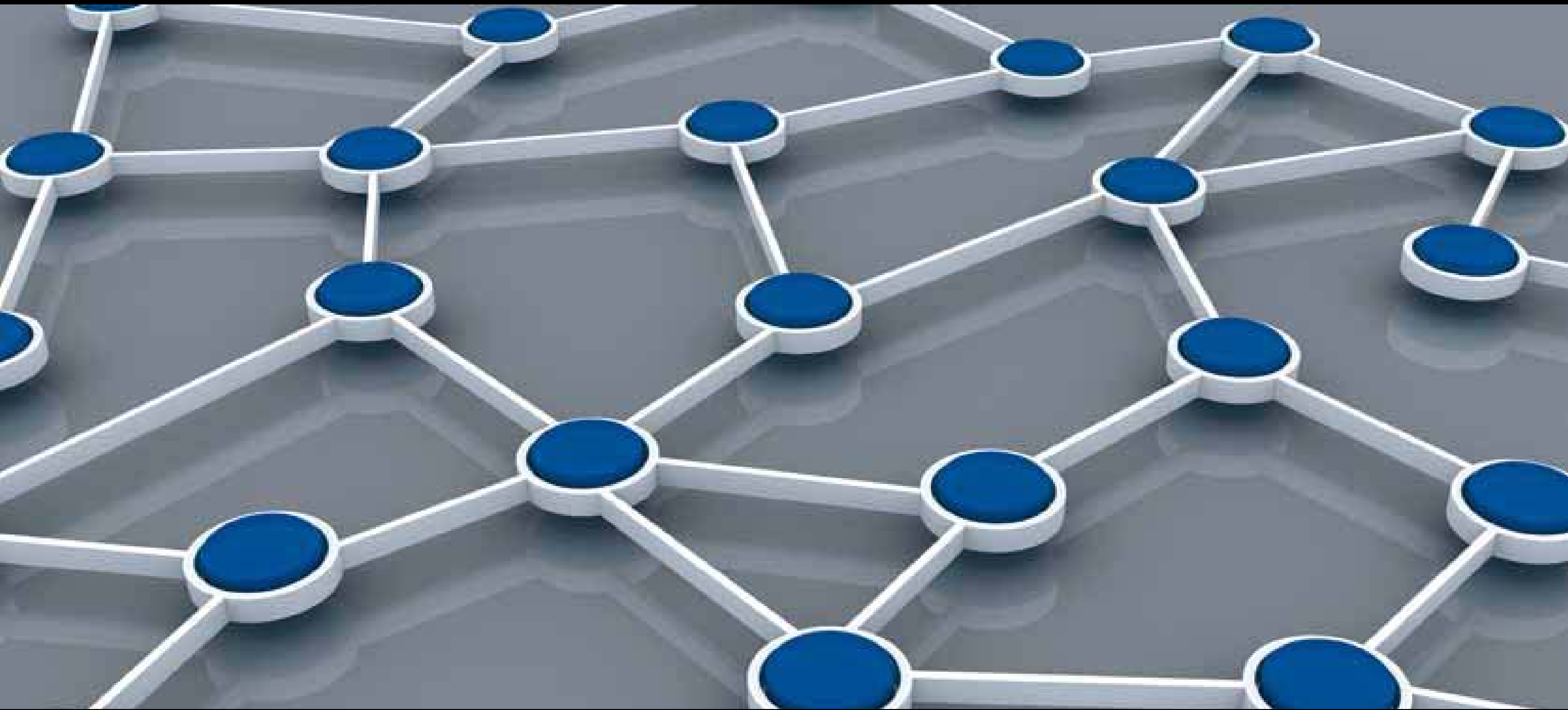
International Journal of Distributed Sensor Networks

Mobile Sensing and Actuating with Ubiquitous Computing

Guest Editors: Mo Li, Yunhao Liu, and Xiang-Yang Li



Mobile Sensing and Actuating with Ubiquitous Computing

International Journal of Distributed Sensor Networks

Mobile Sensing and Actuating with Ubiquitous Computing

Guest Editors: Mo Li, Yunhao Liu, and Xiang-Yang Li

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 296396, 2 pages
doi:10.1155/2012/296396

Editorial

Mobile Sensing and Actuating with Ubiquitous Computing

Mo Li,[1] Yunhao Liu,[2] and Xiang-Yang Li[3]

[1] *School of Computer Engineering, Nanyang Technological University, Singapore 639798*
[2] *TNLIST, Tsinghua University, Beijing 100084, China*
[3] *Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA*

Correspondence should be addressed to Mo Li, limo@ntu.edu.sg

Received 27 May 2012; Accepted 27 May 2012

Advanced wireless sensing and networking techniques enable mobile sensing and actuating with ubiquitous computing on numerous smart sensors, mobile devices, RFIDs, and so forth, which revolutionize a variety of application areas with unprecedented pervasive physical world instrumentation. The development and deployment of such mobile sensing and actuating systems has become significant across a variety of application domains, including environmental surveillance, traffic and logistics, personal area networks, smart buildings, and emergent navigation. The major challenges come with the required system sustainability, mobility, scalability, and fault resilience in ensuring constant services across dynamic application environment. In this special issue, we have accepted a few papers that address the above key aspects in providing mobile sensing and actuating services.

The paper "*InContexto: multisensor architecture to obtain people context from smartphones*" describes a distributed architecture, called inContexto, to recognize user context information using mobile phones. The proposed framework targets at inferring physical actions performed by users such as walking, running, and still.

The paper "*Statistically bounding detection latency in low-duty-cycled sensor networks*" studies the problem of bounding event detection delays when the sensor network works in low-duty-cycled mode. A novel approach for statistically bounding event detection latency is proposed, and key issue like the wakeup scheduling of sensor nodes as well as minimization of wakeup activity is addressed in this paper.

The paper "*Spatial and temporal correlations-based routing algorithm in intermittent connectivity human social network*" proposes the idea of trip history model (THM) which establishes a model on a single person's mobility, and then a spatial and temporal correlations-based routing algorithm (STC) to efficiently transmit data. In STC, the node delivery probability is calculated according to both a node's current moving prediction and its history record to give guidance for message transmission.

The paper "*Dynamic key-updating: privacy-preserving authentication for RFID systems*" proposes a strong and lightweight RFID private authentication protocol, SPA. By designing a novel key updating method, SPA achieves the forward secrecy with an efficient key search algorithm.

The paper "*Improving accuracy for 3D RFID localization*" develops an indoor 3D RFID localization system based on active tag array. The geometric mean is used in this paper to filter the explicit 3D location information with high accuracy. The experimental results demonstrate the system efficiency in tracking objects and improving the localization accuracy.

The paper "*An extended virtual force-based approach to distributed self-deployment in mobile sensor networks*" aims at overcoming the connectivity maintenance and node stacking problem of the traditional virtual force-based algorithms (VFAs). The paper investigates an extended virtual force-based approach for achieving the ideal node deployment.

The paper "*Time-independent data collection protocol in mobility-assistant wireless sensor networks with duty cycles*" studies the MS discovery mechanism as well as the factors that affect the efficiency of data collection in duty-cycled sensor networks. The paper provides a solution to the control problem of how to optimally adjust the system parameters of the sleep/wake scheduling protocol to maximize the network lifetime, subject to a constraint on the expected residual contact time.

The paper "*An energy-efficient CKN algorithm for duty-cycled wireless sensor networks*" investigates the unexplored energy consumption of the CKN algorithm. By building a probabilistic node sleep model, this paper computes the probability that a random node goes to sleep and obtains a lower epoch bound that keeps the network more energy efficient with longer lifetime when running the CKN algorithm.

The paper "*Rendezvous data collection using a mobile element in heterogeneous sensor networks*" studies the rendezvous data collection problem for the mobile element (ME) in heterogeneous sensor networks where data generation rates of sensors are distinct. An $O(n \log n)$ algorithm is proposed to approach the optimal rendezvous points (RPs) to collect global sensory data as well as the optimal data collection trajectory for the mobile relay to gather the cached data from RPs.

The paper "*Hybrid position-based and DTN forwarding for vehicular sensor networks*" addresses data delivery challenge in the possible intermittently connected vehicular sensor networks by combining position-based forwarding strategy with store-carry-forward routing scheme from delay-tolerant networks. The proposed routing method makes use of vehicle driving direction to determine whether holding or forwarding the packet.

Mo Li
Yunhao Liu
Xiang-Yang Li

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 186146, 10 pages
doi:10.1155/2012/186146

Research Article

Hybrid Position-Based and DTN Forwarding for Vehicular Sensor Networks

Fan Li,[1] Lei Zhao,[1] Xiumei Fan,[1] and Yu Wang[2]

[1] *School of Computer Science, Beijing Institute of Technology, Beijing 100081, China*
[2] *Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA*

Correspondence should be addressed to Fan Li, fli@bit.edu.cn

Received 30 January 2012; Accepted 24 March 2012

Academic Editor: Mo Li

Efficient data delivery in vehicular sensor networks is still a challenging research issue. Position-based routing protocols have been proven to be more suitable for dynamic vehicular networks or large-scale mobile sensor networks than traditional ad hoc routing protocols. However, position-based routing assumes that intermediate nodes can always be found to set up an end-to-end connection between the source and the destination; otherwise, it suffers from network partitions which are very common in vehicular networks and leads to poor performances. This paper addresses data delivery challenge in the possible intermittently connected vehicular sensor networks by combining position-based forwarding strategy with store-carry-forward routing scheme from delay tolerant networks. The proposed routing method makes use of vehicle driving direction to determine whether holding or forwarding the packet. Experimental results show that the proposed mechanism outperforms existing position-based solutions in terms of packet delivery ratio.

1. Introduction

Vehicular sensor network (VSN) [1–3] is emerging as a new sensor network application for monitoring the physical world of urban environments, which is also a type of vehicular ad hoc network (VANET) [4]. VSN is a high-speed mobile wireless sensor network containing set of smart vehicles which are equipped with different types of on-broad sensors and can communicate with each other or predeployed road side units via wireless medium. VSNs aim to provide ubiquitous, efficient sensing and networking capabilities for mobile users on the road and also support a variety of urban monitoring and safety applications such as cooperative traffic monitoring, control of traffic flows, blind crossing, prevention of collisions, detection of toxic chemicals, and road surface monitoring. To realize this objective, the Federal Communications Commission (FCC) has allocated 75 MHz of spectrum for short-range vehicle-to-vehicle (V2V) or vehicle-to-roadside communications. IEEE also formed the new IEEE 802.11p task group which adds wireless access in vehicular environments (WAVE) [5]. Because of the increasing popularity of mobile wireless devices and smart vehicles, in the near future, VSNs will become one of the important components of the next generation of Internet and Internet of Things.

Different from traditional wireless sensor networks, VSN has its unique characteristics which pose new challenges in the design of networking protocols, especially for routing protocols. For example, as vehicles move at relevant high speeds, the topology of network is highly dynamic and the network topology may be even frequently disconnected. Vehicles rely on wireless links to relay data, and communication bandwidth is consequently limited and unstable [6, 7]. Last but not least, vehicles are typically not affected by strict energy constraints and can have powerful processing units and storages. Even though some of the existing ad hoc and sensor network routing protocols can still be applied to vehicular networks (both VSNs and VANETs), simulation results [8–11] have showed that they suffer from poor performances because of the fast movements of vehicles and limited chances for information exchanges. Therefore, finding and maintaining stable route is a very challenging task in vehicular networks.

Considering the highly dynamic nature of node mobility in vehicular networks, many different routing protocols (traditional topology-based ad hoc routing, cluster-based routing, or position-based routing) have been proposed for VANETs or VSNs in recent years [12]. However, most of them assume that intermediate nodes can always be found to set up an end-to-end connection between the source and the destination, which is not the case in many VSN environments. On the other hand, the system can tolerate up to seconds or minutes of delay in some VSN applications, for example, those to collect information of the available parking lots or the road surface conditions. Such applications can be supported by network protocols designed for delay-tolerant networks (DTNs) and enable the new research direction in the routing design for VSNs.

In this paper, by leveraging the DTN technology, we study hybrid forwarding schemes for efficient data delivery in VSNs. Specifically, when a vehicle reports a delay-tolerant data to a fixed sink in the network, we propose hybrid forwarding techniques to efficiently route the packet by combining position-based forwarding (such as GFG [13] and GPSR [14]) and the idea of store-carry-forward from DTNs [15–17] where the data can be incrementally moved and stored throughout the network. In our proposed method, when the packet enters perimeter mode, according to the driving direction of the vehicle and the delivery direction of the packet, the vehicle will determine to either hold or deliver this packet. Our simulation results show that this proposed method outperforms existing position-based solutions in VSNs.

The rest of this paper is organized as follows. Section 2 briefly reviews related work on VANET/VSN routing. Section 3 describes the principle idea of our proposed method, and Section 4 presents simulation results on comparison of proposed scheme with GPSR. Finally, Section 5 concludes the paper by discussing some possible future work.

2. Related Works

Because of high-speed vehicular mobility and unreliable channel conditions, data delivery and ensuring delivery is one of the key research issues to both vehicular ad hoc networks and vehicular sensor networks. Routing in vehicular networks has been studied recently, and many different VANET/VSN routing protocols have been proposed.

Node movements in vehicular networks are usually restricted in just bidirectional movements constrained along roads and streets. Thus it is reasonable to use geographical location information obtained from GPS devices on-board the vehicles to make routing decisions. This fact receives support from a number of studies that compare the performance of topology-based routing (such as AODV and DSR) against position-based routing strategies in urban as well highway traffic scenarios [10, 18]. Therefore, position-based routing (geographic routing) has been identified as a more promising routing paradigm for vehicles. In position-based routing, forwarding decisions are made based on location information. For example, *greedy routing* always forwards the packet to the node that is geographically closest to the destination. *Greedy-face-greedy* (GFG) [13] and *greedy perimeter stateless routing* (GPSR) [14] both combine the greedy forwarding with face routing by using face routing to get out of the local minimum where greedy forwarding fails. It works best in a free open-space scenario with evenly distributed nodes but suffers from various obstacles in city conditions. To fix such problems, several position-based routing protocols [19, 20] specifically designed for VANETs in city environments were proposed. For example, *greedy perimeter coordinator routing* (GPCR) [20] utilizes the fact that the nodes at a junction in the street follow a natural planar graph. Thus a restricted greedy algorithm can be followed as long as the nodes are in a street. Junctions are the only places where actual routing decisions are taken. Therefore, packets should always be forwarded to a node on a junction rather than being forwarded across the junction.

Most existing position-based or topology-based VANET/VSN routing protocols [10, 14, 18–22] assume that intermediate nodes can be found to set up an end-to-end connection; otherwise, the data packet will be dropped by the protocols. However, finding end-to-end connections sometimes is extremely difficult for a sparse vehicular network. On the other hand, the high mobility of vehicular networks introduces opportunities for mobile vehicles to connect with each other intermittently. There are ample opportunities for vehicles to set up a short path with few hops in a highway model, as shown in [9]. In addition, a moving vehicle can store and carry the packet when routes do not exist and forward the packet to the new receiver that moves into its vicinity. This *store-carry-forward* fashion enables a new type of routing protocols: *delay-tolerant network* (DTN) routing [23], which can deliver packets to the destination without an end-to-end connection for delay-tolerant applications. Recently, there are also new trends to enhance the traditional store-carry-forward DTN routing for VANET/VSN applications by considering the vehicular mobility or geographic characteristics of vehicular networks. LeBrun et al. [24] described several opportunistic forwarding schemes for VANETs where location or relative velocity of vehicles is used for relay selection. Zhao and Cao [25] also proposed a vehicle-assisted data delivery (VADD) protocol, which calculates the predicable vehicle mobility and takes it into consideration to choose the relay with the lowest data delivery delay. Lee et al. [2] proposed a proactive urban monitoring system which adopts an opportunistic diffusion scheme by exploiting vehicle mobility. Lee et al. [2] also proposed a DTN-based routing protocol for VSNs where each node delivers its packets to a neighboring node which is estimated as the best carrier based on a utility function of distances to the destinations and sizes of its packets. Leontiadis and Mascolo [26] proposed a DTN routing algorithm that exploits the availability of suggested route information from the navigation system in order to opportunistically route a packet to certain geographical location. Cheng et al. [27] also proposed a hybrid geographic and DTN routing, GeoDTN+Nav, which has three forwarding modes: greedy mode, perimeter mode, and DTN mode. The first two modes are the same with those in GPSR and can be switched to each other. In the perimeter mode, if the network is disconnected,

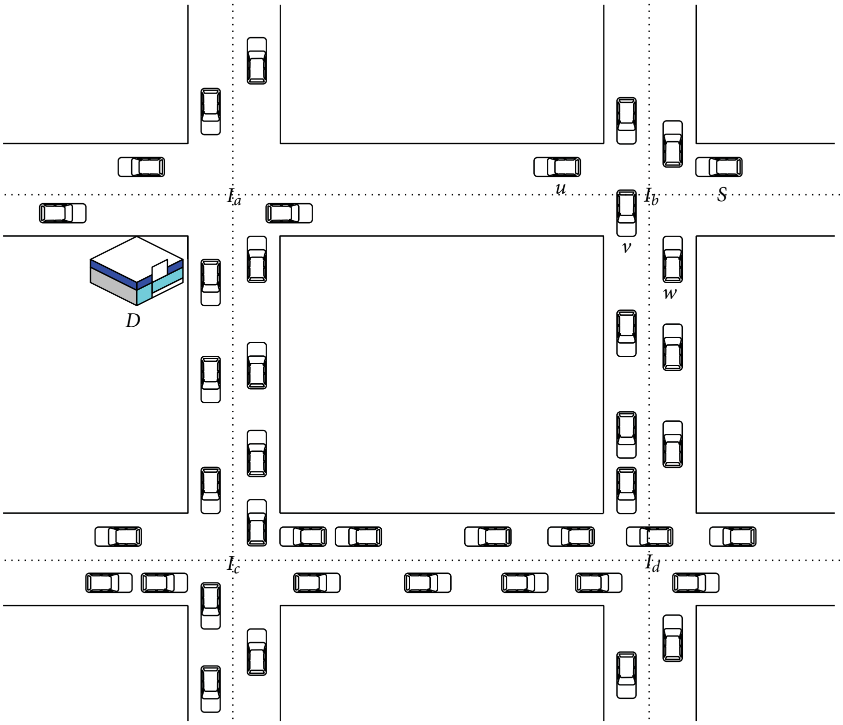


Figure 1: An example of routing problem in VSNs: vehicle S wants to send a packet to a fixed sink D at the parking deck.

GeoDTN+Nav switches to DTN mode where the packet is stored and carried. To determine when to switch to DTN mode, the protocol calculates a "switch score" (based on the hop count a packet has traveled, estimated delivery quality, and direction of neighbors) and compares with a predefined threshold. When current vehicle finds a vehicle closer to the destination, the packet switches from DTN mode to greedy mode.

Notice that our proposed hybrid protocol also exploits the combination of position-based routing and DTN routing as in [2, 24, 26, 27]. However, we take different approaches to consider the traffic, geographical location, and driving direction information.

3. Hybrid Position-Based and DTN Forwarding

In this section, we will present the details of our proposed hybrid position-based and DTN forwarding strategy which is combining GPSR protocol [14] with DTN routing.

3.1. Assumptions. Position-based routing protocol (such as GPSR) is usually used in well-connected networks, where the destination node is set in advance and it broadcasts its position information periodically to the whole network so that every node gets the position information of the destination. However, vehicular sensor networks sometimes are sparse and disconnected networks. In this paper, we assume that vehicles are equipped with an onbroad navigation system loaded with digital maps and a GPS receiver, which provides the location service for the whole region. Mobile vehicle can obtain its location, velocity, and direction through the GPS and the unique location information of any fixed site via the navigation system as well. In addition, vehicles can communicate with nearby vehicles through short-range wireless channel (100 m–250 m) and learn their location information through periodic beacon messages. We assume that the packets of sensing data are generated at mobile vehicles and the destinations are fixed sinks.

3.2. Motivation. Let us first consider an example shown in Figure 1, in which a vehicle S sends a packet of its sensing data to the fixed sink D at the corner of intersection I_a. According to the GPSR protocol, vehicle u will receive the packet (because u is S's closest neighbor to D), but u cannot find a closer neighbor to D; thus the packet enters *perimeter mode* at u. Based on the right-hand rule, u will deliver the packet to v and the packet will be relayed through a path $I_b \rightarrow I_d \rightarrow I_c \rightarrow I_a$ to the destination. In this situation, there are enough vehicles along these three segments so that packet can be delivered to the destination. However, when the segment between intersections I_c and I_d is blocked by either traffic lights or a sudden accident, the route may become disconnected and the packets need to be detoured to a longer route or even be dropped. On the other hand, vehicle u is driving towards the destination, even it currently does not have a nice relay node but it can carry the packet and may deliver it to destination by itself within much shorter time than the route of $I_b \rightarrow I_d \rightarrow I_c \rightarrow I_a$. Therefore, it will be nice to exploit such enhancement to GPSR by considering possible store-carry-forward options.

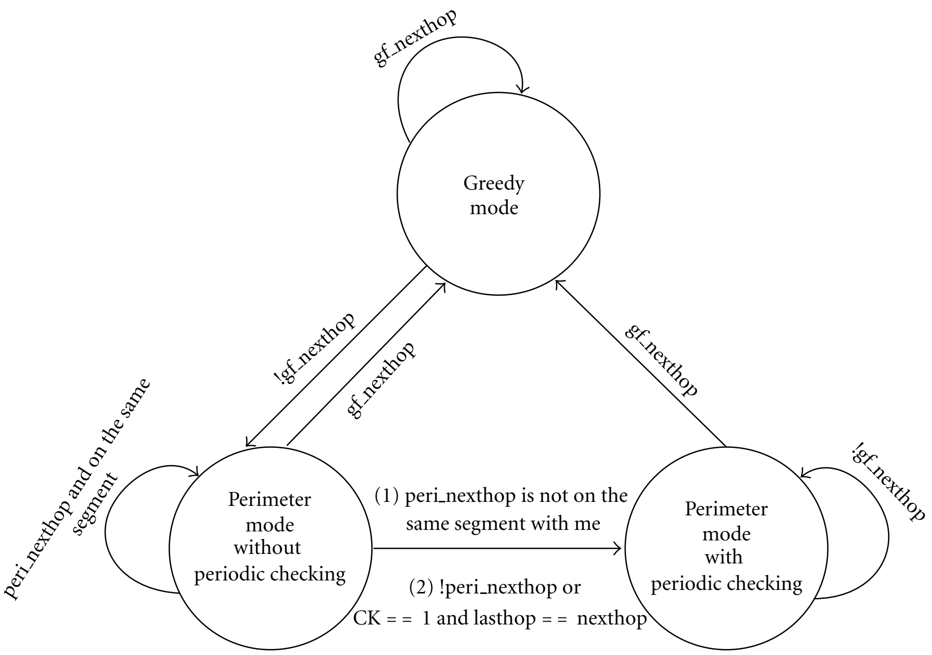

gf_nexthop
Greedy mode
!gf_nexthop
gf_nexthop
gf_nexthop
peri_nexthop and on the same segment
Perimeter mode without periodic checking
(1) peri_nexthop is not on the same segment with me
(2) !peri_nexthop or CK = = 1 and lasthop = = nexthop
Perimeter mode with periodic checking
!gf_nexthop

Figure 2: Transitions among the three possible statuses.

