_w \) is the closest to sink in \( G_u \).

Third: Run the following at each node \( u \).
1) Pick a random rank \( \text{rank}_u \).
2) Broadcast \( \text{rank}_u \) and receive the ranks of its currently awake neighbors \( N_u \). Let \( R_u \) be the set of these ranks.
3) Broadcast \( R_u \) and receive \( R_v \) from each \( v \in N_u \).
4) If \( |N_u| < k \) or \( |N_v| < k \) for any \( v \in N_u \), remain awake. Return.
5) Compute \( C_u = \{v|v \in N_u \text{ and grank}_v < \text{rank}_u\} \).
6) Go to sleep if the following three conditions hold. Remain awake otherwise.
   - Any two nodes in \( C_u \) are connected either directly themselves or indirectly through nodes within \( u \)’s two-hop neighborhood that have \( \text{rank} \) less than \( \text{rank}_u \).
   - Any node in \( N_u \) has at least \( k \) neighbors from \( C_u \).
   - It does not receive a flag.
7) Return.

E. Analysis of GCKN Algorithms

**Theorem:** Node \( u \) will have at least \( k, o_u \) awake neighbors after running GCKN algorithms, if it has \( o_u \) neighbors in the original network.

**Proof:** If \( o_u < k \), all of \( u \)’s neighbors should keep awake (Step 4 of the third part of GCKNF or Step 4 of the second part of GCKNA) and the node will have \( o_u \) awake neighbors.

Otherwise, when \( o_u \geq k \), we prove the theorem by contradiction [22], [23]. Suppose that node \( u \) will not have at least \( k \) awake neighbors after running GCKN algorithms, i.e., we can assume that the \( i \)th lowest ranked (for GCKNF) or granked (for GCKNA) neighbor \( v \) of \( u, i \leq k \), decides to sleep. Then \( C_u \) will have at most \( i - 1 \) nodes that are neighbors of \( u \). But since \( i - 1 < k \), \( v \) cannot go to sleep according to the Step 6 of third part of GCKNF or Step 6 of second part of GCKNA. This is a contradiction. In other words, the \( k \) lowest granked neighbors of \( u \) will all remain awake after running the algorithm, and hence, \( u \) will have at least \( k \) awake neighbors.

**Theorem 2:** Running GCKN algorithms produces a connected-network if the original network is connected.

**Proof:** We prove this theorem by contradiction [22], [23]. Assuming that the output network after running GCKN algorithms is not connected. Then, we put the deleted nodes (asleep nodes decided by GCKN algorithms) back in the network in ascending order of their ranks (for GCKNF) or granks (for GCKNA), and let \( u \) be the first node that makes the network connected again. Note that by the time we put \( v \) back, all the members of \( C_u \) are already present and nodes in \( C_u \) are already connected since they are connected by nodes with \( \text{rank} < \text{rank}_u \) (for GCKNF) or \( \text{rank} < \text{rank}_u \) (for GCKNA). Let
be a node that was disconnected from \( C_u \) but now gets connected to \( C_u \) by \( u \). But this contradicts the fact that \( u \) can sleep only if all its neighbors (including \( v \)) are connected to \( \geq k \) nodes in \( C_u \) (Step 6 of third part of GCKNF or Step 6 of second part of GCKNA).

**Theorem:** GCKN sleep scheduling-based WSN can provide as short as possible transmission path explored by geographic routing when there are mobile sensor nodes.

**Proof:** We prove this by analyzing the resultant topology after running GCKNF or GCKNA. Concerning GCKNF, given that there is a network \( N_{gcknf} \) resulting from GCKNF, based on the algorithm presentation of GCKNF, we can deduce that the neighbor node that is closest to the sink for any node, will be among the awake nodes of the \( N_{gcknf} \) (Step 6 of the third part of GCKNF). In other words, no matter which node the geographic routing chooses to be the first forwarding node, all successor nodes closest to sink can be utilized by the geographic routing. Thus, the length of the first transmission path explored by geographic routing can be as short as possible.

Regarding GCKNA, assume that there is a network \( N_{gckna} \) created by GCKNA. From the algorithm description of GCKNA, we can determine that for any node, say \( u \), if it determines to be asleep, it must make sure that either 1) its all awake 1-hop neighbor nodes are connected by themselves with \( grank < grank_u \), or connected by their two-hop neighbor nodes with \( grank < grank_u \); 2) any of its awake one-hop neighbor nodes should have at least \( k \) neighbor nodes from the subset of the one-hop neighbor nodes with \( grank < grank_u \) (Step 6 of the second part of GCKNA). This means that compared with the asleep nodes, the awake nodes generally have closer geographic distance to the sink. In other words, geographic routing can have access to as many as possible closer neighbor nodes to the sink under the priority of network connectivity after sleep scheduling. Thus, the length of all transmission paths searched by geographic routing can also be as short as possible.