3.3. Detailed Hybrid Forwarding Mechanism for VSNs. Our proposed mechanism is based on a position-based routing, GPSR, and the idea of store-carry-forward from DTN routing. In sparsely connected vehicular sensor networks, when a vehicle finds no better neighbors to be the next hop to relay during geographic forwarding, it can store and carry the packet. The key issues are when and how to store and carry the packet and which next hop node to choose for relaying the packet. The basic idea of our approach is smartly switching between position-based forwarding and store-carry forwarding based on the current traffic situation and locations of neighboring vehicles.

In our design, there are three possible statuses: *greedy mode*, *perimeter mode* without periodic checking, and *perimeter mode* with periodic checking, as shown in Figure 2. Initially, all packets are in *greedy mode* at the source node. When a vehicle receives a packet and the destination is one of current neighbors, it immediately forwards the packet to the destination. Otherwise, it checks the current status of the packet.

When a vehicle receives a packet in *greedy mode*, it tries to forward the packet based on greedy forwarding. If it can find a better next hop (*gf_nexthop*) based on location, it forwards to *gf_nexthop*. Otherwise, it enters *perimeter mode* and needs to further determine whether to store a copy and send the packet or just hold the current packet.

When a vehicle receives a packet in *perimeter mode*, if it has a closer next hop (*gf_nexthop*), the packet switches back to *greedy mode* and is forwarded to *gf_nexthop* based on greedy forwarding. Otherwise, the vehicle carries a copy of the packet and sends the packet to the next hop (*peri_nexthop*) selected by right-hand rule, if it exists. However, not all of the packets in the *perimeter mode* must be carried. For example, in Figure 1, vehicle v finds its *peri_nexthop* (w) on the $I_b \rightarrow I_d$ segment. No matter what the direction of *peri_nexthop* (w) is, v can just send the packet to w and do not need to carry the packet. Here, we use the segment of *peri_nexthop* to determine whether to carry the packet or not. If both the current vehicle and its next hop are on the same segment, there is no need to carry the packet any more.

When the packet is carried by a vehicle (happening in *perimeter mode*), the vehicle needs to check its neighbor list periodically to see whether there is a possible next hop towards the destination. If there is a closer neighbor, it can switch back to greedy forwarding immediately. To save the store space, each packet is held for at most *MAXCTTL* time units. When the timer *CTTL* expires, the packet will be discarded.

The detailed forwarding algorithm and periodic checking mechanism are given as Algorithms 1 and 2, respectively. Figure 2 illustrates the possible transitions among three statuses. Here, *CK* is used to remember whether a packet has been switched from *perimeter mode* with periodic checking to *greedy mode*.

3.4. Examples for Close Look. Next, we use examples shown in Figure 3 (a close look at the intersection I_b from Figure 1) to explain how our algorithm works on a sparse segment when current vehicle v and the next hop vehicle u are on the same direction or on reverse directions.

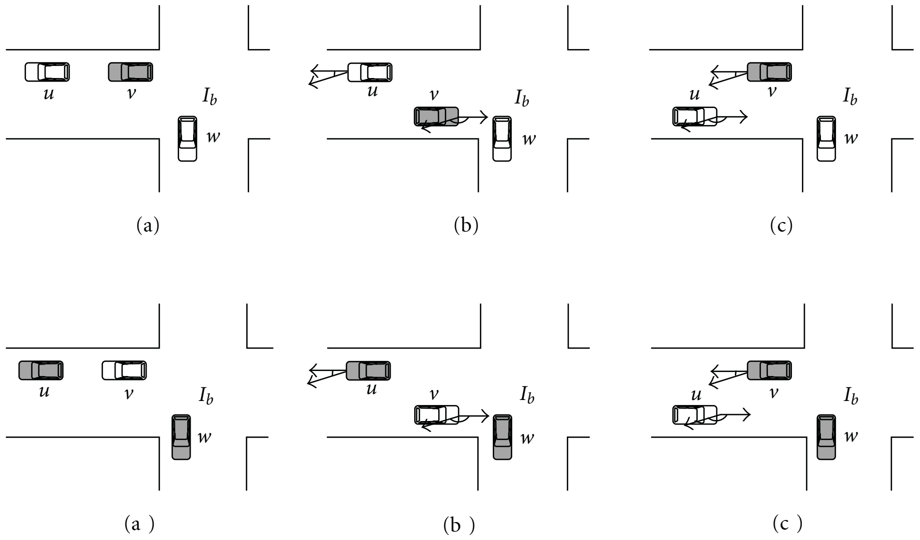
Same Direction Case. Figure 3(a) illustrates the example when vehicles u and v are driving in the same direction towards the destination. When vehicle u receives a packet from v, the packet enters *perimeter mode* and u will send it back to v. When v receives the packet again (but in

```
1:  when generate a new packet p of sensing data:
2:  $mode = greedy\_mode$ and $CK = 0$
3:  continue to Line 5
4:  when receive a data packet p:
5:  if the destination of p is a current neighbor then
6:      forward the packet p to the destination
7:  else
8:      if the packet p in greedy_mode then
9:          if gf_nexthop is found then
10:             $CK = 0$
11:             $lasthop = my\_id$
12:             $nexthop = gf\_nexthop$
13:             send the packet p to nexthop
14:         else
15:             $nexthop = peri\_nexthop$
16:             $mode = perimeter\_mode$
17:             if $CK == 1$ and $lasthop == nexthop$ or
                 peri_nexthop is not found then
18:                 continue to hold the packet p
19:                 enter the Periodic Checking
20:             else
21:                 $CK = 0$
22:                 $lasthop = my\_id$
23:                 if nexthop is not on the same segment with me
                     then
24:                     store a copy of the packet p
25:                     enter the Periodic Checking
26:                 end if
27:                 send the packet p to nexthop
28:             end if
29:         end if
30:     else if the packet p in perimeter_mode then
31:         if gf_nexthop is found then
32:             $lasthop = my\_id$
33:             $nexthop = gf\_nexthop$
34:             $mode = greedy\_mode$
35:             send the packet p to nexthop
36:         else
37:             $nexthop = peri\_nexthop$
38:             if peri_nexthop is not found then
39:                 continue to hold the packet p
40:                 enter the Periodic Checking
41:             else
42:                 $lasthop = my\_id$
43:                 if nexthop is not on the same segment with me
                     then
44:                     store a copy of the packet p
45:                     enter the Periodic Checking
46:                 end if
47:                 send the packet p to nexthop
48:             end if
49:         end if
50:     end if
51: end if
```

Algorithm 1: Hybrid forwarding scheme.

```
1:  **Initialization**:
2:  $CTTL = MAXCTTL$
     ***Periodic Checking***: for any packet $p$ carried
3:  $CK = 1$
4:  **if** $CTTL > 0$ **then**
5:      $CTTL - -$
6:      **if** $gf\_nexthop$ is found **then**
7:          $nexthop = gf\_nexthop$
8:          **if** $nexthop$'s direction is different with $mine$
              and $nexthop == lasthop$ and $my\_angle <$
              $nexthop\_angle$ **then**
9:              continue to hold the packet
10:         **else**
11:             $mode = greedy\_mode$
12:             send the packet to $nexthop$
13:         **end if**
14:     **else**
15:         continue to hold the packet
16:     **end if**
17: **else**
18:     delete the packet form the buffer
19: **end if**
20: Repeat *Periodic Checking* at the next checking interval
```

Algorithm 2: Periodic checking.



(a) (b) (c)

(a) (b) (c)

Figure 3: Examples of three different cases on the same segment: ((a) and (a′)) u, v on the same direction; (b, b′, c, and c′) u, v on reverse directions. Here the shaded vehicles hold a copy of the packet.

perimeter mode), it finds vehicle w is the next hop and they are not on the same segment. According to Algorithm 1 (Line 43), v stores a copy of the packet and forwards the packet to w. Then the copy of the packet at v enters the periodic checking and marks the *CK* bit to 1. This means that the packet has experienced the periodic checking. v finds u is its best neighbor (Algorithm 2: Line 6) and they are driving in the same direction (Algorithm 2: Line 11), so v sends the copy to u again. When u receives the packet, it should hold this packet instead of sending it back (otherwise causing a routing loop). Since the *CK* bit is 1 and *nexthop* = *lasthop* (Algorithm 1: Line 17), u will only hold the packet. Therefore, there will be two copies of the original packet (w and u) as shown in Figure 3(a′) which can explore both possible routes (DTN one and perimeter one).

Reverse Direction Case. In Figures 3(b) and 3(c), vehicles u and v are driving in opposite direction. Our proposed scheme will make use of the angle (i.e., *my_angle*), which is constituted by the vehicular driving direction and the destination's direction (as shown in Figures 3(b) and 3(c)), to determine whether holding or forwarding the packet. Recall that in these examples the destination (i.e., the sink) is in the left-bottom corner of these figures. Like the previous example, when the packet reaches *v* again, *v* stores a copy and sends packet to *w*, then enters periodic checking. *v* will find *u* is its next hop which is closer to destination. As shown in Figure 3(b), the relationship between *u* and *v* (*my_angle* > *nexthop_angle*) does not satisfy Line 8 of Algorithm 2. Thus, *v* will send the packet to *u* again. The final result is same with the last example, as shown in Figure 3(b′). On the other hand, if *u* and *v* are driving as Figure 3(c) shows, since *nexthop* = *lasthop* = *u* and *my_angle* < *nexthop_angle*, which satisfies Line 8 of Algorithm 2, *v* will continue holding the packet and driving towards the destination. Result is shown in Figure 3(c′). In both cases, the routing loop is prevented.

Notice that our proposed scheme can prevent certain types of routing loops. However, due to high mobility of nodes in vehicular sensor networks, it still possible that other types of loops occur during routing. It is very hard to ensure loop-free for vehicular routing.

4. Performance Evaluations

We evaluate the performance of our proposed protocol (GPSR-DTN) and compare it with GPSR and its variant via simulations conducted in NS-2 [28]. Since GPSR is not proposed for sparsely connected networks, to be fair, we extend GPSR by allowing that each node has buffers to hold packet. In this way, GPSR-B (GPSR with buffer) can be considered as combining a simple carry and forward protocol with GPSR but without any loop prevention. Notice that we do not compare our proposed protocol with any classical DTN routing in our simulations. However, [25] did provide some performance comparisons between GPSR-B and existing DTN routing solutions (such as epidemic routing [15]).

4.1. Simulation Environment. We implement the proposed routing scheme GPSR-DTN and GPSR-B in NS-2 [28] and use MOVE (mobility model generator for vehicular networks) [29] to generate realistic mobility model for VSNs. The street is designed in both directions, and traffic lights are deployed at each intersection. The distance between two adjacent traffic lights can have a significant effect on the network connectivity. Specifically, the network can be "fragmented" by the traffic lights when the radio transmission range is smaller than the distance between two adjacent clusters. In order to evaluate the proposed method, we test it in two types of network settings: *almost connected* and *intermittently connected*. In the intermittently one, the network is interrupted periodically because of the traffic lights. All networks are deployed in a 2052 m × 2052 m square map. 100 vehicles are deployed to the street layout. Vehicles move between 0 and 20 m/s along the street. The communication range is set to 250 m, and the period of beacon message is 1 second. Ten vehicles are selected as data sources and keep sending sensing data with different intervals form 0.5 to 5 seconds. The destination of all data packets is a static sink at a predefined position. All experiment parameters are recorded in Table 1.

Table 1: Simulation setup.

Parameter	Value
Simulation area	2052 m × 2052 m
Number of vehicles	100
Vehicle speed	0–20 m/s
communication range	250 m
Number of sources	10
MAC protocol	802.11
CBR rate	1 packet per 0.5–5 s
Vehicle beacon interval	1 s

4.2. Simulation Metrics. In all experiments, we compare GPSR, GPSR-B (with buffer), and GPSR-DTN with the following routing metrics. If the destination receives multiple copies of the same packet, only the first arriving one counts towards statistics.

(i) *Delivery ratio*: the average percentage of successfully delivered packets from the sources to the destination.

(ii) *Average delay*: the average time duration of successfully delivered packets from the sources to the destination.

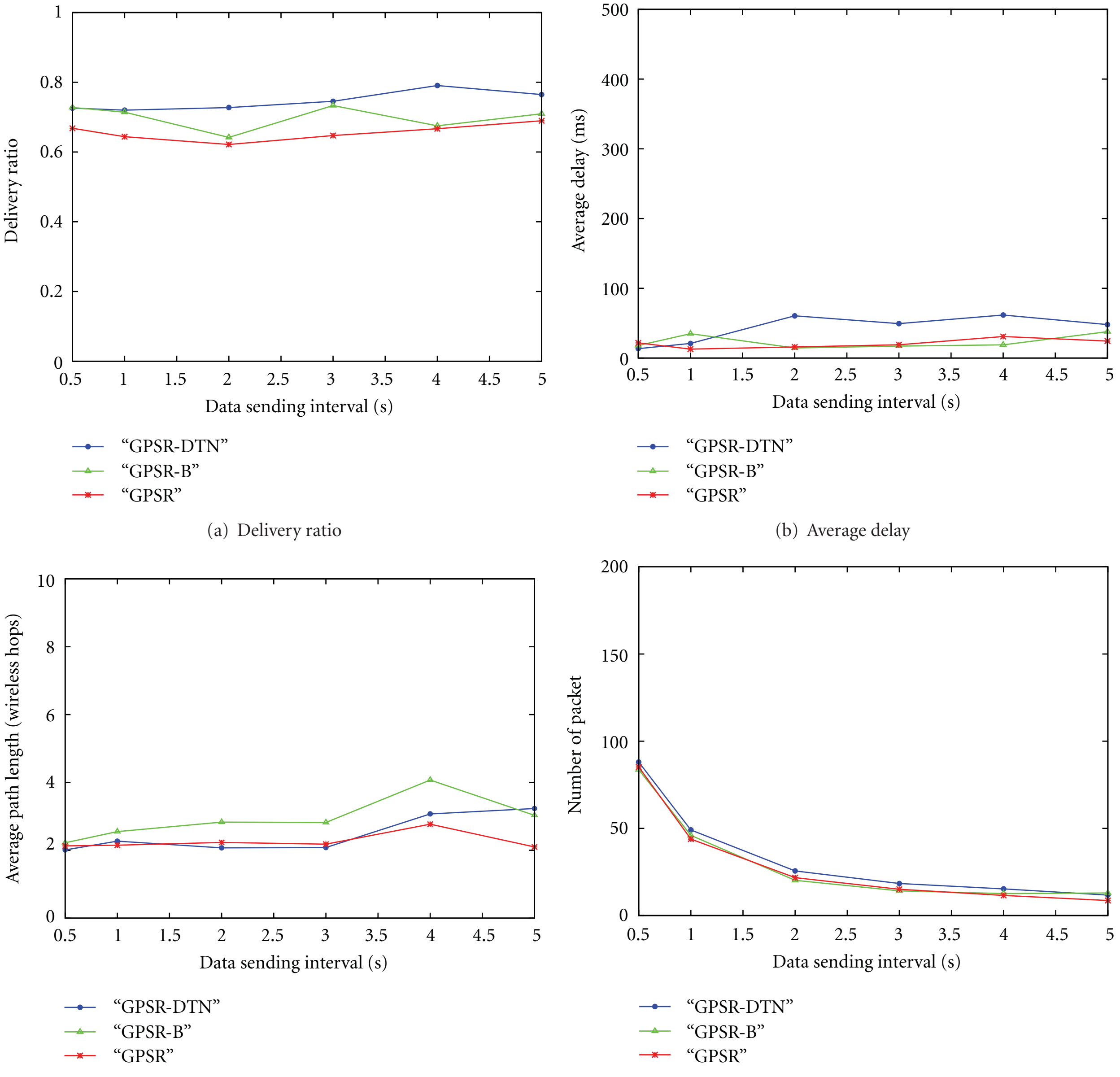
(iii) *Average path length*: the average number of intermediate vehicles of successfully delivered packets passing through from the sources to the destination.

(iv) *Number of packets*: the average number of copies of packets in the network at each second during the simulation.

4.3. Simulation Results. Figures 4 and 5 plot simulation results of our experiments for *almost connected* and *intermittently connected* scenarios, respectively.

As shown in Figures 4(a) and 5(a), GPSR has the lowest data delivery ratio; GPSR-DTN and GPSR-B have higher delivery ratio than GPSR does in both scenarios. This confirms that combining DTN routing strategies with position-based routing improves the chances of final delivery. Notice that GPSR-B may lead to more routing loops due to the lack of knowledge of moving directions; thus it has lower delivery ratio than GPSR-DTN. In GPSR-DTN, we carefully design the loop prevention mechanism based on the moving directions and whether two vehicles are on the same segment, so that the best vehicle is selected to hold or forward the packets. In addition, it is clear that all protocols have better performance under the almost connected scenario than under the intermittently connected scenario.

Figures 4(b) and 5(b) show the average delay of different protocols. It is obvious that all of three protocols have lower average delay for the almost connected scenario than for the intermittently connected scenario. Better connectivity provides better relay selection and thus leads to quicker transmissions. Both GPSR-DTN and GPSR-B usually have longer



(a) Delivery ratio

(b) Average delay

(c) Average path length

(d) Number of packets

Figure 4: Simulation results for almost connected scenario.

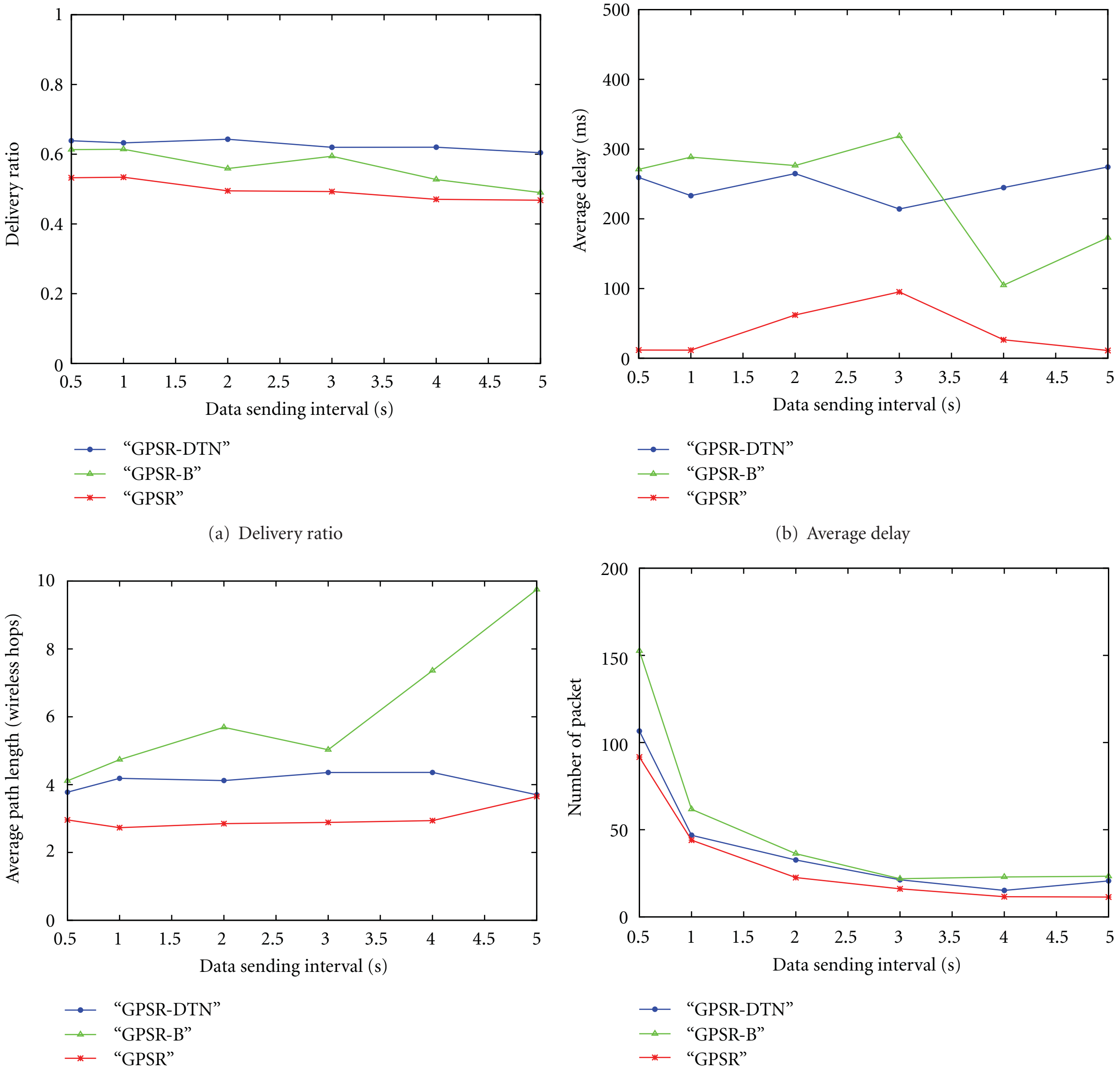
delay than GPSR has, since they both apply store-carry-forward strategy which causes longer delay. For the same reason, the average path lengths of GPSR-DTN and GPSR-B are usually longer than those of GPSR, as shown in Figures 4(c) and 5(c). Note that GPSR-B has longer average path length than GPSR-DTN due to possible routing loops. Even though with longer delay and path length, GPSR-DTN/GPSR-B indeed improve the delivery ratio.

Finally, GPSR-DTN and GPSR-B will cause more packets in the network since both will create and hold new copies of the original packet. This can be verified by Figures 4(d) and 5(d). However, such increases are limited and acceptable as the cost of improvement of delivery ratios.

In summary, even though with longer delay and path length, the proposed GPSR-DTN indeed improves the delivery ratio, especially in intermittently connected networks.