IV. PERFORMANCE EVALUATIONS

A. Evaluation Setup

To evaluate the performance of the proposed GCKN algorithms when applying geographic routing into duty-cycled mobile WSNs, we conduct extensive simulations in NetTopo\(^6\) [39]. We use TPGF [29] as our geographic routing due to the unique desirable characters of TPGF in dealing with the local minimum or hole problem as well as the shortest and multi-path transmission properties of TPGF, which are introduced in Section II. We compare the performance of the proposed GCKN algorithms with CKN and GSS, since CKN and GSS are the only other sleep scheduling algorithms focusing on geographic routing in duty-cycled WSNs, which is also illustrated in Section II. The performance metric is the lengths of the transmission paths searched by TPGF in duty-cycled WSNs employing GCKN, CKN, and GSS, as the length of geographic routing transmission path is widely used to estimate the transmission time, transmission delay, etc. [40]. In addition, the network lifetime\(^7\) of WSNs employing GCKN, CKN, and GSS are also observed to check whether GCKN degrades the network lifetime.

For both GCKNF and GCKNA, the detailed simulation parameters are shown in Table I. The network size is \( 800 \times 600 \) \( \text{m}^2 \). The number of deployed sensor nodes ranges from 100 to 1000 (each time increased by 100) and the value of \( k \) in CKN is changed from 1 to 10 (each time increased by 1). The default transmission radius of each node is 60 \( \text{m} \), and the maximum transmission radius of each node is 120 \( \text{m} \). There is one constant source node deployed at location (50, 50) and we consider two mobility cases: 1) the sink is still at location (750, 550) and all normal sensor nodes randomly move a random number between 100 to 1000 times; 2) all normal sensor nodes are static and the sink moves a random number between 10 and 100 times in 100 different network topologies. The mobility model in both cases is the random waypoint model illustrated in Section III. In addition, the initial energy of each normal node is 100 000 \( \text{mJ} \). The energy consumption of a sensor by transmitting, receiving one byte and transmitting amplifier are 0.0144 \( \text{mJ} \), 0.00576 \( \text{mJ} \), and 0.0288 \( \text{nJ/m}^2 \), respectively [23], [41]. Each packet is 12 bytes long, and each node transmits 1000 packets for each time epoch which is 1 min.

B. Evaluation Results

A snapshot of the first transmission path from the source node to the sink node explored by TPGF in a GCKNF-based WSN when the sink is at different locations is shown in Fig. 1. An example of all transmission paths from the source node to the sink node explored by TPGF in a GCKNA-based WSN when the sink is at different locations is presented in Fig. 2.

1) **GCKNF Versus CKN Versus GSS—Static Sink With Mobile Sensor Nodes:** From Fig. 3(a)-(c), we can clearly see that the average length of the first explored transmission path of TPGF in GCKNF-based WSNs with mobile sensor nodes is nearly always shorter than that in CKN-based WSNs with

\(^6\)NetTopo (available online at http://sourceforge.net/projects/nettopo/) is an open source software on SourceForge for simulating and visualizing WSNs.

\(^7\)The network lifetime is the instant from the network deployment to the instant when the first sensor node runs out of energy.

<table>
<thead>
<tr>
<th>Table I</th>
<th>Evaluation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Parameter value</td>
</tr>
<tr>
<td>Network size</td>
<td>800 x 600 ( \text{m}^2 )</td>
</tr>
<tr>
<td>Sensor nodes number</td>
<td>100-1000</td>
</tr>
<tr>
<td>( k ) in CKN</td>
<td>1-10</td>
</tr>
<tr>
<td>Default transmission radius</td>
<td>60 m</td>
</tr>
<tr>
<td>Maximum transmission radius</td>
<td>120 m</td>
</tr>
<tr>
<td>Source node location</td>
<td>(50, 50)</td>
</tr>
<tr>
<td>Default sink node location</td>
<td>(750, 550)</td>
</tr>
<tr>
<td>Sensor nodes movement times</td>
<td>100-1000</td>
</tr>
<tr>
<td>Sink node movement times</td>
<td>10-100</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random waypoint</td>
</tr>
<tr>
<td>Initial energy</td>
<td>100000 ( \text{mJ} )</td>
</tr>
<tr>
<td>Transmission energy</td>
<td>0.0144 ( \text{mJ} )</td>
</tr>
<tr>
<td>Reception energy</td>
<td>0.00576 ( \text{mJ} )</td>
</tr>
<tr>
<td>Transmission amplifier energy</td>
<td>0.0288 ( \text{nJ/m}^2 )</td>
</tr>
<tr>
<td>Packet size</td>
<td>12 bytes</td>
</tr>
<tr>
<td>Packet number</td>
<td>1000</td>
</tr>
<tr>
<td>Time epoch</td>
<td>1 minute</td>
</tr>
</tbody>
</table>
mobile sensor nodes and almost the same as that in GSS-based WSNs with mobile sensor nodes. It is because of the awake nodes that are closest to the sink in GCKNF-based WSNs, whereas such nodes do not usually appear in CKN-based WSNs. And the potential awake nodes closest to sink in GCKNF-based WSNs with mobile sensor nodes and that in GSS-based WSNs are almost the same. In addition, the average network lifetime in GCKNF-based WSNs with mobile sensor nodes is almost the same as that in CKN-based WSNs and almost the same as that in GSS-based WSNs with mobile sensor nodes. That is also due to that there are much more awake nodes closest to sink in GCKNF-based WSNs, compared with that in CKN-based WSNs. And GCKNF-based WSNs and GSS-based WSNs share the same criteria to choose the awake nodes closest to sink. Furthermore, the average network lifetime in GCKNF-based WSNs with a mobile sink shown in Fig. 4(d) is nearly the same as that in CKN-based WSNs with a mobile sink shown in Fig. 4(e) and much higher than that in GSS-based WSNs presented in Fig. 4(f).

3) GCKNA Versus CKN Versus GSS—Static Sink With Mobile Sensor Nodes: Fig. 5(a)–(c) describe the average length of all transmission paths explored by TPGF in GCKNA and CKN as well as GSS-based WSNs with mobile sensor nodes. From these three figures, we can clearly see that the average lengths of all transmission paths explored by TPGF in GCKNA-based WSNs with mobile sensor nodes are mostly much shorter than that in CKN and GSS-based WSNs. Moreover, there is not too much difference regarding the average network lifetime in GCKNA-based WSNs with mobile sensor nodes and the average network lifetime in CKN-based WSNs with mobile sensor nodes, which are shown in Fig. 5(d) and (e), respectively. In addition, both the average network lifetimes in GCKNA and CKN-based WSNs with mobile sensor nodes are greatly higher than that in GSS-based WSNs presented in Fig. 5(f).

4) GCKNA Versus CKN Versus GSS—Mobile Sink With Static Sensor Nodes: Fig. 6(a)–(c) presents the average lengths of all transmission paths explored by TPGF in GCKNA and CKN as well as GSS-based WSNs with a mobile sink. From these three figures, we can also clearly see that the average length of all transmission paths explored by TPGF in GCKNA-based WSNs with a mobile sink is almost always much shorter than that in CKN and GSS-based WSNs. That also results from the more awake closer nodes to sink in GCKNA-based WSNs than that in CKN and GSS-based WSNs. Apart from that, it is hard to distinguish the average network lifetime in GCKNA-based WSNs with a mobile sink from that in CKN-based WSNs with a mobile sink, which are shown in Fig. 6(d) and (e). And the average network lifetime in GSS-based WSNs with a mobile sink presented in Fig. 6(f) is quite lower than that in GCKNA and CKN-based WSNs.

C. Application Scenarios

Regarding both case 1) and case 3) evaluated above (i.e., static sink with mobile sensor nodes), they can particularly be applied to enhance the data transmission scenarios such as information center (i.e., source node) needs the robots (i.e., sensor nodes with mobility) randomly move to monitor and gather the environment data and then transmit these data along the path between the information center and a remote base station (i.e., static sink node).
Fig. 3. (a) Average length of the first transmission path explored by TPGF in GCKNF-based WSNs with mobile sensor nodes, (b) in CKN-based WSNs with mobile sensor nodes, (c) in GSS-based WSNs with mobile sensor nodes, (d) average network lifetime in GCKNF-based WSNs with mobile sensor nodes, (e) in CKN-based WSNs with mobile sensor nodes, and (f) in GSS-based WSNs with mobile sensor nodes.

Fig. 4. (a) Average length of the first transmission path explored by TPGF in GCKNF-based WSNs with a mobile sink, (b) in CKN-based WSNs with a mobile sink, (c) in GSS-based WSNs with a mobile sin, (d) average network lifetime in GCKNF-based WSNs with a mobile sink, (e) in CKN-based WSNs with a mobile sink, and (f) in GSS-based WSNs with a mobile sink.

With respect to case 2) and case 4) in the previous subsection (i.e., mobile sink with static sensor nodes), they can specifically be employed to speed up services in some information providing applications. For example, a customer with a mobile phone (i.e., mobile sink node) can randomly move and the information center (i.e., source node) needs to provide the traffic information or house monitoring information gathered by various location predetermined environment monitoring sensors (i.e., static sensors) as fast as possible.
For all the above four cases, we can further observe that when there are more number of sensor nodes, the advantage of GCKN algorithms will become more obvious. That is because higher network density will enable GCKN algorithms to create more favorable nodes for TPGF to utilize.