5. Conclusion

Traffic lights, accidents, or low density may lead to intermittent connectivity very common in vehicular networks. While traditional position-based VANET/VSN routing protocols are not suitable for sparsely connected vehicular sensor networks, in this paper, we propose a new hybrid forwarding protocol which combines position-based forwarding with the idea of store-carry-forward from DTNs. In the proposed



(a) Delivery ratio

(b) Average delay

(c) Average path length

(d) Number of packets

Figure 5: Simulation results for intermittently connected scenario.

method, driving directions of vehicles are used to make forwarding or carrying decisions, and a carefully designed loop prevention mechanism is also introduced. Experimental results show that the proposed hybrid method outperforms existing position-based solutions. In the future, we plan to test the proposed algorithm in more realistic scenarios (such as including transmission failures or unexpected vehicle mobility) and extend the proposed method to consider mobile destinations.

Acknowledgment

The work of F. Li is partially supported by the National Natural Science Foundation of China (NSFC) under Grant 60903151, the Beijing Natural Science Foundation under Grant 4122070 and SRF for ROCS, SEM. The work of X. Fan is supported in part by Projects of Major International (Regional) Joint Research Program from the NSFC under Grant 61120106010. The work of Y. Wang is supported in part by the US National Science Foundation under Grants CNS-0915331 and CNS-1050398.

References

[1] M. Nekovee, "Sensor networks on the road: the promises and challenges of vehicular ad hoc networks and vehicular grids," in *Proceedings of the Workshop on Ubiquitous Computing and e-Research*, May 2005.

[2] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 52–57, 2006.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 29–39, ACM, New York, NY, USA, June 2008.

[4] Y. Wang and Li. Fan, "Vehicular ad hoc networks," in *Guide to Wireless Ad Hoc Networks*, S. Misra, I. Woungang, and S. C. Misra, Eds., Springer, Berlin, Germany, 2009.

[5] "IEEE standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6: wireless access in vehicular environments," IEEE Standard 802.11p-2010, pp. 1-51.

[6] F. Bai, H. Krishnan, V. Sadekar, G. Holland, and T. Elbatt, "Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective," in *Proceedings of the IEEE Workshop on Automotive Networking*, December 2006.

[7] H. Gharavi, "Control based mobile ad-hoc networks for video communications," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 383–391, 2006.

[8] R. A. Santos, A. Edwards, R. Edwards, and L. Seed, "Performance evaluation of routing protocols in vehicular ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, no. 1-2, pp. 80–91, 2005.

[9] V. Namboodiri, M. Agarwal, and L. Gao, "A study on the feasibility of mobile gateways for vehicular ad-hoc networks," in *Proceedings of the 1st International Workshop on Vehicular Ad Hoc Networks (VANET '04)*, October 2004.

[10] G. Liu, B.-S. Lee, B.-C. Seet, C. H. Foh, K. J. Wong, and K.-K. Lee, "A routing strategy for metropolis vehicular communications," in *Proceedings of the International Conference on Information Networking (ICOIN '04)*, pp. 134–143, 2004.

[11] S. Y. Wang, C. C. Lin, Y. W. Hwang, K. C. Tao, and C. L. Chou, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 161–166, September 2005.

[12] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[13] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[14] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[15] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-200006, Duke University, 2000.

[16] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned Ad-Hoc networks," in *Proceedings of the International Symposium on Wearable Computers*, October 2001.

[17] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *ACM SIGPLAN Notices*, vol. 37, no. 10, pp. 96–107, 2002.

[18] H. Füßler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "Location-based routing for vehicular ad-hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review (MC^2R)*, vol. 7, no. 1, pp. 47–49, 2003.

[19] C. Lochert, H. Hartenstein, J. Tian, D. Herrmann, H. Füßler, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '03)*, pp. 156–161, Columbus, Ohio, USA, June 2003.

[20] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review (MC^2R)*, vol. 9, no. 1, pp. 69–72, 2005.

[21] V. Dumitrescu and J. Guo, "Context assisted routing protocols for inter-vehicle wireless communication," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 594–600, Dearborn, Mich, USA, June 2005.

[22] H. Gao, S. Utecht, G. Patrick et al., "High speed data routing in vehicular sensor networks," *Journal of Communications*, vol. 5, no. 3, pp. 181–188, 2010.

[23] Y. Shao, C. Liu, and J. Wu, "Delay tolerant networks in VANETs," in *Handbook on Vehicular Networks*, M. Weigle and S. Olariu, Eds., Taylor & Francis, 2009.

[24] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of the IEEE Vehicular Technology Conference*, 2005.

[25] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[26] I. Leontiadis and C. Mascolo, "GeOpps: geographical opportunistic routing for vehicular networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM' 07)*, pp. 1–6, June 2007.

[27] P.-C. Cheng, K.-C. Lee, M. Gerla, and J. Härri, "GeoDTN+ Nav: geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[28] The Network Simulator (NS2), http://www.isi.edu/nsnam/ns/.

[29] F. K. Karnadi, Z. H. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for VANET," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, March 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 106439, 15 pages
doi:10.1155/2012/106439

Research Article

An Energy-Efficient CKN Algorithm for Duty-Cycled Wireless Sensor Networks

Lei Wang,[1] Zhuxiu Yuan,[1] Lei Shu,[2] Liang Shi,[3] and Zhenquan Qin[1]

[1] *School of Software, Dalian University of Technology, Dalian 116621, China*
[2] *Department Multimedia Engineering, Osaka University, Osaka 565-0871, Japan*
[3] *School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*

Correspondence should be addressed to Lei Shu, lei.shu@ieee.org

Received 21 December 2011; Revised 14 March 2012; Accepted 15 March 2012

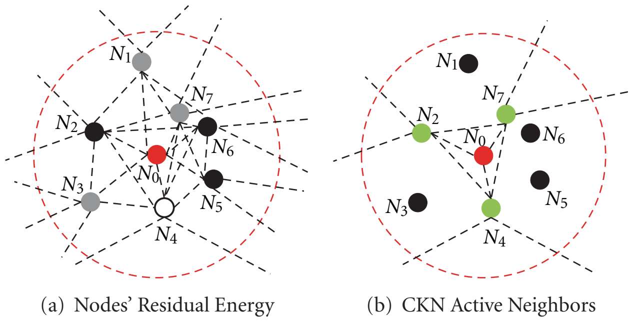
Academic Editor: Yunhao Liu

To prolong the lifetime of a wireless sensor network, one common approach is to dynamically schedule sensors' active/sleep cycles (i.e., duty cycles) using sleep scheduling algorithms. The connected K-neighborhood (CKN) algorithm is an efficient decentralized sleep scheduling algorithm for reducing the number of awake nodes while maintaining both network connectivity and an on-demand routing latency. In this paper, we investigate the unexplored energy consumption of the CKN algorithm by building a probabilistic node sleep model, which computes the probability that a random node goes to sleep. Based on this probabilistic model, we obtain a lower epoch bound that keeps the network more energy efficient with longer lifetime when it runs the CKN algorithm than it does not. Furthermore, we propose a new sleep scheduling algorithm, namely, Energy-consumption-based CKN (ECCKN), to prolong the network lifetime. The algorithm EC-CKN, which takes the nodes' residual energy information as the parameter to decide whether a node to be active or sleep, not only can achieve the k-connected neighborhoods problem, but also can assure the k-awake neighbor nodes have more residual energy than other neighbor nodes in current epoch.

1. Introduction

Wireless sensor networks (WSNs) are normally powered by batteries with limited energy, which are difficult or impossible to be recharged or replaced. A common approach for saving the sensor nodes' energy is to select a subset of nodes to remain active/awake and let others go to sleep in a given epoch. Most of current literatures on sleep scheduling in WSNs are to achieve *point coverage* and/or *node coverage* problems [1]. *Point coverage* problem (also called *spatial coverage*) focuses on selecting a set of active nodes in an epoch so that every point of the deployment space is covered, while considering some optimization goals, for example, minimizing energy consumption [2], minimizing average event detection latency [3]. *Node coverage* problem (also called *network coverage*) focuses on choosing a set of active nodes, in which (1) they construct a connected backbone and (2) sleeping nodes are direct neighbors of at least one active node [4]. This node coverage problem is to ensure that any two nodes in the network can communicate with each other through the connected backbone.

The Connected K-Neighborhood (CKN) algorithm is a distributed sleep scheduling algorithm [1], which can reduce the number of active nodes efficiently. It keeps the network k-connected and optimizes the geographic routing performance. Supporting the geographic routing performance is not studied in any previous *point coverage* and *node coverage* researches. Although, the CKN algorithm performs well with the geographic routing protocols, the following questions are not addressed in paper [1]. (1) *How frequently should the CKN algorithm be executed in the network so that it can really help to save energy, for each time executing the CKN algorithm also consumes energy?* Intuitively, executing the CKN algorithm will consume a mass of energy with substantial data transmission to exchange local information between nodes and their neighbors, which influences the energy consumption distribution of network. (2) *Do all active sensor nodes in the CKN algorithm* [1] *consume the*



(a) Nodes' Residual Energy (b) CKN Active Neighbors

Figure 1: Nonuniform energy consumption problem. In (a), the black nodes represent the nodes which have more residual energy than those of node N_0, the grey and white nodes have less energy than N_0's, and the white node has the least energy. In (b), the green nodes represent the active nodes according to the CKN algorithm in a WSN, and the sensor nodes with less residual energy are selected.

energy uniformly in each epoch? We identify the *nonuniform energy consumption* problem, as shown in Figure 1. It is recognized that when executing the CKN algorithm in WSNs, sensor nodes with less residual energy are possible to be selected, which may result in that the energy of these sensor nodes can be fast consumed. The problem is caused by: *the CKN algorithm chooses the active nodes absolutely based on the* ranks *that are randomly given at the beginning of executing the CKN algorithm in each epoch*. In other words, the CKN algorithm cannot ensure the network energy is balancedly consumed.

Motivated by above two major issues, we conduct theoretical studies on two important questions based on the CKN algorithm. The first question is as follows. *Is the CKN algorithm energy saving for any given value of k and the epoch? If not, how frequently should the CKN algorithm be executed so that the network is energy saving?* In order to find out the relationship between the epoch and the energy consumption, we build a probabilistic model for the CKN algorithm to compute the probability that each random node goes to sleep and the expected total number of epochs during each node's lifetime. We formulate the lower bound of an epoch to keep the CKN algorithm energy efficient.

We address the second question based on the analysis for the first problem: *How do we design a new sleep scheduling algorithm based on the CKN algorithm that can balance the energy consumption to prolong network lifetime further?* Satisfying all those requirements that the CKN algorithm holds, a new decentralized sleep scheduling algorithm is challenging. In the light of the discussions for the question 1, we propose a new sleep scheduling algorithm, named energy-consumption-based CKN (EC-CKN), to prolong the network lifetime. The advantage of the EC-CKN algorithm over the original CKN algorithm is that it takes the nodes' residual energy information as parameter to decide whether a node to be active or sleep. The EC-CKN algorithm inherits all the major properties of the CKN algorithm, that is, solving the k-connected neighborhoods problem. Meanwhile, it also makes a significant new contribution to the energy efficiency by assuring the k-active neighbor nodes have more residual energy than other neighbor nodes in the current epoch. A theoretical analysis on the energy consumption of the EC-CKN algorithm is given to show the correctness of the new contribution.

The rest of the paper is as follows. Section 2 shows the network model. Section 3 presents the original CKN algorithm regulation and its properties. Section 4 builds a probabilistic model to compute the probability that a random node goes to sleep. Section 5 presents the EC-CKN algorithm. Section 6 demonstrates the properties of the EC-CKN algorithm. Section 7 shows the simulation results about the original CKN algorithm and the EC-CKN algorithm, comparing theoretical values and simulation results. Finally, Section 9 concludes the paper.

2. Network Model

2.1. Communication Network Model. A multihop sensor network is modeled by a graph $G = (S, E)$, where $S = \{s_1, s_2, \ldots, s_n\}$ is the set of sensor nodes and E is the set of directed links. Each node has a uniform transmission radius of r_t, and the necessary condition of $(s_i, s_j) \in E$ is $|s_i - s_j| \leq r_t$ and a node s_j is the next hop of s_i to the sink by the routing protocol. If $(s_i, s_j) \in E$, we use $l_{i,j}$ to denote (s_i, s_j). Each node also has a uniform interference radius of r_f. An node s_j is interfered by the signal from s_i, if $|s_i - s_j| \leq r_f$ and s_j is not the intended receiver. Let I_i be the interference region that centers at s_i with the interference radius r_f. Each node is only equipped with a single radio interface and has the uniform initial energy $\mathcal{E}_0$. The entire network lifetime is divided into epochs, and each epoch is T. At the beginning of each epoch, a node transmits packets in T_1, and then it runs the sleep scheduling algorithm to decide the state of the next epoch in T_2 (where $T = T_1 + T_2$) as shown in Figure 2.

2.2. Event Generation Model. Assume each node has a uniform sensing radius r_s. Let C_u denote the sensing region of the node s_u, which centers at s_u with the sensing radius r_s. An event occurs when the sensing unit of a node s_u picks up a signal with the power above a predetermined threshold within the sensing region C_u [5]. Suppose the

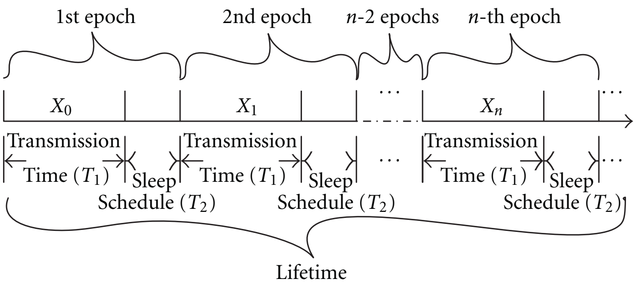


Figure 2: The lifetime of a node s_u consists of many epochs. Each epoch includes packet transmission time and sleep scheduling algorithm's execution time.

temporal event behavior over the entire sensing region, $\mathcal{A}$, is a Poisson process with an average event rate λ. Let $p_{XY}(x, y)$ denote an independent probability distribution of the spatial distribution of events. Let p_e denote the probability that an event is detected by a node s_u, given the fact that it occurred in $\mathcal{A}$:

$$p_e = \frac{\int_{C_u} p_{XY}(x, y)\,dx\,dy}{\int_{\mathcal{A}} p_{XY}(x, y)\,dx\,dy}, \tag{1}$$

where $p_{XY}(x, y)$ is the spatial distribution of events that is characterized by an independent probability distribution. Let $p_m(T, n)$ denote the probability that n events occur in an epoch T at a node s_u. Therefore, the probability of no events occurring in C_u over an epoch T is given by

$$\begin{aligned} p_m(T, 0) &= \sum_{i=0}^{\infty} \frac{e^{-\lambda T (\lambda T)^i}}{i!} (1 - p_e)^i \\ &= e^{-p_e \lambda T}. \end{aligned} \tag{2}$$

Let $p_m(T)$ denote the probability that at least one event occurs in an epoch T at a node s_u:

$$p_m(T) = 1 - p_m(T, 0) = 1 - e^{-p_e \lambda T}. \tag{3}$$

That is the probability of at least one event occurring at the node s_u is an exponential distribution characterized by a spatially weighted event arrival rate $\lambda_{u,u} = \lambda \times p_e$.

2.3. Buffer Analysis. Now, we consider two sources of traffic as an input to the buffer of each node [6].

Generated Packets. The sensing unit of a node senses events and generates packets as discussed in Section 2.2. These packets are *generated packets*. For a node s_u, the rate of the generated packets is denoted by $\lambda_{u,u}$.

Relay Packets. A node also receives packets from its upstream nodes and then forwards them to the sink node. (Along the data stream from a source node to the sink node by the routing protocol, downstream nodes are closer to the sink node, and receive packets sent by the node. Upstream nodes are far away from the sink node, and transmit packets to the node). These packets are referred as *relay packets*. The rate at which a node s_u receives relay packets from a node s_v is denoted as $\lambda_{v,u}$.

Therefore, the input packet rate of s_u's buffer, λ_u, can be written as

$$\lambda_u = \lambda_{u,u} + \lambda_{u,r} = \lambda_{u,u} + \sum_{v \in \mathcal{N}_u^{\text{in}}} \lambda_{v,u}, \tag{4}$$

where $\lambda_{u,r}$ is the total relay packet rate at the node s_u, $\mathcal{N}_u^{\text{in}}$ is the set of nodes that have the node s_u as the next hop, and $\lambda_{v,u}$ is the packet rate from the node s_v to the node s_u. Let γ_u be the output rate of a node, which is given by

$$\gamma_u = (1 + e_u)\lambda_u, \tag{5}$$

where e_i is the packet error rate.

2.4. Channel and Energy Consumption Model. The energy consumption model characterizes energy consumption of a node in the network. Suppose there is no energy consumption when a node is sleep. If a node is active, we classify the energy consumption into three general categories.

(1) The Constant Energy Consumption. is the minimum energy needed to sustain a node when it is active without the packet transmission. It includes, for example, the battery leakage, energy consumed during the state transformation.

(2) The Additional Energy Consumption. is the energy consumed by the data transmission during the sleep scheduling algorithm running time.

(3) The Conventional Energy Consumption. includes the receiving energy consumption and the transmitting energy consumption except the local information exchange in the sleep scheduling algorithm, which is based on the first-order radio model [7].

The energy loss is due to the channel transmission, ϵ_{amp} is the transmit amplifier. And the transmitting energy consumption for a bit packet is

$$E_{tx} = \mathcal{E} + \epsilon_{\text{amp}} \cdot r_t^2, \tag{6}$$

and the receiving energy consumption is

$$E_{rx} = \mathcal{E}, \tag{7}$$

where $\mathcal{E}$ is energy consumed by the transmitter or receiver circuitry.

2.5. Lifetime Definition. There is no universally agreed definition of network lifetime as it depends on the specific application. The lifetime can be measured by the time when the first node exhausts its energy, or when a certain fraction of nodes is dead, or even when all nodes are dead. Alternately, it may be reasonable to measure the network lifetime by application-specific parameters, such as the time when the network can no longer relay sensory data packets. In this paper, we define the network lifetime is the time when the first sensor node run out its energy from the beginning. The general network lifetime is the exact individual lifetime of each active node [8].

Theorem 1. *For a sensor network, each node has nonrechargeable initial energy $\mathcal{E}_0$, the average general network lifetime $\mathbb{E}[\mathcal{L}]$, is given by*

$$\mathbb{E}[\mathcal{L}] = \frac{\mathcal{E}_0}{E_c + \lambda_r \mathbb{E}[E_{rx}] + \gamma \mathbb{E}[E_{tx}]}, \tag{8}$$

where E_c is the constant energy consumption on the first died node, $\mathbb{E}[E_{rx}]$ is the expected receiving energy consumption, and $\mathbb{E}[E_{tx}]$ is the expected transmitting energy consumption.

Proof. Suppose there are M independently and identically distributed trials on the same sensor network to record the network lifetime $\mathcal{L}$, the receiving energy consumption of each bit E_{rx}, and the transmitting energy consumption of each bit E_{tx}. For the mth trial ($1 \le m \le M$), the total energy consumed by the first died node during the whole lifetime is

$$\mathcal{E}_0 = E_c \mathcal{L}^m + \sum_{i=1}^{N_{rx}^m} E_{rx}^m(i) + \sum_{i=1}^{N_{tx}^m} E_{tx}^m(i), \tag{9}$$

where N_{rx}^m is the number of bits to be received, and N_{tx}^m is the number of bits to be transmitted of the first died node during the network lifetime of the mth trial. Summing (9) up over the M trials and dividing both sides by M, we obtain

$$\begin{aligned}\mathcal{E}_0 = & \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}^m \Bigg[E_c \\ & + \left(\frac{\sum_{m=1}^{M} N_{rx}^m}{\sum_{m=1}^{M} \mathcal{L}^m}\right) \times \left(\frac{\sum_{m=1}^{M}\sum_{i=1}^{rx} E_{rx}^m(i)}{\sum_{m=1}^{M} N_{rx}^m}\right) \\ & + \left(\frac{\sum_{m=1}^{M} N_{tx}^m}{\sum_{m=1}^{M} \mathcal{L}^m}\right) \times \left(\frac{\sum_{m=1}^{M}\sum_{i=1}^{N_{tx}^m} E_{tx}^m(i)}{\sum_{m=1}^{M} N_{tx}^m}\right) \Bigg].\end{aligned} \tag{10}$$

Note that $\lim_{M \to \infty}(\sum_{m=1}^{M} N_{rx}^m)/(\sum_{m=1}^{M} \mathcal{L}^m) = \lambda_r$ is the average receiving rate and $\lim_{M \to \infty}(\sum_{m=1}^{M} N_{tx}^m)/(\sum_{m=1}^{M} \mathcal{L}^m) = \gamma$ is the average transmitting rate.

The average receiving energy consumed in the ith received bit can be written as

$$\mathbb{E}[E_{rx}] = \lim_{M \to \infty} \frac{\sum_{m=1}^{M} E_{rx}^m(i)\mathcal{X}_m(i)}{D_{rx}(i)}, \quad 1 \le i \le N_{rx}, \tag{11}$$

where $\mathcal{X}_m(i) = 1$ for $1 \le i \le N_{rx}^m$ and 0 otherwise. $D_{rx}(i) = \sum_{m=1}^{M} \mathcal{X}_m(i)$ is the total number of the occurrence of the ith received bit among the M trials, and $N_{rx} = \max_m\{N_{rx}^m\}$ is the maximum number of received bits during the network lifetime. The probability that the received bit chosen randomly happens to the ith received bit is given by

$$p_{rx}(i) = \lim_{M \to \infty} \frac{D_{rx}(i)}{\sum_{m=1}^{M} N_{rx}^m}, \quad 1 \le i \le N_{rx}. \tag{12}$$

Averaging (11) over the received bit chosen randomly indexing i, the expected receiving energy consumption is defined as

$$\begin{aligned}\mathbb{E}[E_{rx}] &\triangleq \mathbb{E}_{rx}^{i}\{\mathbb{E}[E_{rx}(i)]\} \\ &= \lim_{M \to \infty} \frac{\sum_{m=1}^{M}\sum_{i=1}^{N^m} E_{rx}^m(i)}{\sum_{m=1}^{M} N_{rx}^m},\end{aligned} \tag{13}$$

where $\mathbb{E}[E_{rx}(i)]$ is the average energy consumed in ith bit packet, $\mathbb{E}_{rx}^{i}\{\cdot\}$ denotes the expectation over the randomly chosen received packet indexing i.

Similarly, the expected transmitting energy consumption is

$$\begin{aligned}\mathbb{E}[E_{tx}] &\triangleq \mathbb{E}_{tx}^{i}\{\mathbb{E}[E_{tx}(i)]\} \\ &= \lim_{M \to \infty} \frac{\sum_{m=1}^{M}\sum_{i=1}^{N_{tx}^m} E_{tx}^m(i)}{\sum_{m=1}^{M} N_{tx}^m},\end{aligned} \tag{14}$$

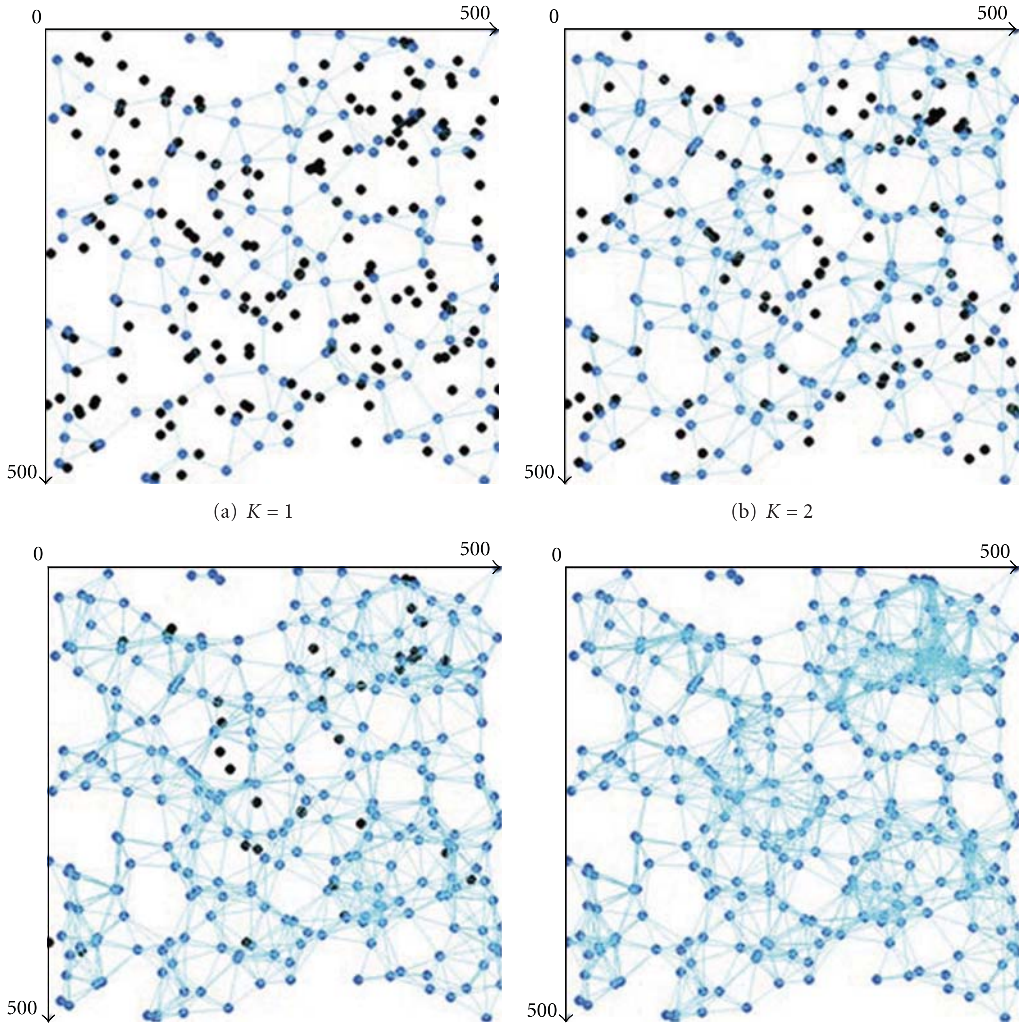
where $\mathbb{E}[E_{tx}(i)]$ is the average energy consumed in i th transmitted packet, $\mathbb{E}_{tx}^{(i)}\{\cdot\}$ denotes the expectation over the randomly chosen transmitted packet indexing i. □

3. A Brief Description of CKN

In [1], the studied WSN is represented as an undirected communication graph $G = (S, E)$. N_u is the set of s_u's neighbors. The connected K-neighborhood problem is defined as (i) each node has at least $\min\{k, |N_u|\}$ active neighbors, which can be called awake neighbors; (ii) all active nodes are connected. To solve the problem, the authors developed a sleep scheduling algorithm: connected K-neighborhood (CKN).

In CKN, each node s_u picks a random rank rank_u, broadcasts the rank_u, and collects its neighbors' ranks in R_u. And then, s_u broadcasts R_u and collects R_v from its neighbors, where $s_v \in N_u$. If s_u or its neighbors has less than k neighbors, s_u will remain awake. Otherwise, s_u computes a subset C_u of N_u that is a set of nodes having rank $< \text{rank}_u$. "Before the node s_u goes to sleep it needs to make sure that all nodes in C_u are connected by nodes with rank $< \text{rank}_u$ and each of its neighbors has at least k neighbors from C_u" [1].

The CKN algorithm has the following properties: first, each node s_u (awake or not) with $|N_u|$ neighbors must have at least $\min\{k, |N_u|\}$ awake neighbors in each epoch; second, there is the minimal average number of awake nodes per epoch; finally, awake nodes change from epoch to epoch. For any $k \ge 1$, there are more than $4(k + \ln N)$ neighbors for each node, where N is the number of nodes in the network. Then,



(a) $K = 1$

(b) $K = 2$

(c) $K = 4$

(d) $K = 8$

Figure 3: Four examples of executing the CKN algorithm with different values for k. When $k = 1$, a large number of sensor nodes can turn to sleep mode, but, when $k = 8$, almost all sensor nodes have to be always-on. Here, black and unconnected sensor nodes are sleeping nodes.

with high probability, $|\mathrm{CKN}_k| = O(\ln N) \cdot |\mathrm{OPT}_k|$, where $|\mathrm{CKN}_k|$ is the number of awake nodes by the CKN algorithm and $|\mathrm{OPT}_k|$ is the number of awake nodes by an optimal algorithm that finds a minimum connected k-neighborhood as Figure 3.

4. Analysis of the CKN Algorithm

In this section, we build up a probabilistic model for the CKN algorithm to compute the probability that one node s_u goes to sleep in each epoch. Based on the probability, we formulate the lower bound of an epoch to keep the CKN algorithm energy efficient.

Notations used in this section. N_u is the set of s_u's neighbors, and N'_u is the set of s_u's 2-hop neighbors. C_u and C'_u are the subsets of N_u and N'_u having rank $\leq \mathrm{rank}_u$. $|N_u|$, $|N'_u|$, $|C_u|$, and $|C'_u|$ are the number of the elements in N_u, N'_u, C_u, and C'_u, respectively. Graphs G_{C_u} and $G_{C'_u}$ are composed of nodes and potential links in C_u and C'_u.

For a homogeneous Poisson point process in two dimensions, the probability that a random node has n neighbors is [9]

$$P(|N_u| = n) = \frac{(\rho\pi r_t^2)^n}{n!} \cdot e^{-\rho\pi r_t^2}, \tag{15}$$

where $\rho = N/\mathcal{A}$ is the nodes density. And a node is isolated with a probability of $P(|N_u| = 0) = e^{-\rho\pi r_t^2}$. The expectation of the number of s_u's neighbors is

$$\mathbb{E}[|N_u|] = \rho\pi r_t^2. \tag{16}$$

If there are at least k different paths connecting any two different vertices in the graph G, the graph is *k-connected*

($k = 1, 2, \ldots$). The probability that the graph G is k-connected is

$$P\left(G^k\right) = \left(1 - \sum_{n}^{k-1} \frac{\left(\rho\pi r_t{}^2\right)^n}{n!} \cdot e^{-\rho\pi r_t{}^2}\right)^N. \tag{17}$$

4.1. Sleep Probability in the CKN Algorithm. For the CKN algorithm, when a node s_u has at least k neighbors, whether it goes to sleep is decided by two factors: (1) "any two nodes in C_u are connected either directly or indirectly through s_u's 2-hop neighbors that have rank less than rank_u" and (2) "any node in N_u has at least k neighbors for C_u" [1].

Lemma 2. *$|C_u| \sim B(|N_u|, p)$ and $|C'_u| \sim B(|N'_u|, p)$ are binomial distributions, where p is the probability that a node has* rank < rank_u.

Proof. Let $\{\text{rank}_1, \text{rank}_2, \ldots, \text{rank}_{|N_u|}\}$ be the random ranks for nodes in N_u. Suppose $\text{rank}_i \in (0, 1)$ and $i \in [1, |N_u|]$. Let

$$\begin{gathered} x_i = \begin{cases} 1, & \text{if } \text{rank}_u - \text{rank}_i > 0, \\ 0, & \text{otherwise}, \end{cases} \\ p = p_i(x_i = 1) = E\left(P_{\text{rank}_u - \text{rank}_i > 0}\right) = \frac{1}{2}, \\ q = p_i(x_i = 0) = 1 - p. \end{gathered} \tag{18}$$

Let $Z = \sum_{i \in N_u} x_i$, and the probability $z_j = j$ is

$$\begin{aligned} p_j &= \binom{|N_u|}{j} \cdot p^j \cdot q^{|N_u| - j} \\ &= \binom{|N_u|}{j} \cdot p^{|N_u|}, \end{aligned} \tag{19}$$

where $j \in [0, |N_u|]$ and $|C_u| \sim B(|N_u|, p)$. Similarly, $|C'_u| \sim B(|N'_u|, p)$. □

Theorem 3. *Under the CKN algorithm, the sleep probability of a node s_u is*

$$p_s(|C_u|) = \text{Prob}_1 \cdot \text{Prob}_2, \tag{20}$$

and the awake probability is

$$p_a(|C_u|) = 1 - p_s(|C_u|), \tag{21}$$

where Prob_1, *the probability a node s_u satisfies the first condition, is defined as*

$$\text{Prob}_1 \triangleq \left(1 - e^{-\rho_1\pi r_t{}^2}\right)^{|C'_u|}, \tag{22}$$

and Prob_2, *the probability s_u satisfies the second condition, is defined as*

$$\text{Prob}_2 \triangleq P(|C_u| \geq k + 1) \cdot P\left(G^k_{C_u}\right) \cdot P\left(C^{\,k}_{uN_u - C_u}\right). \tag{23}$$

Proof. If s_u and its neighbors have at least k neighbors, the two conditions deciding s_u whether to sleep or not can be interpret as the following corresponding conditions: (1) the graph $G_{C'_u}$ is connected, and (2) the graph G_{C_u} is k-connected and each node in the set $N_u - C_u$ has at least k neighbors in C_u.

The probability that the graph $G_{C'_u}$ is k-connected is

$$\begin{aligned} \text{Prob}_1 &\leq \text{Prob}\left(G_{C'_u} \text{ is connected}\right) \\ &\leq P(|N_u|_{\min} \geq 1) \\ &= \left(1 - e^{-\rho_1\pi r_t{}^2}\right)^{|C'_u|}, \end{aligned} \tag{24}$$

where $|N_u|_{\min}$ is the minimum degree in the graph $G_{C'_u}$, and $\rho_1 = |C'_u|/(4\pi t_r{}^2)$ is the node density in the graph $G_{C'_u}$.

The probability of the condition (2) is

$$\text{Prob}_2 = P(|C_u| \geq k + 1) \cdot P\left(G^k_{C_u}\right) \cdot P\left(C^{\,k}_{uN_u - C_u}\right), \tag{25}$$

where $P(G^k_{C_u})$, the probability that the graph G_{C_u} is k-connected, can be expressed as

$$P\left(G^k_{C_u}\right) \leq \left(1 - \sum_{n=0}^{k-1} \frac{\left(\rho_2\pi r_t{}^2\right)^n}{n!} \cdot e^{-\rho_2\pi r_t^2}\right)^{|C_u|}, \tag{26}$$

and $P(C^{\,k}_{uN_u - C_u})$ is the probability that a node $s_v \in (N_u - C_u)$ has at least k neighbors in C_u, which is

$$P\left(C^{\,k}_{uN_u - C_u}\right) = \left(1 - \sum_{n=0}^{k-1} \frac{\left(\rho_2\pi r_t{}^2\right)^n}{n!} \cdot e^{-\rho_3\pi r_t{}^2}\right)^{|N_u - C_u|}, \tag{27}$$

where $\rho_2 = |C_u|/(\pi t_r{}^2)$, $\rho_3 = (|C_u| + 1)/(\pi t_r{}^2)$, and $|N_u - C_u|$ is the number of the elements in the set $N_u - C_u$. □

4.2. Energy Consumption of the CKN Algorithm. Based on the result of the probability a random node goes to sleep, we can now analyze the node energy consumption for two cases: (1) it runs the CKN algorithm; (2) it does not run the CKN algorithm.

Lemma 4. *When a node s_u executes the CKN algorithm, its energy consumption is*

$$\mathcal{E}_{\text{ckn}}(s_u) = \mathcal{E}^1_c + 2(E_{tx} + |N_u| \cdot E_{rx}), \tag{28}$$

and the energy consumption of a node during each epoch is

$$\mathcal{E}_{\text{epoch}}(s_u) = \mathcal{E}^2_c + T_1(\lambda_r E_{rx} + \gamma E_{tx}), \tag{29}$$

where $\mathcal{E}^1_c$ and $\mathcal{E}^2_c$ are the constant energy consumptions of a node during the time of the CKN algorithm executed and an epoch, respectively.

Theorem 5. *Under the CKN algorithm, the lower bound of an epoch keeping the network energy efficient is*

$$\underline{T} \geq \frac{\mathbb{E}[\mathcal{L}]\left(\mathbb{E}[\mathcal{E}_{\text{ckn}}] + p_a \cdot \mathbb{E}\left[\mathcal{E}_{\text{epoch}}\right]\right)}{\mathcal{E}_0}, \tag{30}$$

where $\mathbb{E}[\mathcal{E}_{\text{ckn}}]$ and $\mathbb{E}[\mathcal{E}_{\text{epoch}}]$ are the expectation energy consumption during the time of the CKN algorithm executed and the expectation of the energy consumption during an epoch.

```
Input: The least number of neighbors $k$
Output: Connected $k$-Neighborhood Network
    (1) Get the current residual energy $Erank_u$;
    (2) Broadcast $E\mathrm{rank_u}$ and receive the residual energy of its neighbors $N_u$. Let $R_u$ be the set of the residual energy of nodes in $N_u$.
    (3) Broadcast $R_u$ and receive $R_v$ from each node $s_v$ where $s_v \in N_u$.
    (4) If $|N_u| < k$ or $|N_v| < k$ for any $s_v \in N_v$, remain awake.
        Return.
    (5) Compute $E_u = \{s_v \mid s_v \in N_u$ and $E\mathrm{rank}_v > E\mathrm{rank}_u\}$;
    (6) Go to sleep if both the following conditions hold. Remain awake otherwise.
        (i) Any two nodes in $E_u$ are connected either directly or indirectly through nodes that are the $s_u$'s 2-hop neighbors that have $E\mathrm{rank}_v$ larger than $E\mathrm{rank}_u$;
        (ii) Any node in $N_u$ has at least $k$ neighbors from $E_u$.
    (7) Return.
```

Algorithm 1: Energy-consumption-based CKN (*run the following at each node s_u^*).

Proof. Suppose there are M *i.i.d.* trials on the same network that runs the CKN algorithm as the sleep schedule to record the network lifetime, $\mathcal{L}_{\text{ckn}}$, and the epoch is T in each trial. For the mth trial, the total energy consumed by the first died node during the lifetime is

$$\mathcal{E}_0 = \mathcal{N}^m_{\text{epoch}} \mathcal{E}_{\text{ckn}} + \sum_{i=1}^{\mathcal{N}^m_{\text{epoch}}} \mathcal{E}^m_{\text{epoch}}(i), \tag{31}$$

where $\mathcal{N}^m_{\text{epoch}}$ is the number of epochs in the mth trial. Summing (31) up to the M trials and dividing both sides by M, we obtain

$$\mathcal{E}_0 = \frac{1}{M}\left[\mathcal{E}_{\text{ckn}} \sum_{m=1}^{M} \mathcal{N}^m_{\text{epoch}} + \sum_{m=1}^{M} \sum_{i=1}^{\mathcal{N}^m_{\text{epoch}}} \mathcal{E}^m_{\text{epoch}}(i)\right]. \tag{32}$$

Note that $T\lim_{M\to\infty}(1/M)\sum_{m=1}^{M} \mathcal{N}^m_{\text{epoch}} = \mathbb{E}[\mathcal{L}_{\text{ckn}}]$.

The average energy consumed in the ith epoch can be written as

$$\mathbb{E}\left[\mathcal{E}_{\text{epoch}}\right] = \lim_{M\to\infty} \frac{\sum_{m=1}^{M} E^m_{\text{epoch}}(i)\mathcal{Y}_m(i)}{D_{\text{epoch}}(i)}, \quad 1 \le i \le \mathcal{N}_{\text{epoch}}, \tag{33}$$

where $\mathcal{Y}_m(i) = 1$ if $1 \le i \le \mathcal{N}^m_{\text{epoch}}$ and the node is awake in the ith epoch, and 0 otherwise. $D_{\text{epoch}}(i) = \sum_{m=1}^{M} \mathcal{Y}_m(i)$ is the total number of the ith epoch that the node is awake among the M trials, and $\mathcal{N}_{\text{epoch}} = \max_m\{\mathcal{N}^m_{\text{epoch}}\}$ is the maximum number of epochs during the network lifetime. The probability that a randomly chosen awake epoch of a node happens to the ith epoch is given by

$$p_a(i) = \lim_{M\to\infty} \frac{D_{\text{epoch}}(i)}{\sum_{m=1}^{M} \mathcal{N}^m_{\text{epoch}}} = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} p_a(m), \tag{34}$$

where $p_a(m)$ is the nodes awake probability in the mth trial. Averaging (33) over the randomly chosen epoch indexing i, the expected ith epoch energy consumption except for the energy consumed by the CKN algorithm is defined as

$$\mathbb{E}\left[\mathcal{E}_{\text{epoch}}\right] \triangleq \lim_{M\to\infty} \sum_{m=1}^{M} \frac{E^m_{\text{epoch}}(i)\mathcal{Y}_m(i)}{\mathcal{N}^m_{\text{epoch}}}. \tag{35}$$

□

5. The Energy-Consumption-Based CKN Algorithm

We develop a new sleep scheduling algorithm to extend the network lifetime, which can still have all properties of the CKN algorithm.

A scalable distributed solution to the connected k-neighborhoods problem based on the nodes' current residual energy information is challenging for several reasons. First, a node can go to sleep assuming that there are at least k neighbors being awake to keep it k-connected. Second, the outcome of the algorithm must change over epochs so that all nodes have opportunities to sleep. Third, even though nodes decide to sleep or wake up based on their local information, the whole network must be globally connected. The aforementioned three challenges have been achieved by the CKN algorithm [1], which keep the network duty-cycled and connected k-neighborhood. Fourth, awake neighbors of any node s_u have k-top residual. The last one makes sure the energy of the network consumed balancedly, which is the main strength that the EC-CKN algorithm has over the original CKN algorithm.

We address the challenges by proposing the EC-CKN algorithm. The pseudocode of Algorithm 1 depicts the EC-CKN algorithm, which is repeated in each epoch on each node. The algorithm takes an input parameter, k, the required minimum number of per node's awake neighbors. In EC-CKN algorithm, a node s_u broadcasts its current residual energy information $E\mathrm{rank}_u$ (Step 1). It computes a subset E_u of neighbors having $E\text{rank} < E\text{rank}_u$ (Step 5). Before s_u goes to sleep, it makes sure that any two nodes in E_u are connected either directly or indirectly through the node that is in the s_u's 2-hop neighbors having $E\text{rank} < E\text{rank}_u$, and its neighbors have at least k neighbors from E_u (Step 6). These requirements ensure that when a node has less than k neighbors, none of its neighbors goes to sleep, and when it has more than k neighbors, at least k neighbors decide to remain awake. Note that these requirements are easy to keep by computing locally with 2-hop neighborhood information. The current residual energy is exchanged in Steps 2 and 3.

6. Properties of the EC-CKN Algorithm

This section analyzes the network lifetime, the awake probability, and the energy consumption of the network under the EC-CKN algorithm.

Theorem 6. *For any $k \geq 1$, the average network lifetime of the EC-CKN algorithm increases with the increases of the ratio of the network size N and k, N/k.*

Proof. Suppose N nodes are placed uniformly at random within a deployment area such that the average number of neighbors per node is $\xi \geq 4(k + \ln N)$. Let $\delta = (ck \ln N)/\xi$, for constant $c > 96$ determined by the analysis. Consider executing the algorithm EC-CKN on the network, and let $E\text{rank}^{(i)*}$ be the residual energy of the δth largest residual energy selected by a node in the ith epoch. We claim that, *w.h.p.*, all nodes with residual energy $< E\text{rank}^{(i)*}$ go to sleep. Because there are at most δ nodes with residual energy at least $E\text{rank}^{(i)*}$, we have the average number of the awake nodes in each epoch is

$$|\text{EC}|_k \leq \delta = \frac{ck \ln N}{\xi}. \tag{36}$$

And the average network lifetime under the algorithm EC-CKN can be written as

$$\mathbb{E}[\mathcal{L}_{\text{EC}}] = \frac{NT\mathcal{E}_0}{|\text{EC}|_k \cdot \mathcal{E}_{\text{epoch}} + (N - |\text{EC}|_k) \cdot \mathcal{E}_{\text{EC}}}, \tag{37}$$

where $\mathcal{E}_{\text{EC}} = \mathcal{E}_{\text{ckn}}$ is the energy consumed by executing the EC-CKN algorithm, and T is the length of an epoch. In comparison with the energy consumed by the awake nodes in an epoch, the energy consumed by the sleep nodes in an epoch is considered negligible:

$$\mathbb{E}[\mathcal{L}_{\text{EC}}] \approx \frac{NT\xi\mathcal{E}_0}{ck \ln N \cdot \mathcal{E}_{\text{epoch}}}. \tag{38}$$

□

For the algorithm EC-CKN, a node could have four states: *Init*, *Awake*, *Sleep*, and *Dead*. Let $\mathcal{S} = \{Init = 0, Awake = 1, Sleep = 2, Dead = 3\}$ be the set of the node's states, and $N_s = |\mathcal{S}|$ is the capacity of the states. Nodes can turn into the states *Awake*, and *Sleep* from the states *Init*, *Awake* and *Sleep*, respectively. And the state *Dead* can be only transformed from the states *Awake* and *Sleep*. Figure 4 shows the states transition graph in the algorithm EC-CKN, in which vertices are the states of nodes and the weights of edges are the transition probability between the two states in the ith epoch.