V. CONCLUSION

In this paper, we have explored geographic routing in duty-cycled mobile WSNs and proposed two geographic-distance-based connected-k neighborhood (GCKN) sleep scheduling algorithms for geographic routing schemes to be applied into...
duty-cycled mobile WSNs which can incorporate the advantage of sleep scheduling and mobility. The first geographic-distance-based connected-neighborhood for first path (GCKNF) sleep scheduling algorithm minimizes the length of first transmission path explored by geographic routing in duty-cycled mobile WSNs. The second geographic-distance based connected-neighborhood for all paths (GCKNA) sleep scheduling algorithm reduces the length of all paths searched by geographic routing in duty-cycled mobile WSNs. In duty-cycled mobile WSNs, from the view of sleep scheduling, GCKNF and GCKNA do not require the geographic routing to change its original geographic forwarding mechanism, and they both consider the connected-k neighborhood requirement and geographic routing requirement to change the asleep or awake state of sensor nodes. Detailed design of both GCKNF and GCKNA as well as further theoretical analysis and evaluation with respect to GCKNF and GCKNA has been shown in this paper. They demonstrate that GCKNF and GCKNA are very effective in shortening the length of the transmission path explored by geographic routing in duty-cycled mobile WSNs compared with the CKN sleep scheduling algorithm and the GSS algorithm. Our work has shown that sleep scheduling is a worthy research direction to adapt geographic forwarding methods into duty-cycled mobile WSNs.

REFERENCES

Chunsheng Zhu (S’12) received the B.E. degree in network engineering from Dalian University of Technology, Dalian, China, in 2010, and the M.Sc. degree in computer science from St. Francis Xavier University, Antigonish, NS, Canada, in 2012. He is currently working toward the Ph.D. in the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada. His research interests include wireless sensor networks and cloud computing.

Laurence T. Yang (M’97) received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 2006. Currently, he is a Professor in the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. His research interests include parallel and distributed computing, and embedded and ubiquitous/pervasive computing. His research is supported by the National Sciences and Engineering Research Council, Canada, and the Canada Foundation for Innovation.

Lei Shu (M’07) received the B.Sc. degree in computer science from South Central University for Nationalities, Wuhan, China, in 2002, the M.Sc. degree in computer engineering from Kyung Hee University, Seoul, Korea, in 2005, and the Ph.D. degree in computer engineering from the Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland, in 2010. Currently, he is a Full Professor at the College of Electronic Information and Computer, Guangdong University of Petrochemical Technology, Maoming, China, and also the Vice-Director of the Guangdong Petrochemical Equipment Fault Diagnosis (PEFD) Key Laboratory. He was a Specially Assigned Research Fellow in the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan. His research interests include wireless sensor network, sensor network middleware, multimedia communication, and security. Dr. Shu is a member of the Association for Computing Machinery.

Victor C. M. Leung (S’75–M’89–SM’97–F’03) received the B.A.Sc. (Hons.) and Ph.D. degrees in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977 and 1981, respectively. Currently, he is a Professor and TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering, UBC. His research interests include wireless networks and mobile systems. Dr. Leung is a Registered Professional Engineer in the Province of British Columbia, Canada. He is a Fellow of the Royal Society of Canada, a Fellow of the Engineering Institute of Canada, and a Fellow of the Canadian Academy of Engineering.

Joel J. P. C. Rodrigues (S’01–M’06–SM’06) received the B.Sc. degree (licentiate) in informatics engineering from the University of Coimbra, Coimbra, Portugal, in 1995, the M.Sc. degree in knowledge management from the University of Beira Interior, Covilhã, Portugal, in 2002, and the Ph.D. degree in informatics engineering from University of Beira Interior, Covilhã, Portugal, in 2006. Currently, he is a Professor in the Department of Informatics, University of Beira Interior, and a Researcher at the Instituto de Telecomunicações, Covilhã, Portugal. His research interests include sensor networks, e-health, e-learning, vehicular delay-tolerant networks, and mobile and ubiquitous computing. Dr. Rodrigues is a Licensed Professional Engineer (as a Senior Member), a member of the Internet Society, an International Academy, Research and Industry Association Fellow, and a Senior Member of the Association for Computing Machinery.

Lei Wang (M’01) received the B.S., M.S., and Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 1995, 1998, and 2001, respectively. Currently, he is a Full Professor in the Department of Network Engineering, Dalian University of Technology, Dalian, China. He was a member of the Technical Staff with Bell Labs Research China, Beijing, China (2001–2004); a Senior Researcher at Samsung, Seoul, Korea (2004–2006); a Research Scientist at Seoul National University, Seoul (2006–2007); and a Research Associate at Washington State University, Pullman, WA, USA (2007–2008). His research interests include wireless ad hoc networks, sensor networks, and embedded systems. Dr. Wang is a member of the Association for Computing Machinery and a Senior Member of the China Computer Federation (CCF).