Theorem 7. *Under the algorithm EC-CKN, the difference of the energy consumption between nodes s_u and s_v in the ith epoch is*

$$d^n = \begin{cases} T_2(\Lambda E_{rx} + \Gamma E_{tx}), & \text{if } n = 0, \\ T_2 p_u^a(i-1)(\Lambda E_{rx} + \Gamma E_{tx}), & \text{otherwise}, \end{cases} \tag{39}$$

where Λ and Γ are both the Skellam distributions and $p_u^a(i-1)$ is the average probability that a node is awake in the $(i-1)$-th epoch.

Proof. Let $\mathcal{S}_u = \{\mathcal{S}_u^0, \mathcal{S}_u^1, \ldots, \mathcal{S}_u^i, \ldots\}$ denote the states of s_u in epochs, and let the chain

$$d : d^0, d^1, \ldots, d^i, \ldots \tag{40}$$

denote the difference between the energy consumed by nodes s_u and s_v in each epoch, where $s_v \in N_u$. $d^i = \alpha\mathcal{E}_{v,\text{epoch}}(i) - \beta\mathcal{E}_{u,\text{epoch}}(i)$, where $\mathcal{E}_{u,\text{epoch}}(i)$ and $\mathcal{E}_{v,\text{epoch}}(i)$ are the conventional energy consumption of nodes s_u and s_v if they are awake in the ith epoch. The factors α and β depend on the state of nodes s_u and s_v. If $\mathcal{S}_u^i = 0$ *or* 1, $\alpha = 1$, and if $\mathcal{S}_u^i = 3$, $\alpha = 0$, which is the same with β.

Suppose there are M independently and identically distributed trials. For the mth trial $(1 \leq m \leq M)$, the difference between the energy consumption of the two nodes in the ith epoch is

$$d_m^i = \alpha\mathcal{E}_{v,\text{epoch}}^m(i) - \beta\mathcal{E}_{u,\text{epoch}}^m(i). \tag{41}$$

The average difference of energy consumption in the ith epoch can be defined as

$$d^{(i)} = \lim_{M \to \infty} \frac{T_2}{M} \left[\frac{\sum_{m=1}^{M} \left[\left(\alpha N_{v,rx}^{m,i} - \beta N_{u,rx}^{m,i} \right) E_{rx} \right]}{T_2} + \frac{\sum_{m=1}^{M} \left[\left(\alpha N_{v,tx}^{m,i} - \beta N_{u,tx}^{m,i} \right) E_{tx} \right]}{T_2} \right], \tag{42}$$

where $N_{u,rx}^{m,i}$ and $N_{u,tx}^{m,i}$ are the number of received bits and transmitted bits of the node s_u in the ith epoch during the mth trial. T_2 is the time of the conventional data transmission time of each epoch. We discuss the difference between the energy consumption of two nodes in the following two cases.

Case 1 $(i = 0)$. According to (33) and (14), we obtain

$$d_{(0)} = T_2[(\lambda_{v,r} - \lambda_{u,r})E_{rx} + (\gamma_v - \gamma_u)E_{tx}]. \tag{43}$$

Case 2 $(i > 0)$. According to (33) and (34), we obtain

$$d_{(i)} = T_2 p_u^a(i-1)[(\lambda_{v,r} - \lambda_{u,r})E_{rx} + (\gamma_v - \gamma_u)E_{tx}], \tag{44}$$

where $p_u^a(i-1)$ is the expected probability that the node is awake in the $(i-1)$-th epoch.

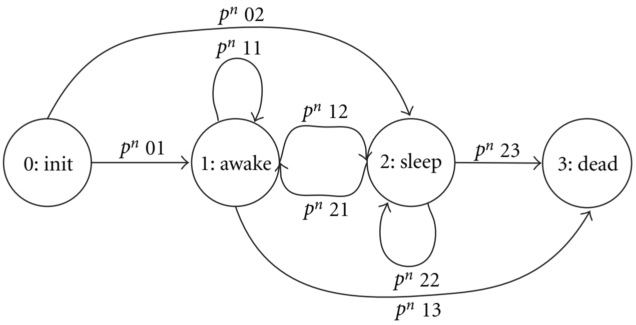


FIGURE 4: The Markov state transition probability graph of the EC-CKN algorithm.

Note for $\lambda_{v,r}$ and $\lambda_{u,r}$ are two independent Poisson distributions, $\Lambda = \lambda_{v,r} - \lambda_{u,r}$ is the Skellam distribution. Similarly, $\Gamma = \gamma_v - \gamma_v$ is also the Skellam distribution. □

Theorem 8. *Consider a random node s_u, which has more than k neighbors, the probability that a node s_u is awake in the ith epoch under the EC-CKN algorithm is*

$$p_a^i = \begin{cases} p_a(|E_u^0|), & \text{if } i = 0, \\ p_a(|E_u^i|), & \text{otherwise.} \end{cases} \tag{45}$$

Proof. We introduce a new chain

$$D : 0, d^0, D^0, d^1, D^1, \ldots, d^i, D^i, \ldots, \tag{46}$$

to denote the difference between the residual energy of the two nodes s_u and s_v, where $s_v \in s_u$. Then, we obtain

$$D^i = \begin{cases} d^0, & \text{if } i = 0, \\ D^{(i-1)} + d^0, & \text{otherwise.} \end{cases} \tag{47}$$

Therefore, the number of elements in the set E_u^i is $|E_u^i| = |N_u|\varrho^i$, where

$$\varrho^{(i)} = \begin{cases} \dfrac{D^0}{\max\{d^0\} - \min\{d^0\}} & \text{if } i = 0, \\ \dfrac{D^i}{\max\{D^i\} - \min\{D^i\}} & \text{otherwise.} \end{cases} \tag{48}$$

□

Theorem 9. *Under the EC-CKN algorithm, the upper bound of the network lifetime is*

$$\overline{\mathcal{L}}_{\text{EC-CKN}} = \frac{\mathcal{E}_0}{\min \sum_i \sum_a \sigma_{i,a} \mathcal{E}_{\text{epoch}}^n}, \tag{49}$$

where $\sigma_{i,a}$ is the steady-state probability that the action a is chosen when the chain is in the state i under the policy $\mathcal{K}$.

Proof. Now, we construct a Markov State Decision chain for each node s_u:

$$\mathcal{S}_u^0, a^0, \mathcal{S}_u^1, a^1, \ldots, \mathcal{S}_u^n, a^n, \ldots, \tag{50}$$

where $\mathcal{S}_u^0 = 0$, and $\mathcal{S}_u^n \in \{1, 2, 3\}$ denotes the state in the nth epoch ($n \geq 1$), and a^n is the action under the state $\mathcal{S}_u^n$

$$a^n = \begin{cases} \mathcal{E}_{u,\text{epoch}}^n, & \text{if } n = 0 \text{ or } \mathcal{S}_u^n = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{51}$$

Let the policy $\mathcal{K} = \{\kappa_{u,i}^n(a)\}$ denote the probability that the action a is chosen when $\mathcal{S}_u = i$, which satisfies the following two conditions:

$$\begin{gathered} 0 \leq \kappa_{u,i}(a) \leq 1, \quad \forall i, a, \\ \sum_a \kappa_{u,i}(a) = 1, \quad \forall i. \end{gathered} \tag{52}$$

Under the policy $\mathcal{K}$, the sequence of states $\mathcal{S}_u^n$ constitutes a Markov chain with the transition probability $p_u^n(\mathcal{K}, i, j)$, which can be written as

$$\begin{aligned} p_u^n(\mathcal{K}, i, j) &= p_u\{\mathcal{S}_u^{n+1} = j \mid \mathcal{S}_u^n = i\} \\ &= \sum_a p_u^n(i, j)\kappa_{u,i}(a). \end{aligned} \tag{53}$$

For the policy $\mathcal{K}$, let $\sigma_{i,a}$ denote the steady-state probability that the chain is in the state i and the action a is chosen:

$$\sigma_{i,a} = \lim_{n \to \infty} P_{\mathcal{K}}\{\mathcal{S}_u^n = i, a^n = a\} \tag{54}$$

The vector $\sigma = \{\sigma_{i,a}\}$ satisfies

$$\begin{aligned} &\text{(i) } \sigma_{i,a} \geq 0, \quad \forall i, a, \\ &\text{(ii) } \sum_i \sum_a \sigma_{i,a} = 1, \quad \forall i, a, \\ &\text{(iii) } \sum_a \sigma_{j,a} = \sum_i \sum_a \sigma_{j,a} p_{i,j}, \quad \forall i, a, j, \end{aligned} \tag{55}$$

Equations (55)(i) and (55)(ii) are obvious, and (55)(iii) follows as the left-hand side equals the steady-state probability of being in the state j and the right-hand side is the same probability computed by conditioning on the state and action chosen one epoch earlier.

Suppose that a reward $\mathcal{R}(Y_i^n, a_i^n) = a^n$ is earned whenever the action a_i^n is chosen in the state i in the n'th epoch. Since $\mathcal{R}(Y_i^n, a_i^n)$ denotes the reward earned at the epoch n, the expected average reward per epoch under the $\mathcal{K}$ policy can be written as

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\mathcal{K})] &= \lim_{n\to\infty}\frac{\sum_n \sum_{i\in\mathcal{S}} \mathcal{R}(Y_i^n, a_i^n)}{n}\\ &= \sum_i\sum_a \sigma_{i,a_i^n}\mathcal{R}(Y_i^n, a_i^n)\\ &= \sum_i\sum_a \sigma_{i,a_i} a^n.\end{aligned} \tag{56}$$

Therefore, $\mathbb{E}[\mathcal{R}(\mathcal{K})]$ can be interpreted as the following linear program:

$$\begin{aligned} &\min\sum_i\sum_a \sigma_{i,a}\mathcal{R}(Y_i^n, a_i^n)\\ &\text{subject to } \sigma_{i,a} \geq 0, \quad \forall i, a,\\ &\sum_i\sum_a \sigma_{i,a} = 1, \quad \forall i, a,\\ &\sum_a \sigma_{ja} = \sum_i\sum_a \sigma_{i,a} p_{i,j}, \quad \forall i, a, j,\end{aligned} \tag{57}$$

$\mathbb{E}[\mathcal{R}(\mathcal{K})]$ is a special case of the linear programm and is solved by a standard linear linear programming algorithm known as *the simplex algorithm*. The simplex algorithm solves the linear program by moving from an extreme point of the feasibility region to a better extreme point until the optimal is reached. So we can figure out the lower bound and upper bound of the lifetime by the linear programming (57). □

Lemma 10. *Under the EC-CKN algorithm, the upper bound network lifetime is*

$$\underline{\mathcal{L}}_{\text{EC-CKN}} = \frac{\mathcal{E}_0}{\max \sum_i \sum_a \sigma_{i,a}\mathcal{E}^n_{\text{epoch}}}, \tag{58}$$

where $\sigma_{i,a}$ is the steady-state probability that the action a is chosen when the chain is in the state i under the policy $\mathcal{K}$.

7. Simulation

Simulation Setup. In NetTopo [10], we conduct extensive simulation experiments. The studied WSN has the network size 800 × 600 m^2. The number of deployed sensor nodes are increased from 100 to 1000 (each time increased by 100). The value of k is changed from 1 to 10 (each time increased by 1). For every number of deployed sensor nodes, we use 100 different seeds to generate 100 different network deployment. A source node is deployed at the location of (50, 50), and a sink node is deployed at the location of (750, 550). The transmission radius for each node is 60 m.

Routing Algorithm. TPGF routing algorithm [11] is one of the earliest geographical multipath routing algorithms designed for facilitating the multimedia stream data transmission in static and always-on wireless sensor networks (WSNs). It focuses on exploring the maximum number of optimal node-disjoint routing paths in network layer in terms of minimizing the path length and the end-to-end transmission delay. TPGF routing algorithm includes two phases. Phase 1 is responsible for exploring the possible routing path. Phase 2 is responsible for optimizing the found routing path with the least number of hops. The simulation results with changed k values in this figure reflect the comparison between the original average length of paths and the optimized average length of paths.

Sleep Probability of the CKN Algorithm. The node's sleep probability has been analyzed in Section 4.1. Figure 5 describes the node's theoretic sleep probability based on the probability model that has been set up and covers the comparison between the simulation results and the theoretic value. We enlarge the some factors of the probabilistic model so that the theoretic value is greater than the simulation when $N/k > 100$. While $N/k < 100$, the theoretical results and simulation results approximate every much. Moreover, the same variation trend proves that our model is ponderable for nodes' sleep probability in the CKN algorithm.

Network Lifetime Under the CKN Algorithm. Based on the node sleep probability from the probabilistic model and simulation, we can get the relative probability stretch variation curve with the epoch. *Relative probability stretch* is defined as a function of the expected number of epochs with k active neighbors compared to the expected number of epochs with a larger $|N_u|$ active neighbors. In our work, we assume that the radio dissipates $E_{\text{elec}} = 50$ nJ/bit to the transmitter or receiver circuity and $\epsilon = 100$ pJ/bit/m^2, and the data rate is 20 kbps. In Figure 6, there is a key value of the length of the epoch time ($M \cdot t$) when N and k are certain, which is the intersection of relative probability stretch and reference axis. The CKN algorithm is energy efficient if the epoch is less than the key value. Otherwise, the node will consume more energy by the CKN algorithm than it is always active.

Network Lifetime Comparison between the CKN Algorithm and the EC-CKN Algorithm. The network lifetime of the CKN algorithm and the EC-CKN algorithm in a WSN is represented by the number of epochs. We conduct simulation for the CKN algorithm and the EC-CKN algorithm in a WSN and compare the network lifetime under the same situation in Figures 7 and 8. Results in Figures 7(a) and 7(c) confirm that the energy consumption of the EC-CKN algorithm is better managed and balanced than the CKN algorithm. Results in Figures 7(b) and 7(d) reveal the influence of changing the value of k: decreasing the value of k in the EC-CKN algorithm can prolong the network lifetime, particularly when the network nodes are densely deployed. Figures 7(a) and 7(b) show the whole distribution and tendency varying from the different combinations between k and N (the total number of deployed nodes in the network). Correspondingly, Figures 7(c) and 7(d) have the same meaning for the EC-CKN algorithm. Figure 8 compares the network lifetime between the CKN algorithm and the EC-CKN algorithm, by conducting the two algorithms under the same scenario. And the results give a straight proof that the

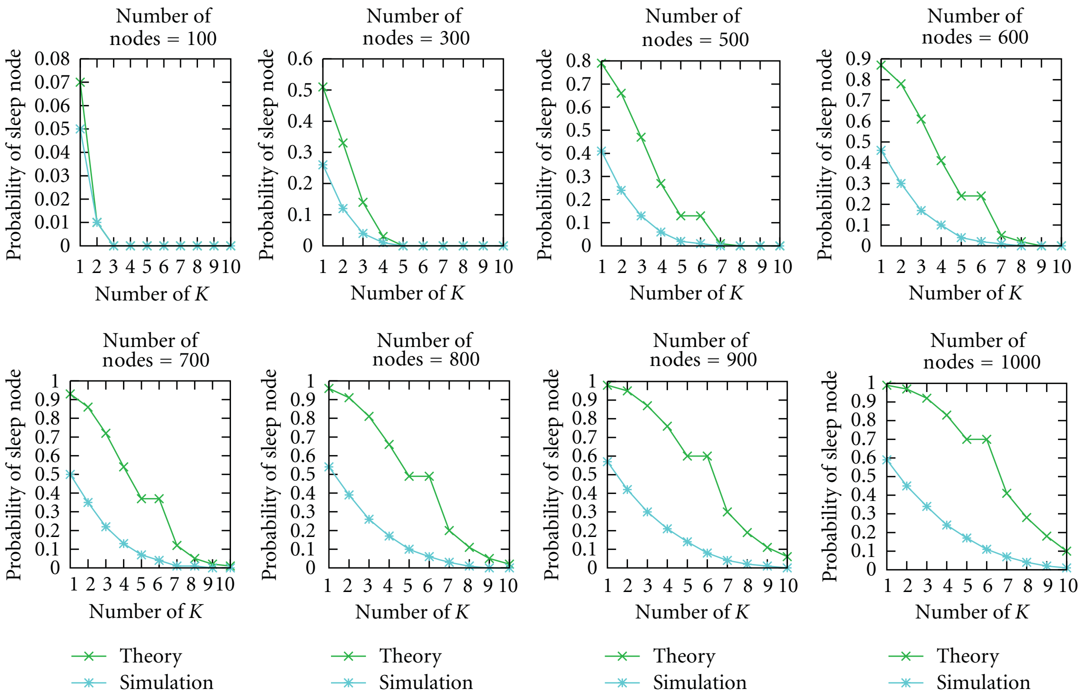


Figure 5: Node's sleep probability. The theoretical results and simulation results do not match well when the network size gets bigger, such as $N = 700\ 1000$. The reason is that we enlarge the probability that any two nodes in C_u are connected either directly themselves or indirectly through s_u's 2-hop neighbors.

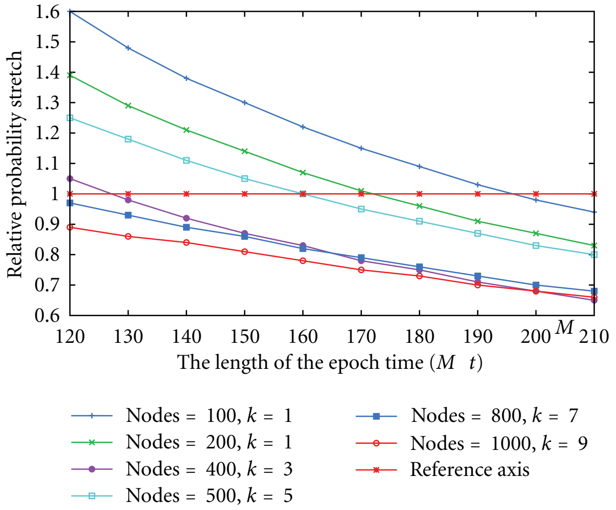
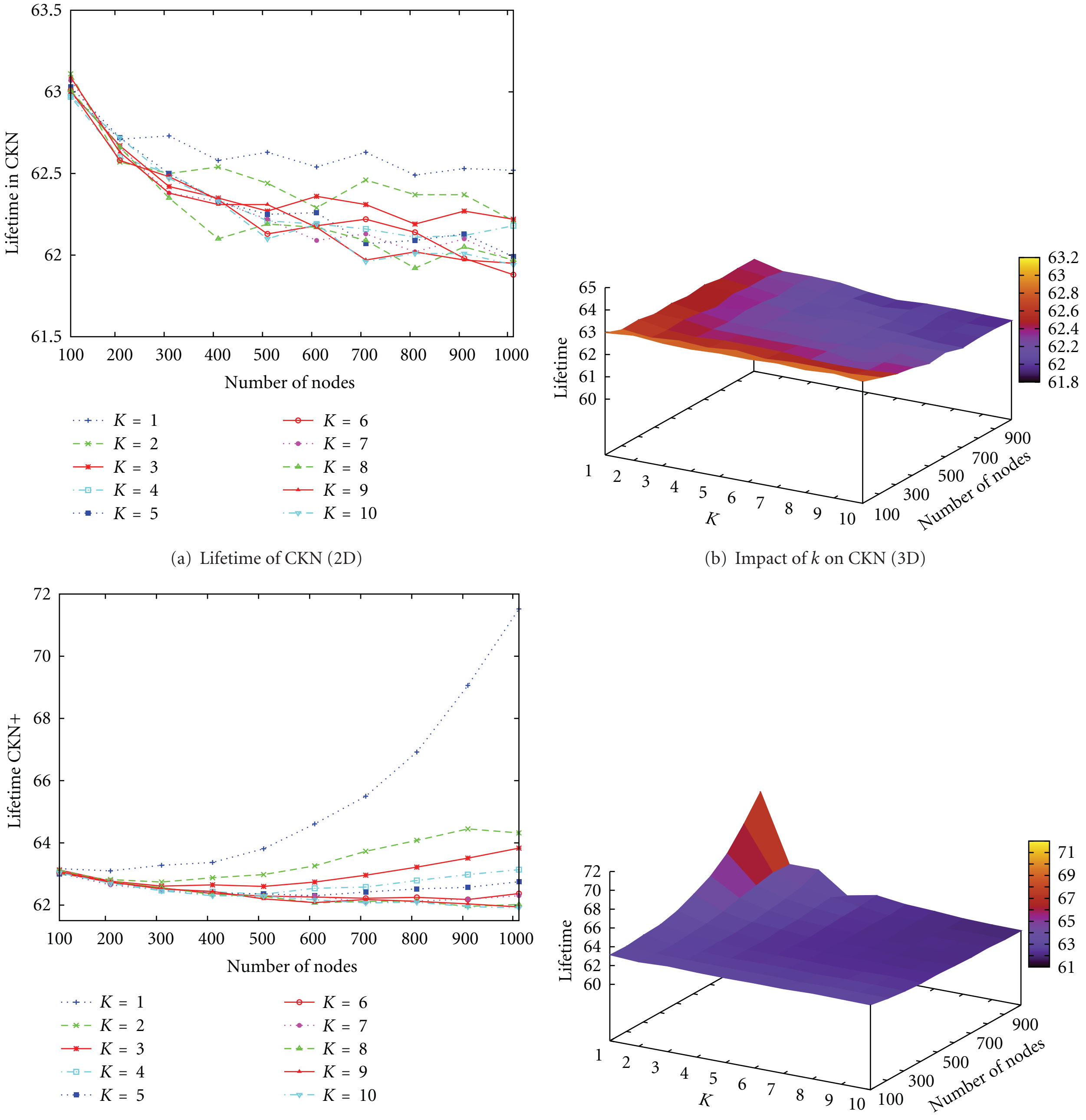


Figure 6: Relative probability stretch varies with epoch.

EC-CKN algorithm can provide longer network lifetime than the CKN algorithm.

8. Related Work

Network lifetime has been defined in various ways [12–17], and an energy-efficient mechanism may choose to maximize a certain type of network lifetime. One useful mechanism is the MAC layer power saving scheme, which reduces energy consumption by minimizing radio transceivers' idle time. SMAC [18] is an important MAC protocol designed for sensor networks, which forces sensor nodes to operate at low duty cycle by putting them into periodic sleep instead of idle listening. The timeout-MAC protocol (TMAC) improves SMAC by using an adaptive duty cycle [19]. Data-gathering MAC (DMAC) also uses an adaptive duty cycle, which provides low node-to-sink latency in convergecast communication by staggering the wake-up times of the nodes in the convergecast tree [20]. Pattern MAC (PMAC)



(a) Lifetime of CKN (2D)

(b) Impact of *k* on CKN (3D)

(c) Lifetime of EC-CKN (2D)

(d) Impact of *k* on EC-CKN (3D)

Figure 7: The network lifetime of running the CKN algorithm and the EC-CKN algorithm, respectively. In (a), the ten lines of the lifetime results according to the CKN algorithm have a lot intersections, which means the energy consumption by the CKN algorithm in a WSN is not managed towards the energy-balancing direction. However, in (c), the ten lines of the lifetime results by the EC-CKN algorithm present smooth changing when the number of nodes and the value of *k* are changed and the lifetime increases when the ratio N/k increases. This point clearly reflects that the energy consumption by the EC-CKN algorithm in a WSN is well managed towards the energy-balancing direction. Furthermore, simulation results in (b) and (d) also reveal that decreasing the value of *k* (let more nodes sleep) can definitely help to prolong the network lifetime of the EC-CKN algorithm in a WSN, but not the CKN algorithm.

[21] allows each sensor node determines the sleep-wake-up schedules based on its own traffic and the traffic patterns of its neighbors. BMAC [22] and XMAC [23] are two asynchronous duty-cycle-based protocols. In BMAC, each sensor node periodically wakes up to check whether there is any activity currently on the wireless channel or not. If so, the node remains active to receive a possible incoming packet. In this way, the node will receive one or more packets that are actually destined for other nodes. XMAC uses a strobed preamble to solve the overhearing

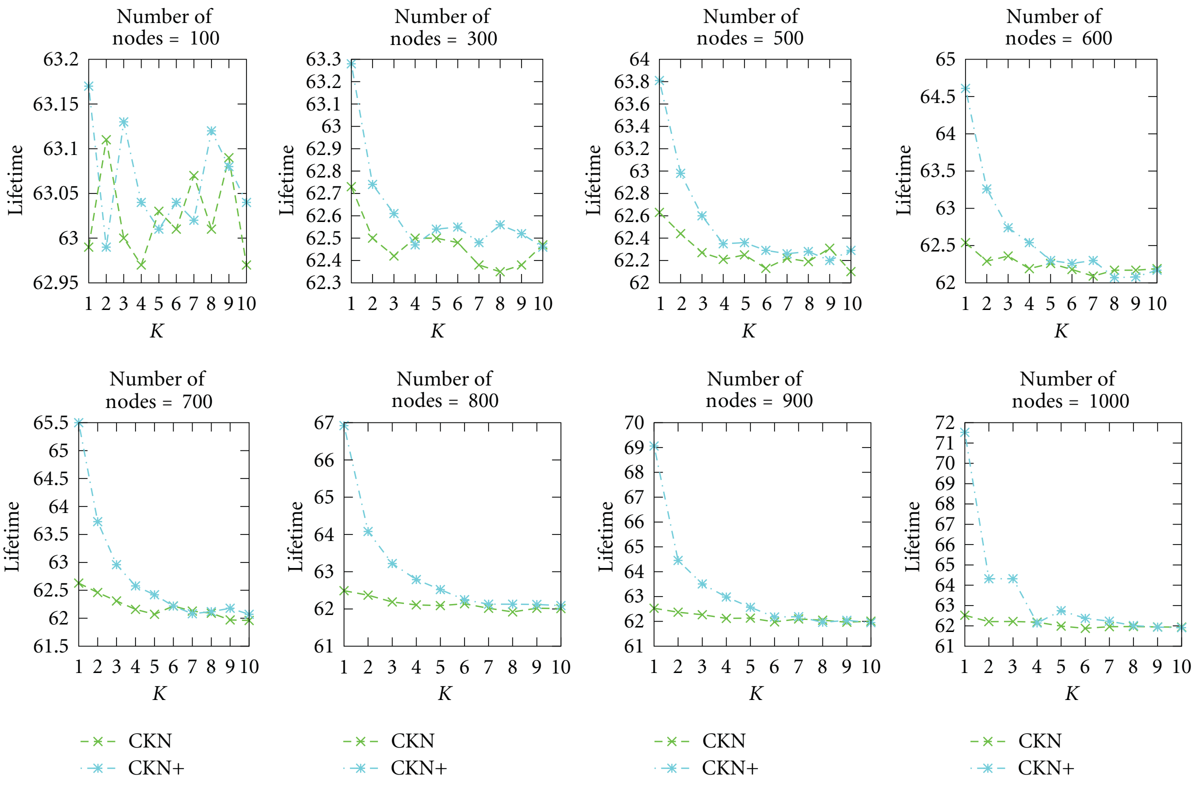


Figure 8: Network lifetime comparison between the CKN algorithm and the EC-CKN algorithm. Y-axle represents the recorded number of epochs when the first node runs out of its energy.

problem in BMAC. A strobed preamble includes a sequence of short preambles prior to DATA transmission. Obviously, the preamble transmission of BMAC and XMAC occupies the wireless medium for a long time under the light traffic load. To solve the problem, WiseMAC [24] is proposed. WiseMAC is similar to BMAC except that senders efficiently reduce the length of the wake-up preamble by exploiting the sampling of the schedules of its neighbors.

Another common technique to minimize the energy consumption and extend the network lifetime is to put some sensors in the sleep state and put others in the active state for the sensing and communication tasks. When a sensor is in the sleep state, its processor is turned off, but a timer or some other triggering mechanism may be running to wake up the sensor. On the other hand, all the components in the sensor are turned on when it is in the active state. Therefore, the energy consumed in the sleep state is only a tiny fraction of that consumed in the active state. One complexity here is that different types of sensors may support different sets of states. For example, the μAMPS sensor has three sleep states: *Monitor*, *Observe*, and *Deep Sleep* [25]. A sleep scheduling mechanism allows each sensor to determine when it should switch its state and what state it will switch to. Kumar et al. adopt the randomized independent scheduling (RIS) mechanism extending network lifetime while achieving asymptotic K-coverage [26]. RIS assumes that time is divided into epochs based on a time synchronization method. At the beginning of a epoch, each sensor independently decides weather to become active with probability p or go to sleep with probability $1 - p$. Thus, the network lifetime is increased by a factor close to $1/p$. Berman et al. presented a centralized and a distributed algorithm to maximize network lifetime while achieving K-coverage [27]. In the distributed algorithm, each sensor is in one of three states: *active*, *idle*, or *vulnerable*. In the vulnerable state, if the sensor discovers that part of its sensing area cannot be covered by any of its active or vulnerable neighbors, it immediately enters the active state. Otherwise, it enters the idle state if its sensing area can be monitored by either active neighbors or vulnerable neighbors with a higher energy level. Another distributed scheduling mechanism named lightweight deployment aware scheduling (LDAS) is proposed in [16]. Unlike the aforementioned distributed algorithm, LDAS does not ask if the sensor nodes are equipped with GPS or other devices to obtain location information. It assumes that each active node knows the number of its active neighbors. If the number of the active nodes exceeds a threshold, the node randomly selects some of its neighbors and sends tickets to them. When a node receives enough tickets from its neighbors, it may enter the sleep state after a random backoff period. Probing environment and adaptive sensing (PEAS) mechanism was designed for high-density sensor networks in a harsh environment [17]. Each node broadcasts a message with a transmission range of t_r after sleeping for a random period. A node will go to the active state only if it receives no replies from its active neighbors. PEAS assumes that sensor nodes may fail frequently and unexpectedly, which makes synchronized sleeping algorithm infeasible. Coverage configuration protocol (CCP) is an integrated coverage and connectivity configuration protocol [4]. The protocol defines that nodes have three states: ACTIVE, LISTEN, and SLEEP. Each node is initially ACTIVE. When it receives a message, it goes to the LISTEN state and starts a random LISTEN timer. The node will go to SLEEP if it satisfies the following conditions: (i) its LISTEN timer expires; (ii) the network is still connected when it goes to SLEEP. In the SLEEP state, a node will set a random SLEEP timer. When the timer expires, it will enter the LISTEN state. PECAS (probing environment and collaborating adaptive sleeping) is an extension of PEAS. A active node in PECAS will go back to sleep after a specified period of time. It piggybacks the remaining ACTIVE time in its reply messages to its neighbors' probe message. Therefore, an active neighbor who will go to SLEEP can schedule itself to wake up before the node goes to SLEEP, which prevents the occurrence of blind spots. Adaptive self-configuring sensor networks topologies (ASCENT) is similar to PEAS, which is also designed for high-density sensor networks. However, unlike PEAS, ASCENT does not guarantee network connectivity. Low-energy adaptive clustering hierarchy (LEACH) [28] is a cluster-based protocol, which utilizes randomized rotation of cluster heads to distribute work load among the sensors. In LEACH, the lifetime of network is divided into epochs, and each epoch includes a *set-up* phase and a *steady* phase. During the set-up phase, cluster heads are selected and each sensor joins a cluster by choosing a cluster that needs the minimum communication energy. During the steady phase, each cluster head aggregates the data from the sensors in its cluster and then forwards them to the sink. To conserve energy, nonhead sensors go to sleep at all time except that they are transmitting data. E-LEACH [15] is an extension of LEACH, which adapts the cluster heads selection algorithm to the nonuniform starting energy level among the sensors and the required number of cluster heads.

9. Conclusion

When deploying real WSNs for practical applications, it is extremely important to have a good sleeping scheduling algorithm to balance sensor nodes' energy consumption and a reasonable length for an epoch towards energy saving is extremely important. In this paper, we make the following major contributions for supporting the real WSNs applications. (1) A theoretical study is given for the CKN algorithm, which formulates the lower bound of an epoch to keep the CKN algorithm feasible. (2) A new sleep scheduling algorithm, named as EC-CKN, is proposed to balance the energy consumption and prolong the network lifetime. (3) Extensive simulation work is conducted, which proved the energy consumption in the EC-CKN algorithm is well balanced.

Acknowledgments

This work is partially supported by Natural Science Foundation of China under Grant no. 61070181 and Natural Science Foundation of Liaoning Province under Grant no. 20102021. L. Shu's research in this paper was supported by Grant-in-Aid for Scientific Research (S)(21220002) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] S. Nath and P. B. Gibbons, "Communicating via fireflies: geographic routing on duty-cycled sensors," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 440–449, April 2007.

[2] C. F. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 433–442, April 2004.

[3] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 20–27, April 2005.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenS '03)*, pp. 28–39, November 2003.

[5] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, pp. 62–74, 2001.

[6] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, "A cross-layer protocol for wireless sensor networks," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS '06)*, pp. 1102–1107, March 2006.

[7] 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 Siences (HICSS '00)*, p. 223, January 2000.

[8] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communications Letters*, vol. 9, no. 11, pp. 976–978, 2005.

[9] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02)*, pp. 80–91, June 2002.

[10] L. Shu, C. Wu, Y. Zhang, J. Chen, L. Wang, and M. Hauswirth, "NetTopo: beyond simulator and visualizer for wireless sensor networks," in *Proceedings of the 2nd International Conference on Future Generation Communication and Networking (FGCN '08)*, pp. 17–20, December 2008.

[11] L. Shu, Z. Zhou, M. Hauswirth, D. Le Phuoc, P. Yu, and L. Zhang, "Transmitting streaming data in wireless multimedia sensor networks with holes," in *Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia (MUM '07)*, pp. 24–33, Oulu, Finland, December 2007.

[12] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," in *Proceedings of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 1278–1287, June 2002.

[13] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.

[14] T. He, S. Krishnamurthy, J. A. Stankovic et al., "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 270–283, 2004.

[15] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[16] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 837–852, 2005.

[17] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 28–37, May 2003.

[18] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 1567–1576, New York, NY, USA, June 2002.

[19] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 171–180, Los Angeles, Calif, USA, November 2003.

[20] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 3091–3098, April 2004.

[21] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, p. 237, April 2005.

[22] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, November 2004.

[23] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys' 06)*, pp. 307–320, November 2006.

[24] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proceedings of the 1st International Workshop on Algorithm Aspects of Wireless Sensor Networks*, vol. 3121 of *Lecture Notes in Computer Science*, pp. 18–31, 2004.

[25] E. Shih, S. H. Cho, N. Ickes et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, pp. 272–286, July 2001.

[26] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 144–158, October 2004.

[27] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power efficient monitoring management in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, pp. 2329–2334, 2004.

[28] 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 Siences (HICSS '00)*, p. 223, January 2000.

[29] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, pp. 62–74, 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 947251, 10 pages
doi:10.1155/2012/947251

Research Article

Time-Independent Data Collection Protocol in Mobility-Assistant Wireless Sensor Networks with Duty Cycles

Xiwei Zhang,[1, 2, 3] Lijie Xu,[1, 2] and Guihai Chen[1, 2]

[1] *Department of Computer Science, Nanjing University, Nanjing 210000, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210000, China*
[3] *Department of Computer and Information, Hohai University, Nanjing 210000, China*

Correspondence should be addressed to Xiwei Zhang, zxw@hhu.edu.cn

Received 16 December 2011; Accepted 26 February 2012

Academic Editor: Mo Li

Mobility-assistant sensor networks comprise mobile elements, and static sensors are established for the purpose of solving the serious problems such as overlapping or energy holes in wireless sensor networks (WSNs). In such systems, most of the energy is consumed when the radios are on, waiting for a mobile sink (MS) to arrive. Sleep/wake scheduling is an effective mechanism to prolong the lifetime of these energy-constrained wireless sensors. However, sleep/wake scheduling could result in substantial discovery delays because the sensor needs time to receive the beacon-ID signals when MS entered its communication range. In this paper, we first study on the MS discovery mechanism and the factors which affect the efficiency of data collection. Based on these results, we then provide a solution to the control problem of how to optimally adjust the system parameters of the sleep/wake scheduling protocol to maximize the network lifetime, subject to a constraint on the expected residual contact time. Our numerical results indicate that the proposed solution can balance the network consumption, especially in sparse sensor networks.

1. Introduction

Recently, wireless sensor networks (WSNs) are developed and used for information collection such as environmental monitoring [1, 2], automatic controlling [3, 4], and target tracking [5, 6]. In many data-centric applications, they often produce high-bandwidth sensor data that need to be collected under stringent delay constraints. There are multiple ways in which the sensor readings are transferred from the sensors to a central location. Usually, the readings are relayed to a stationary sink for processing using the ad-hoc multihop network formed by the sensor nodes. Though this is surely a feasible technique for data transfer, it creates a bottleneck in the network. This leads to a nonuniform depletion of network resources, and the nodes near the sink are the first to run out of batteries. If these nodes die, the network for all practical purposes will be disconnected. Periodically replacing the battery of the nodes for the large scale deployments is also infeasible.

In this paper, we exploit the mobility-assistant network to improve the data collection performance of WSNs. A mixture of networked *mobile sink* (MS) and static sensors reduces the cost but preserves the flexibility and advantageous capacities of a mobile system. In a hybrid sensor network, MS is generally equipped with more resources such as sensors, power, and computation, so forth. It can collect data from the sensor nodes and handle or merge the data directly. As a result, significant network energy saving can be achieved by reducing or completely avoiding numerous wireless communications and will prolong the lifetime of network.

The communication between the MS and sensor nodes takes place in two different phases. First, sensor nodes have to discover the presence of the MS in their communication range. Then, they can transfer collected data to the MS while satisfying certain reliability constraints, if required. Different from MS, sensor nodes have a limited energy budget, so that both discovery and data transfer should be energy efficient in order to prolong the network lifetime [3]. As the radio component is usually the major source of energy consumption, the overall activity of the radio should be minimized. To this end, a duty cycle approach can be used, so that sensors alternate sleep and active periods. However, the effects of the duty cycle have to be properly investigated: if sensor nodes are sleeping when the MS comes, they cannot detect it and transmit data, so that they are only wasting their energy.

In general, a low duty cycle provides better energy efficiency, it is also true if the contact time, which is the time when MS within the communication area of sensor, is large enough to allow the reliable transfer of a significant amount of data. However, when the contact time is limited, a very low duty cycle is not convenient as the energy saved during discovery is overcome by the decrease in the number of messages successfully transferred [7]. Hence, it is necessary to properly define the discovery parameters and the duty cycle to maximize the network lifetime.

The most efficient MS discovery method is the synchronous scheme which assumes the sensor nodes know exactly when the MS will enter the contact area and can thus wake-up at predefined times [8]. Obviously, such approaches require that the mobility of the MS is accurately known in advance. However, this assumption is rather strong, and the synchronization of MS and sensors is difficult to realize. In this paper, we develop an asynchronous and low-complexity solution to data gathering problem: a time-independent data collection protocol. We formulate the *lifetime maximization problem*: given a constraint on the expected residual contact time, how to maximize the network lifetime by controlling the wake-up rates but not concern the time when MS reached the sensors. We show how to use the solution to residual contact time maximization problem to construct an optimal solution to the lifetime-maximization problem for a specific definition of network lifetime.

The rest of the paper is organized as follows. Section 2 presents an overview of the relevant literature in the field. Section 3 introduces the system model and the related assumptions. Section 4 describes the MS discovery process in detail. In Section 5, we solve the lifetime-maximization problem using time-independent model. In Section 6, we provide simulation results that illustrate the performance of our proposed algorithm. Finally, Section 7 concludes the paper.

2. Related Works

Mobile nodes often used as data collectors are named as Data Mules or Data Ferries [4–6]. These nodes could be persons, cars, and animals, and so forth. In the GreenOrbs project [9], mobile user equipped communication device, like PDA, walks alone the trails to collect data from sensors which were deployed in Tianmu Mountain. Luo et al. [10] investigate the event collection problem in tough communication environment (e.g., underground coal mine) with the help of mobile sinks.

In order to reduce the transmission delay caused by slow-speed movement of MS, path selection problem are elaborately studied in these papers. In [11], authors studied path selection problem, assuming that each sensor buffers the sensing data and the data mule needs to collect it before the buffer overflows. Gandham et al. [12] used multiple mobile base stations to prolong the lifetime of the sensor networks. Xing et al. [13] presented path selection algorithms for rendezvous-based approach.

Synchronous schemas are used in some mobility-assistant discovery protocols. In [8], for instance, MEs are assumed to be on board of public transportation shuttles which visit sensor nodes according to a tight schedule. As an alternative, nodes can just define a networkwide active time and wake-up accordingly, so that they can contact the neighboring nodes which are available at that time. This kind of approach is adopted in ZebraNet [14], where nodes are synchronized through a Global Positioning System (GPS). Obviously, such approaches require strict synchronization or that the mobility of the nodes is accurate enough to obey schedules, and this limits the applicability of scheduled synchronous schemes to practical scenarios.

Asynchronous schemes define sleep/wake-up patterns such that nodes can communicate without explicitly agreeing on their activation instants [7, 15–17]. More specifically, the MS sends periodic discovery messages, while the static node cyclically wakes up and listens for advertisements for a short time. If it does not detect any discovery message, it can return to sleep; otherwise, it can start transferring data to the MS [18]. To ensure that the sensor can receive a discovery message independent of its sleep/wake-up schedule, the discovery parameters and the duty cycle have to be properly defined [19]. In both cases, it is analytically shown that using mobile elements leads to a network lifetime much longer than with a static sink. However, these papers only give the analysis of the mobility discovery and transfer protocol, they did not consider the network lifetime or other performance metric. Our approach also is an asynchronous scheme which does not concern when the MS enters the contact area, it can satisfy the transmission demand of application and maximize the network lifetime at the same time.

3. System Model

In this section, we will introduce the reference network scenario depicted in Figure 1, which is also used in [15, 16]. We consider a wireless sensor network with N nodes; let $\mathcal{N}$ denote the set of all nodes in the network. One mobile sink moves along a linear path at a fixed vertical distance (D_y) from a static node.

Data collection takes place only during a contact, that is, when the static node and the MS can reach each other. Furthermore, the area within the radio transmission range r of the static node is called *contact area*, while the overall time spent by the MS inside the contact area is called *contact time* and is referred to as $C_{\max}$.

The overall data collection process can be split into three main phases. As MS arrivals are generally unpredictable, the static node initially performs a discovery phase for the timely detection of the MS. Indeed, the successful MS detection by the static sensor is not immediate but requires a certain amount of time, called *discovery time*, and denoted as d in Figure 1. Upon detecting the MS, the static node switches from the discovery state to the data transfer state and starts transmitting data to the MS. As a result of the discovery

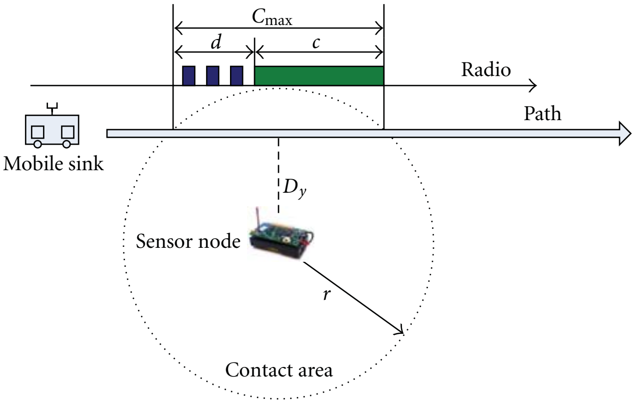


Figure 1: Reference scenario.

process, the static node cannot exploit the whole available contact time for data transfer. The portion of the contact time which can be actually used for subsequent data transfer is called *residual contact time*. After the end of the data transfer phase, the static node may switch to the discovery state again in order to detect the next MS passage.

To advertise its presence in the surrounding area, the MS periodically sends a beacon signal and an ID signal (carrying the receiver information) for time periods t_B and t_C, respectively, and then hears the medium for time period t_A (see Figure 2). For ease of notation, we define the *beacon period* indicated by $t_I \triangleq t_C + t_B + t_A$. If the MS does not hear any acknowledgment signal from the sensor i, it repeats this signaling procedure. When the sensor i wakes up and obtains the beacon signal, it keeps awake, waiting for the following ID signal to recognize the receiver. When node wakes up in the middle of an ID signal, it keeps awake, waiting for the next ID signal. If the sensor successfully recognizes itself the receiver of the MS, it then communicates with MS to receive the packet. In order to save energy during the discovery phase, the static node operates with a duty cycle, defined by the active time t_{on} and the sleep time t_{off}. Since nodes consume energy when awake, t_{on} should be as small as possible. However, if t_{on} is too small, a node that wakes up right after an ID signal could return to sleep before the following beacon signal. In order to avoid this case, we set $t_{\text{on}} = t_A + \epsilon_{\text{detect}}$, where ϵ_{detect} is a small amount of time required for a node to detect signal in the wireless medium. In the rest of the paper, we assume that ϵ_{detect} is negligible compared to t_A.

We make the following assumptions in this paper:

(1) the communication range of sensor node is a circular area with radius of r and the speed of MS is a constant of v;

(2) the MS is power of energy and computing ability and with a large storage;

(3) an arbitrary moving path is deterministic which connected all sensors that would transmit data to MS, and whole data of one sensor must transfer to MS in one contact;

(4) to simplify the computation, we assume the moving path is linear, and the movement distance in one contact area is $2r$, that is, $D_y = 0$ and $C_{\max} = 2r/v$;

(5) the time instants that a node i wakes up follow a Poisson random process with rate λ_i and the wake-up processes of different nodes are independent.

4. Analysis of MS Discovery Process

Although MS discovery by sensors with sleep/wake-up patterns is an asynchronous schema whose properties ensure that sensors are able to communicate without explicitly agreeing on their activation instants, the residual contact time is decided by the time MS enters the contact area. The residual contact time will be a maximum if the node's state is ON at that time, otherwise it will be shorten. On the other hand, the residual contact time is not a deterministic value for the MS moving alone different paths and the contact will be occurred at different time.

In detail, the time instant at which the MS transmits the first beacon-ID signal while in the contact area is denoted as t_0. As beacon transmissions are periodic and start at t_0, the actual instants of subsequent beacon transmissions can be expressed as $t_n = t_0 + n * t_I$, with $n \in [1, N-1]$, where $N = \lceil C_{\max}/t_I \rceil$ is the maximum number of beacon-ID signals the MS can send while in the contact area. Therefore, if the MS is discovered by means of the mth beacon, the discovery time is $d = d_0 + m * t_I$, and the corresponding residual contact time is $C_{\max} - d$.

As shown in [15], the state of the static node at a given instant can be specified by its composite state (s, l) where s denotes the radio state, that is, ON or OFF, and l represents the residual time, that is, the amount of time the node will remain the same state s. The initial state of the static node at the time $t = 0$ is referred to as (s_0, l_0) as shown in Figure 3. Let us denote by $s(t)$ and $l(t)$ the radio state and the residual time, respectively, at a generic time t. Because of the duty cycle, both $s(t)$ and $l(t)$ evolve in a deterministic way. In detail, the radio state of the static node is periodic, with period equal $t_{\text{ON}} + t_{\text{OFF}}$. We focus now on the radio state $s(t_n)$ of the static node at the time MS enters the contact area. As $s(t)$ is periodic, it is sufficient to investigate the remainder of the ratio between the beacon-ID transmission times and the period of the duty cycle. By comparing this remainder against the initial residual state s_0 and the initial residual time l_0, we can derive $s(t_n)$. The number of the state changing from ON to OFF (or OFF to ON) is $\lfloor t_n/(t_{ON} + t_{\text{OFF}}) \rfloor$, and the remaining time is $\Delta t = t_n \mod (t_{ON} + t_{\text{OFF}})$, the state of sensor is

$$s(t_n)_{s_0=\text{ON}} = \begin{cases} \text{ON}, & \text{if } 0 \le \Delta t < l_0, \\ \text{OFF}, & \text{if } l_0 \le \Delta t < l_0 + t_{\text{OFF}}, \\ \text{ON}, & \text{if } l_0 + t_{\text{OFF}} \le \Delta t < t_{\text{ON}} + t_{\text{OFF}}, \end{cases}$$

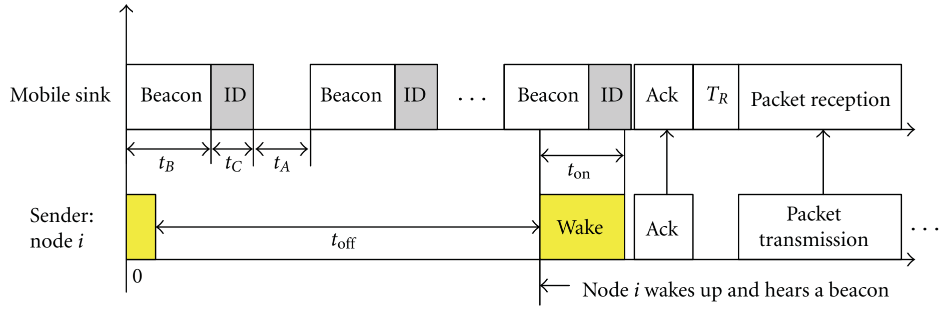


Figure 2: MS discovery model.

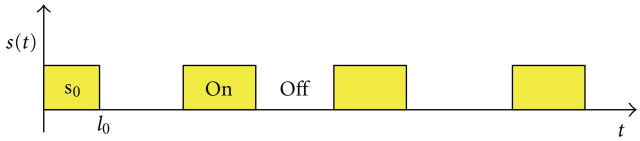


Figure 3: State changing of sensors.

$$s(t_n)_{s_0=\text{OFF}} = \begin{cases} \text{OFF}, & \text{if } 0 \le \Delta t < l_0, \\ \text{ON}, & \text{if } l_0 \le \Delta t < l_0 + t_{\text{ON}}, \\ \text{OFF}, & \text{if } l_0 + t_{\text{ON}} < \Delta t < t_{\text{ON}} + t_{\text{OFF}}. \end{cases} \tag{1}$$

Similarly, we can also derive the residual time $l(t_n)$:

$$l(t_n)_{s_0=\text{ON}} = \begin{cases} l_0 - \Delta t, & \text{if } 0 \le \Delta t < l_0, \\ t_{\text{OFF}} + l_0 - \Delta t, & \text{if } l_0 \le \Delta t < l_0 + t_{\text{OFF}}, \\ t_{\text{ON}} + t_{\text{OFF}} + l_0 - \Delta t, & \text{if } l_0 + t_{\text{OFF}} \le \Delta t < t_{\text{ON}} + t_{\text{OFF}}, \end{cases}$$

$$l(t_n)_{s_0=\text{OFF}} = \begin{cases} r_0 - \Delta t, & \text{if } 0 \le \Delta t < l_0, \\ t_{\text{ON}} + l_0 - \Delta t, & \text{if } l_0 \le \Delta t < l_0 + t_{\text{ON}}, \\ t_{\text{ON}} + t_{\text{OFF}} + l_0 - \Delta t, & \text{if } l_0 + t_{\text{ON}} \le \Delta t < t_{\text{ON}} + t_{\text{OFF}}. \end{cases} \tag{2}$$

5. Time-independent Data Collection Protocol

5.1. Wake-Up Rates and Awake Probability. In this section, we present the time-independent data collection protocol to maximize the network lifetime. As we assumed, the sleep/wake schedule is determined by the wake-up rate λ_i of the Poisson process with which each node i wakes up. Obviously, if λ_i increases, the expected discovery delay will be reduced and the data communication time between MS and sensor will be extended. However, a larger wake-up rate leads to higher energy consumption and reduces network lifetime.

In the rest of the paper, it is more convenient to work with the notion of awake probability which is a function of λ_i. Suppose that MS sends the first beacon signal at the initial time, as in Figure 2. If the sensor has not heard the first $(m-1)$ beacon and ID signals, then node transmits the mth beacon and ID signal in the time interval $[(t_C + t_B + t_A)(m-1), (t_C + t_B + t_A)(m-1) + t_C + t_B]$. For the node i to hear the mth signals and to recognize the sender, it should wake-up during $[(t_C + t_B + t_A)(m-1) - t_C - t_A, (t_B + t_A)m - t_C - t_A]$. Therefore, provided that MS is sending the mth signal, the probability that node i wakes up and hears this signal is

$$p_i = 1 - e^{-\lambda_i (t_C + t_B + t_A)}. \tag{3}$$

We call p_i the *awake probability* of sensor i.

Note that there is a one-to-one mapping between the awake probability p_i and the wake-up frequency λ_i. Hence, the awake probability is also closely related to both delay and energy consumption. Let $\vec{p} = (p_i, i \in \mathcal{N})$ represent the global awake probability vector.

5.2. Performance Metrics. In this subsection, we define the performance metrics of the time-independent policy and the sleep/wake scheduling policy that we intend to optimize. We know, although the sleep-wake patterns and the data collection policy are applied in the operation phase of the network, their control parameters are optimized in the configuration phase.

5.2.1. Residual Contact Time. As mentioned above, the residual contact time is used for data communication after the MS has been discovered by sensor node. Upon receiving a beacon-ID from the MS, the static node enters the data transfer state. While, in this state, the static node remains always active to exploit the residual contact time as much as possible. On the other hand, the MS enters the data transfer phase as soon as it receives the first message sent by the static node, and stops beacon transmissions. Hence, the amount of data exchanged between MS and sensor is decided by the residual contact time, it must be maximized to transfer large mount of data. The maximum value of the residual contact time is $2r/v$ if the state of sensor is ON when MS enters the contact area, and the minimum value is 0 if the state is OFF before MS leaves the contact area.

For the definition of the residual contact time $c = C_{\max} - d$, it is affected by the discovery time which is determined by the awake probability of the sensor node. We denote $C_i(p_i, v, r)$ as the expected residual contact time when the awake probability vector p_i, the speed of MS, and the communication radius of sensor i are given.

Our objective is to maximize the residual contact time to satisfy the data transfer demand:

$$\max C_i(p_i, v, r) \geq \xi_i, \tag{4}$$

where ξ_i is the minimum time to satisfy the data transfer demand. This problem is to find optimal sleep/wake scheduling policy to maximize the sensor's residual contact time.

5.2.2. Network Lifetime. We now introduce metric, the network lifetime, and the corresponding lifetime maximization problem (subject to residual contact time constraints). Let Q_j denote the energy available to node j. We assume that node j consumes μ_j units of energy each time it wakes up. We define the expected lifetime of node j as $Q_j/\mu_j\lambda_j$. Note that implicitly in this definition of lifetime we have chosen not to account for the energy consumption by data transmission. As mentioned in the introductory section, this is a reasonable approximation for event-driven sensor networks in which events occur very rarely because the energy consumption of the sensor nodes is dominated by the energy consumed during the sleep/wake scheduling.

By introducing the power consumption ratio $e_j = \mu_j/Q_j$, we can express the lifetime of node j as

$$T_j(p_j) = \frac{1}{e_j\lambda_j} = \frac{t_C + t_B + t_A}{e_j \ln\left(1/\left(1 - p_j\right)\right)}. \tag{5}$$

Here, we have used the definition of the awake probability $p_j = 1 - e^{-\lambda_j(t_C+t_B+t_A)}$ from (3).

We define *network lifetime* as the shortest lifetime of all nodes. In other words, the network lifetime for a given awake probability vector $\vec{p} = (p_i, i \in \mathcal{N})$ is given by

$$T(\vec{p}) = \min_{i\in\mathcal{N}} T_i(p), \tag{6}$$

Based on the above performance metrics, we present the *lifetime maximization problem* as follows:

$$\begin{gathered} \max_{\vec{p}} T(\vec{p}) \\ \text{subject to}\, C_i(p_i, v, r) \geq \xi_i, \quad \forall i \in \mathcal{N}, \\ \vec{p} \in (0, 1]^N. \end{gathered} \tag{P}$$

The objective of the above problem is to choose the optimal sleep/wake scheduling policies that maximize the network lifetime and also guarantee that the residual contact time of each node is not less than the minimum allowable data transfer time of this node.

5.3. Data Collection Protocol. We first derive the relationship between the expected residual contact time and awake probability.

Proposition 1. *For any sensor $i \in \mathcal{N}$, if the MS enters its contact area with a constant speed v, the expected residual contact time for a given awake probability vector p_i is*

$$C_i(p_i, v, r) = \frac{2r}{v} - t_I \times \frac{1 - (1 - p_i)^{2r/v*t_I}}{p_i}. \tag{7}$$

Proof. The MS sends beacon-ID signals when it is within sensor's contact area. We define the probability p_i, h, which denotes sensor i that received the hth beacon-ID signals is equal to the probability that i has not woke up for the past $h - 1$ beacon-ID signaling iterations and it wakes up at hth beacon-ID signals, that is,

$$p_{i,h} = (1 - p_i)^{h-1} \times p_i. \tag{8}$$

We define the number of beacon-ID signals during the time MS within the contact area $n = 2r/(v \times t_I)$, so

$$\begin{aligned} C_i(p_i, v, r) &= \sum_{h=1}^{n} p_{i,h} \times \left(\frac{2r}{v} - h \times t_I\right) \\ &= \sum_{h=1}^{n} (1 - p_i)^{h-1} \times p_i \times \left(\frac{2r}{v} - h \times t_I\right) \\ &= p_i * \frac{2r}{v} * \sum_{h=1}^{n} (1 - p_i)^{i-1} - p_i * \frac{2r}{v * n} \\ &\quad * \sum_{h=1}^{n} (1 - p_i)^{i-1} * i \\ &= p_i * \frac{2r}{v} * \frac{1 - (1 - p_i)^n}{p_i} - p_i \\ &\quad * \frac{2r}{v * n} * \left(\frac{1 - (1 - p_i)^n}{p_i^2} - \frac{n * (1 - p_i)^n}{p_i}\right) \\ &= \frac{2r}{v} - \frac{2r}{v * n} * \frac{1 - (1 - p_i)^n}{p_i}. \end{aligned} \tag{9}$$

Hence, we can get the result when substituting n with $2r/(v \times t_I)$. □

From (5), the lifetime T_i and the awake probability p_i have a one-to-one mapping. Hence, we convert Problem (P) to the following equivalent problem that controls $\vec{T} = (T_1, T_2, \ldots, T_N)$,

$$\begin{gathered} \max \min_{i\in\mathcal{N}} T_i, \\ \text{subject to}\, C_i(p_i, v, r) \geq \xi_i, \quad \forall i \in \mathcal{N}, \end{gathered} \tag{P1}$$

$$p_i = 1 - e^{-t_I/e^i T_i}, \quad \forall i \in \mathcal{N}. \tag{10}$$

Proposition 2. *If $\vec{T}^* = (T_1^*, T_2^*, \ldots, T_N^*)$ is an optimal solution to Problem* (P1)*, then so is $\vec{T}$ such that $\vec{T} = (T_i = \min_k T_k^*, i \in \mathcal{N})$. In other words, according to the lifetime definition* (6)*, it is no worse in terms of both the network lifetime and the residual contact time to let all nodes set their lifetime to the shortest lifetime.*

Proof. Since both solutions have the same objective value under our network lifetime definition in (6), it is sufficient to show that if $\vec{T}^*$ is in the feasible set, so is $\vec{T}$. Let $\vec{p}^*$ and $\vec{p}$ be the awake probability vectors that correspond to $\vec{T}^*$ and $\vec{T}$, respectively, by (10). Since p_i monotonically decreases as T_i increases, and $\vec{T} \preceq \vec{T}^*$, we have $\vec{p}^* \preceq \vec{p}$. (The symbol "$\preceq$" denotes componentwise inequality, i.e., if $\vec{T} \preceq \vec{T}^*$, then $\vec{T} \leq \vec{T}^*$ for all $i \in \mathcal{N}$.) □

We prove the residual contact time from any node i is a nondecreasing function with respect to each component of p_i. (The detailed proof is provided in the appendix). For a given p_i, assume that node increases its awake probability to p_i'. Since the increased awake probability does not decrease the residual time of the node, we have

$$C_i(p_i, v, r) \leq C_i(p_i', v, r). \tag{11}$$

The inequality in (11) is due to the residual time optimality of for p_i. Hence, the delay is nondecreasing with respect to each component of $\vec{p}$. Since $\vec{p}^* \preceq \vec{p}$, we have $C_i(\vec{p}, v, r) \geq C_i(\vec{p}^*, v, r)$. Hence, if $\vec{T}^*$ satisfies problem (P1), so does $\vec{T}$.

Using the above proposition, we can rewrite problem (P1) into the following problem with one-dimensional variable that corresponds to the network lifetime:

$$\begin{aligned} &\max T, \\ &\text{subject to}\, C_i(p_i, v, r) \geq \xi^*, \quad \forall i \in \mathcal{N}, \\ &\min_{i \in \mathcal{N}} p_i = 1 - e^{-t_I/e^i T}, \quad \forall i \in \mathcal{N}, \end{aligned} \tag{P2}$$

where $\xi^* = \max \xi_i, i \in \mathcal{N}$.

Note that $\min_{i\in\mathcal{N}} C_i(p_i, v, r)$ is nondecreasing with respect to each component of p_i. (See the proof of Proposition 2). Since each component p_i of is a decreasing function of T, $\min_{i\in\mathcal{N}} C_i(p_i, v, r)$ is decreasing as increases T. Hence, we can develop an efficient binary search algorithm (see Algorithm 1 that follows) for computing the optimal value T^* of such that $\min_{i\in\mathcal{N}} C_i(p_i, v, r) = \xi^*$.

After $m_{\max}$ iterations of the Binary Search Algorithm, the difference between the optimal lifetime T^* and the algorithm output T_{record} is smaller than $(T^{(1)}/2^{m_{\max}})$. If we want to make this difference less than ε, the complexity of the Binary Search Algorithm is $O(\log(1/\varepsilon) \cdot N)$.

6. Simulations and Experiments

In order to evaluate the performance of MS discovery and our time-independent data collection protocol, we test the contact time and network lifetime with NS2 simulator and physical experiments, respectively.

6.1. Residual Contact Ratio. In this section, we will use the analytical formulas derived in the previous sections to perform an analysis of the MS discovery process. To this end, we will consider the following performance metrics of *residual contact ratio*, which is defined as the average of the ratio between the residual contact time and the whole contact time:

$$\eta = \mathbb{E}\left[\frac{C_i(p_i, v, r)}{c_{\max}}\right]. \tag{12}$$

In all simulations we performed 10 replicas, each consisting of 1000 passages. The obtained confidence intervals are always very low (below 1%) and are thus omitted. In the following, we will show both analytical and simulation results. However, unless stated otherwise, we will refer to the analytical results.

And the simulations are written in C++. The radio parameters are set according to the data sheet of the CC2420 radio on Mica2 motes [20]. Radio bandwidth is 40 Kbps, and transmission power is 4 dbm with the current consumption of 11.6 mA. The size of each packet is 30 bytes. To simulate the highly unreliable links of WSNs, we implemented a link layer model from USC [21]. Experimental data shows that the USC model can capture the highly probabilistic link characterization of Mica2 motes [21].

From formula (7), the effectiveness of the discovery strictly depends on the beacon period t_I, the moving speed v and the awake probability of the sensor. Here, the parameter t_I must be larger than or equal to the beacon duration t_B (10 ms in our scenario). In addition, the discovery process is clearly influenced by these parameters.

Figures 4 and 5 show the residual contact ratio for three different beacon periods and several wake-up rates (mapping to awake probability), when the MS moves at 1 m/s and 8 m/s, respectively. To validate our analytical model we also performed some simulation experiments under the same conditions. The comparison between simulation and analytical results shows that our model is very accurate.

The results in Figure 4 show that the static node quickly discovers the MS when it moves slowly (i.e., at 1 m/s). As expected, for a fixed wake-up rate, the residual contact ratio decreases when t_I increases. However, for a given beacon period, the residual contact ratio increases with the wake-up rate. In this scenario, most of the contact time can be effectively used for communication even when the wake-up rate is very low. The results are different, instead, when the speed of the MS is high (i.e., 8 m/s). From Figure 5, we can see that the residual contact ratio drops sharply when the wake-up interval $(1/\lambda)$ is from 10 to 10^{-1}, especially for large values of t_I. Intuitively, this behavior can be explained as follows. For the same beacon period, a lower wake-up rate involves longer delays in the detection of the MS. When the MS moves fast, the time it remains in the contact area is lower (in this scenario the average contact time is about 20 s): hence, a late discovery results in a small amount of time left for data transfer.

6.2. Network Lifetime. We start our physical experiments by collecting data using the setup shown in Figure 6 where we placed an 11 × 11 grid on a 15 m × 15 m indoor environment. The sensor motes are also TelosB which uses the CC2420 chip. They are IEEE 802.15.4 compliant. We

```
**Initial Setup**: **The sink** sets $T^{(1)}$ to a half of the maximum possible lifetime and sets $T_{\text{record}} \leftarrow 0$
**for** $m = 1$ to $m_{\max}$ do
    **Every node** $i$ computes $p_i^{(m)} = 1 - e^{-(t_I/(T^{(m)}e_i))}$.
    **Node** $i$ sends their feedback of their transmission time $C_i$ to the sink.
    **The sink** sets $C_{\min} \leftarrow \min_i C_i$ and
    **if** $C_{\min} < \xi^*$ **then**
    $T^{(m+1)} \leftarrow T^{(m)} - (T^{(1)}/2^m)$.
    **else**
        $T^{(m+1)} \leftarrow T^{(m)} - (T^{(1)}/2^m)$ and $T_{\text{record}} \leftarrow T^{(m)}$.
        **end if**
**end for**
**return** $T^* \leftarrow T_{\text{record}}$
```

Algorithm 1

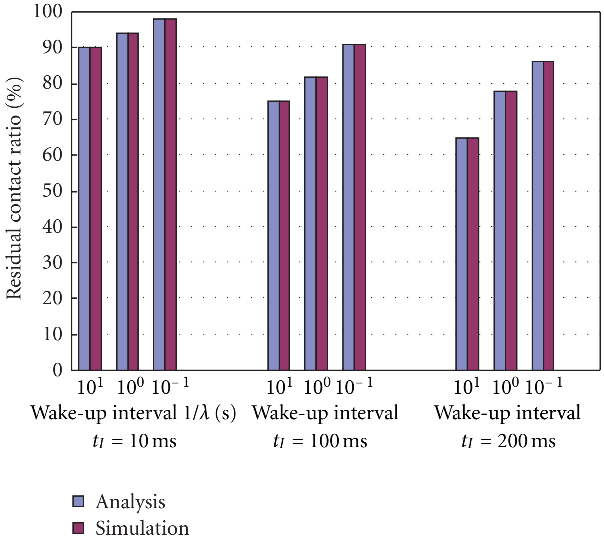


Figure 4: Residual contact ratio as a function of the beacon period for v =1 m/s.

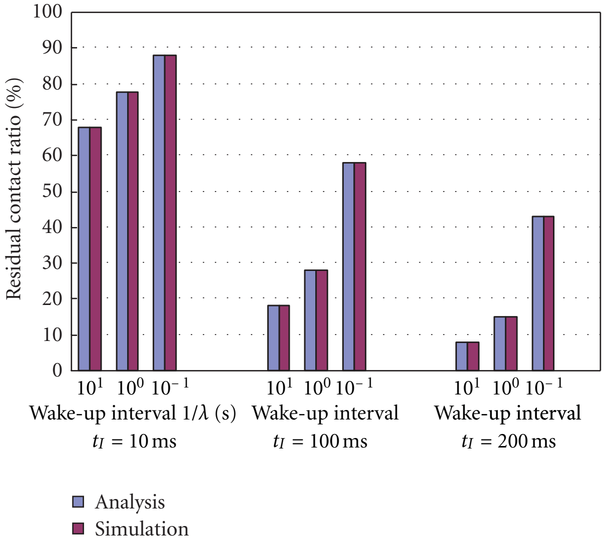


Figure 5: Residual contact ratio as a function of the beacon period for v =8 m/s.

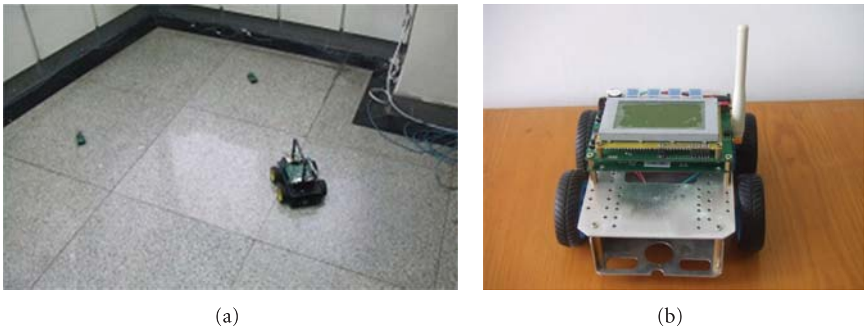
deployed 20 static motes in the random grids and each grid with one sensor. The control program on the TelosB motes was written in the nesC language then compiled and programmed onto the mote using TinyOS 2.x. The static sensors are randomly awake and sense the temperature.

We use the DataTruck [22] as the mobile sink to download data from static sensors. DataTruck is a new sensor node platform designed to support mobility experiments in sensor networks. Although our design is driven by the research requirements of our group, extra effort was taken during the design phase to specify a feature set that is complimentary to existing platforms and can serve multiple aspects of research and education in sensor networks.

While DataTruck enters the contact area, it will send beacon-ID signals to static nodes and set CC2430 in receiving mode. Address resolution will be done while DataTruck receives sensing data correctly; otherwise, it sends a retransmission signal. When the DesNo in the received package is matched with the current DataTruck address, the package will be handled in the local node else it will be drop.

At first, we concern the discovery ratio while using different schemas. Discovery ratio is defined as the average of the ratio between the number of contacts correctly detected by the static sensor and the total number of contacts. We compare the performance of our protocol with other approaches: DIRL [17] and SORA [23], which are two heuristic-based reinforcement learning schemas. From Figure 7, we can see that when the mobility is low, almost all contacts are detected, independent from the adopted schemes, here we denoted our protocol as TIER. The situation is different, however, when the speed is high (i.e., 2 m/s). In this case there is a slight improvement of TIER over other schemas, since it obtains a discovery ratio always over 95%. The reason of this result is our protocol that will adjust the awake probability to satisfy the transmission demand in every round.

In Figure 8, we compare the network lifetime under the different value of speed, where x-axis represents different maximum residual contact time ξ^* in our original Problem (P) and y-axis represents the maximum lifetime for each ξ^*.


(a)

(b)

Figure 6: (a) Experimental setup to collect data from the static sensor nodes with a mobile sink node moving on a uniform grid. (b) The mobile sink node named DataTruck designed by ourselves.

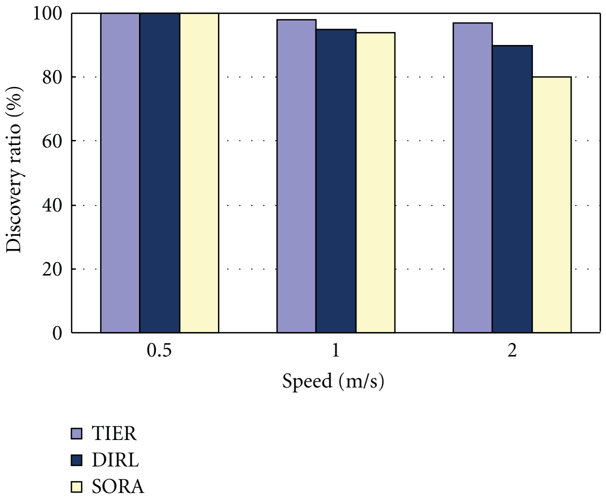


Figure 7: The discovery ratio of different schmas.

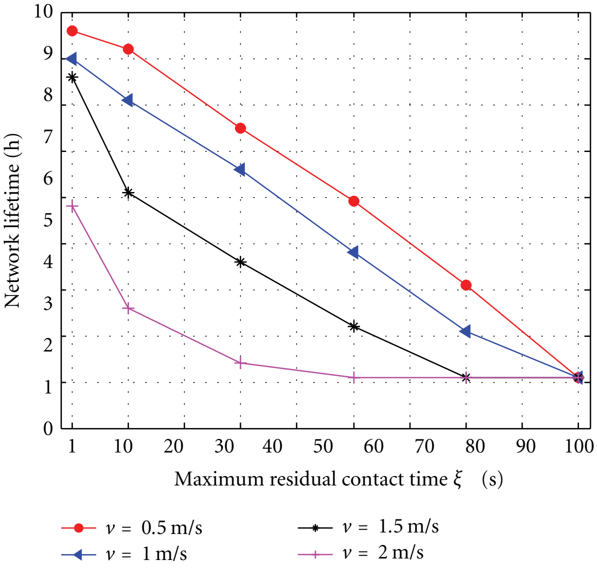


Figure 8: The network lifetime subject to different residual contact time ξ^*.

Here, the communication radius is 100 m, and the maximum value of speed is 2 m/s, so the maximum residual contact time is 100 s. The beacon period t_I is 10 ms.

From Figure 8, we observe the network lifetime is reduced when increasing the speed of MS with the same residual contact time requests. This is because the time MS stay in the contact area is short if the MS has a large speed. Moreover, large speed may delay the MS detection, leading to lower residual contact times. In addition, they may also produce high contact miss ratios, so that the energy spent during discovery is simply wasted, as the sensor node cannot transmit any data.

7. Conclusion

In this paper, we develop a time-independent data collection scheme to reduce the MS discovery delay and to prolong the lifetime of wireless sensor networks employing asynchronous sleep/wake scheduling. Specifically, we study two optimization problems. First, when the wake-up rates of the sensor nodes are given, we develop an efficient and distributed algorithm to maximize the residual contact time for satisfying the data collection task. Second, using a specific definition of the network lifetime, we study the lifetime-maximization problem to optimally control the sleep/wake scheduling policy and discovery policy in order to maximize the network lifetime subject to a lower limit on the expected residual contact time. Our numerical results indicate that the proposed solution can balance the network consumption, especially in a sparse sensor networks. For future work, we plan to generalize our solution to take into account non-Poisson wake-up processes and other lifetime definitions.

Appendix

Proposition A.1. *$C_i(p_i, v, r)$ is a nondecreasing function.*

Proof. Here, we use mathematical introduction to proof this problem. There are two arbitrary awake probability of sensor i, p_a and p_b, and they satisfy $0 \le p_a \le p_b \le 1$. The residual contact time function of p_a and p_b is $C_i(p_a, v, r)$ and $C_i(p_b, v, r)$. Hence, we have

$$\begin{aligned} C_i(p_b,v,r) - C_i(p_a,v,r) & \\ &= \frac{2r}{v} - t_I * \frac{1-(1-p_b)^{2r/v*t_I}}{p_b} \\ &\quad - \frac{2r}{v} + t_I * \frac{1-(1-p_a)^{2r/v*t_I}}{p_a} \\ &= t_I * \frac{p_b - p_b*(1-p_a)^{2r/v*t_I}}{p_a*p_b} \\ &\quad - \frac{p_a + p_a*(1-p_b)^{2r/v*t_I}}{p_a*p_b} \\ &= t_I*\frac{(p_b-p_a)}{p_a*p_b} \\ &\quad + \frac{\left(p_a*(1-p_b)^{2r/v*t_I} - p_b*(1-p_a)^{2r/v*t_I}\right)}{p_a*p_b}. \end{aligned} \tag{A.1}$$

For $p_b - p_a > 0$ and $t_I > 0$, the original problem could be converted to proof $(p_b - p_a) + (p_a * (1-p_b)^{2r/v*t_I} - p_b * (1-p_a)^{2r/v*t_I}) > 0$, that is, $(p_b - p_a) + p_a * (1-p_b)^n - p_b * (1-p_a)^n > 0$, where $n = 2r/(v \times t_I)$. Assuming $L_n = (p_b - p_a) + p_a * (1-p_b)^n - p_b * (1-p_a)^n$, we have the following.

(1) When $n = 1$, obviously $L_1 = 0$, when $n = 2$, $L_2 = (p_b - p_a) * p_b * p_a > 0$.

(2) Assuming $n = k$ $(k = 1, 2, 3, 4, \ldots)$, we have

$$L_k = (p_b - p_a) + p_a * (1-p_b)^k - p_b * (1-p_a)^k \geq 0. \tag{A.2}$$

(3) When $n = k + 1$, we have

$$\begin{aligned} L_{k+1} &= (p_b - p_a) + p_a * (1-p_b)^{k+1} - p_b * (1-p_a)^{k+1} \\ &= (p_b - p_a) + p_a * (1-p_b)^k \\ &\quad * (1-p_b) - p_b * (1-p_a)^k (1-p_a) \\ &\geq (p_b - p_a) + p_a * (1-p_b)^k - p_b \\ &\quad * (1-p_a)^k (1-p_a) \\ &\geq (p_b - p_a) + p_a * (1-p_b)^k - p_b * (1-p_a)^k \\ &= L_K \\ &\geq 0. \end{aligned} \tag{A.3}$$

Using this result, we show that, for any p_a and p_b which satisfy $0 \leq p_a \leq p_b \leq 1$, the inequality $L_n \geq 0$ is correct. This implies when awake probability increases, the residual contact time is also done. □

Acknowledgments

The work is partly supported by China NSF Grants (61073152, 61133006, and 60825205) and the Fundamental Research Funds for the Central Universities (2009B21514) and Changzhou Science Fund grant (CJ20110025).

References

[1] G. Tolle, J. Polastre, R. Szewczyk et al., "Amacroscope in the redwoods," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 51–63, ACM, San Diego, Calif, USA, November 2005.

[2] K. Mayer, K. Ellis, and K. Taylor, "Cattle health mon-itoring using wireless sensor networks," in *Proceedings of the ACM (IASTED '04)*, pp. 375–380, 2004.

[3] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour map matching for event detection in sensor networks," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD '06)*, pp. 145–156, Chicago, Ill, USA, June 2006.

[4] W. Du, L. Fang, and P. Ning, "LAD: localization anomaly detection for wireless sensor networks," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, pp. 874–886, Denver, Colo, USA, April 2005.

[5] A. Arora, P. Dutta, S. Bapat et al., "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[6] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 150–161, San Diego, Calif, USA, November 2003.

[7] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "A hybrid adaptive protocol for reliable data delivery in WSNs with multiple mobile sinks," *Computer Journal*, vol. 54, no. 2, pp. 213–229, 2011.

[8] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Proceedings of the ACM 2nd International Workshop on Information Processing in Sensor Networks(IPSN '03)*, pp. 129–145, April 2003, Palo Alto, Calif, USA.

[9] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proceedings of the IEEE (INFOCOM '11)*, pp. 2246–2254, Hong Kong, April 2011.

[10] J. Luo, Q. Zhang, and D. Wang, "Delay tolerant event collection for underground coal mine using mobile sinks," in *Proceedings of the 17th International Workshop on Quality of Service (IWQoS '09)*, Charleston, SC, USA, July 2009.

[11] Y. Gu, D. Bozdağ, and E. Ekici, "Mobile element based differentiated message delivery in wireless sensor networks," in *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 83–92, Buffalo-Niagara Falls, NY, USA, June 2006.

[12] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, pp. 377–381, San Francisco, Calif, USA, December 2003.

[13] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing 2008 (MobiHoc '08)*, pp. 231–239, Hong Kong, May 2008.

[14] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in ZebraNet," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 227–238, Baltimore, Md, USA, November 2004.

[15] G. Anastasi, M. Conti, and M. Di Francesco, "Reliable and energy-efficient data collection in sparse sensor networks with mobile elements," *Performance Evaluation*, vol. 66, no. 12, pp. 791–810, 2009.

[16] G. Anastasi, M. Conti, and M. Di Francesco, "An analytical study of reliable and energy-efficient data collection in sparse sensor networks with mobile relays," *Lecture Notes in Computer Science*, vol. 5432, pp. 199–215, 2009.

[17] D. M. Francesco, K. Shah, M. Kumar, and G. Anastasi, "An adaptive strategy for energy-efficient data collection in sparse wireless sensor networks," in *Proceedings of the (EWSN '10)*, pp. 322–337, Coimbra, Portugal, February 2010.

[18] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.

[19] G. Anastasi, M. Conti, E. Monaldi, and A. Passarella, "An adaptive data-transfer protocol for sensor networks with data mules," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, Espoo, Finland, June 2007.

[20] Crossbow, "Mica and mica2 wireless measurement system datasheets," 2003.

[21] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 517–526, Santa Clara, Calif, USA, October 2004.

[22] X. Zhang and G. Chen, "Energy-efficient platform designed for SDMA applications in mobile wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '11)*, pp. 2089–2094, Quintana Roo, Mexico, March 2011.

[23] G. Mainland, D. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation (NSDI '05)*, pp. 315–328, Boston, Mass, USA, May 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 417307, 14 pages
doi:10.1155/2012/417307

Research Article

An Extended Virtual Force-Based Approach to Distributed Self-Deployment in Mobile Sensor Networks

Jun Li, Baihai Zhang, Lingguo Cui, and Senchun Chai

School of Automation, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Baihai Zhang, smczhang@bit.edu.cn

Received 16 December 2011; Accepted 21 February 2012

Academic Editor: Mo Li

Virtual physics based approach is one of the major solutions for self-deployment in mobile sensor networks with stochastic distribution of nodes. To overcome the connectivity maintenance and nodes stacking problems in the traditional virtual force algorithm (VFA), an extended virtual force-based approach is investigated to achieve the ideal deployment. In low-R_c VFA, the orientation force is proposed to guarantee the continuous connectivity. While in high-R_c VFA, a judgment of distance force between node and its faraway nodes is considered for preventing node stacking from nonplanar connectivity. Simulation results show that self-deployment by the proposed extended virtual force approach can effectively reach the ideal deployment in the mobile sensor networks with different ratio of communication range to sensing range. Furthermore, it gets better performance in coverage rate, distance uniformity, and connectivity uniformity than prior VFA.

1. Introduction

Wireless sensor networks can be deployed manually or spread randomly over the interested regions for practical applications [1–3]. However, in many working environments, such as remote harsh fields, disaster areas, and toxic gas regions, it is almost impossible to deploy sensors by human beings. In such case, deploying the sensor nodes randomly may not satisfy the requirement of precise placement. Sensor nodes may cluster for stacking in a small region or may distribute sparsely without connectivity guarantee.

A mobile sensor network is composed of a distributed collection of nodes, each of which has communication, sensing, computation, and locomotion capabilities. Mobility of sensor nodes allows more complex application scenarios. With locomotion capabilities, sensor nodes can adjust their positions after stochastic distribution, thus enhancing the coverage and reaching more precise placement.

Many efforts have been put on the algorithms of repositioning sensors in order to obtain a required placement and improve the coverage rate. Previous work on self-deployment issue of mobile sensor networks can be classified into two categories: virtual physics-based approach and computational geometry-based approach. In terms of virtual physics-based strategy [4–7], it models the mobile sensor nodes as the electrons or molecules, and nodes move toward or away each other by their virtual forces or potential fields. However, except the factor of oscillation moving, it does not consider any more crucial problems such as connectivity maintenance and topology control [8, 9]. According to the computational geometry-based approach [10–13], nodes adjust their positions in order to construct a uniform Voronoi diagram or Delaunay triangulation. Nevertheless, decentralized algorithm is too hard to realize because global position information of network should be provided in establishing the Voronoi diagram or Delaunay triangulation.

In this paper, the stability and formation of self-deployment by virtual physic based methods is analyzed. More precisely, the connectivity maintenance problem caused by few neighbors and nodes stacking problem by nonplanar connectivity graph in the existing VFA algorithms are discussed. This paper aims to solve the above problem by introducing an extended virtual force-based approach that can achieve the ideal deployment after self-deployment. The extended virtual force approach can be applied in the mobile sensor networks with different ratio value of communication range and sensing range. In low-R_c VFA, the orientation force is proposed to guarantee the continuous connectivity. While in high-R_c VFA, the judgment of distance force between node and its faraway nodes is adopted in order to prevent node stacking from nonplanar connectivity.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 describes fundamental model of network, ideal deployment, and virtual force approach. In Section 4, the analysis of stability and formation of self-deployment in mobile sensor networks is given. Our proposed extended virtual force approach is introduced in Section 5. Simulation results that illustrate the performance are shown in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

There exists prior works on self-deploying mobile sensor nodes in recent years. Specifically, those closely related to our filed are summarized below.

The concept of self-deployment in mobile sensor networks is derived from dealing with coordination in behavior control of many robots teams [14–16]. Gage [14] has investigated the use of robot swarms to provide blanket, barrier, or sweep coverage of area. According to this taxonomy, the deployment problem of this paper focuses on the blanket coverage. Simmons et al. [15] have calculated desired deployment locations by attempting to minimize overlap in information gain. Blach and Hybinette [16] have suggested the leverage of "attachment sites" that mimic the geometry of crystals. However, most of the behavior based control approaches are centralized and may not perform well on large-scale networks.

In self-deployment approaches, most of strategies are based on the virtual physics. Howard et al. [4] have presented a potential-field-based approach to spread sensor nodes in a target environment. Control force is defined as the negative gradient of the potential. This approach models robots as like electric charges in order to cause uniform deployment into an unknown enclosed area. VFA [5] works in a similar fashion with potential-field-based algorithm. It increases sensor coverage by considering the virtual attractive and repulsive forces exerted on each sensor node by neighbor nodes and obstacles. Heo and Varshney [6] have developed a distributed self-spreading algorithm (DSSA). The force models in DSSA are similar to internuclear repulsion and attraction between molecules. There are also some flaws in these virtual force based self-deployment algorithms. First, they have not considered the connectivity maintenance on the processing of adjusting positions. Moreover, all of these models use fully connected graphs or unit disk graph (UDG), there is a possibility that nonplanar graph may cause significant stacking of sensor nodes [8].

Improved virtual force algorithm (IVFA) and exponential virtual force algorithm (EVFA) [17] have improved traditional VFA to some extent. IVFA sets a maximum movement per iteration in order to prevent nodes from moving out of the region of interest, and incorporates an effective communications distance measure into the force equations to assist the wireless sensor network in reaching a steady state. EVFA provides an exponential force model so that achieve steady state more quickly in mobile sensor networks with a large communication range. However, discontinuous connectivity and coverage holes may appear in IVFA and EVFA because they only assume the effective communication distance is twice as long as the sensing range.

Connectivity-preserved virtual force (CPVF) scheme [18] considers the connectivity preserving by having disconnected sensors move toward the base station to establish connectivity. But CPVF does not conceal the drawback of stacking in large communication range. Self-deployment by density control (SDDC) is presented in [7]. In SDDC, virtual force is decided by density at a sensor node and obstacles. Although compact initial deployment can be spread out, and the stacking problem can be solved, it does not perform well in sparse initial distribution. Distributed robotic macrosensor (DRM) algorithm [8] by virtual spring force control eliminates stacking. The mesh is decidedly planar without intersection of edges by acute-angle test. Extend virtual spring mesh algorithms (EVSM) [19] extends the DRM algorithm with several enhancements such as exploration of unknown areas and obstacle avoidance. In fact, the topology by acute-angle test mesh is Gabriel Graph (GG). However, sensor nodes can only get information from its logical neighbor nodes through GG edges, the uniformity of GG is worse than regular hexagonal deployment structure. Lam and Liu [9] have proposed a self-deployment algorithm named ISOGRID. In ISOGRID, virtual force only exerts on each sensor node by its six closet neighbor nodes, and each node try to move to the vertex of its neighbor nodes' hexagonal placement structure. Though ISOGRID performs very well when the communication range R_c is twice as long as the sensing range R_s, it fails to avoid the stacking in large value of R_c/R_s. Ion-6 [20] is a position unrelated self-deploying method. Ion-6 computes sensor nodes' moving directions and distances independently without a priori position information. However, each sensor has to install a precise antenna array according to this algorithm.

Another commonly used self-deployment approach relies on the use of computational geometry such as Voronoi diagram and Delaunay triangulation. Wang et al. [11] have presented three independent algorithms: the vector-based algorithm (VEC), the Voronoi-based algorithm (VOR), and MiniMax. These algorithms use Voronoi diagrams to partition the coverage field into many subareas and maximize the coverage via pushing or pulling nodes to cover the coverage gaps on virtual force. In computational geometry-based strategy, stacking can be limited. However, the Voronoi diagram is a global structure, in which all Voronoi vertices and cells can only be obtained when the global location information with all other nodes in the network is known.

3. Preliminaries

3.1. Network Model and Assumptions. This paper focuses on the self-deployment issue in 2-dimensional plane and

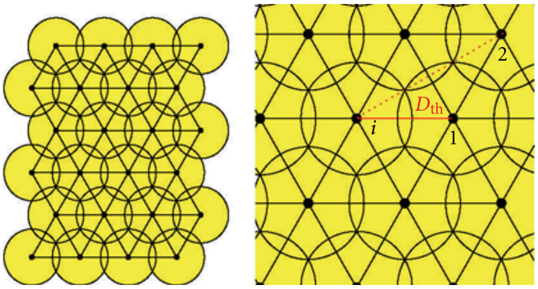


Figure 1: Ideal deployment for full coverage.

leverages Euclidian plane $\mathbb{R}^2$ to model the coordinate system. Node's position is represented by its coordinate. The position of node i is described as $S_i(x_i, y_i)$. The distance between node i and node j is defined as Euclidean distance d_{ij}. The initial deployment is a stochastic placement in unknown distribution. It assumes that each sensor node can learn its own exact position via GPS or other localization technology, and sensor node is capable of receiving its neighbors' message without losing data. At the same time, it can also calculate or measure the relative distances and orientations between them. The sensor node's communication and sensing models are modeled circular discs. All sensors have the identical communication range R_c and sensing range R_s, respectively. R_c is larger than R_s.

3.2. Ideal Deployment. The essential aim of self-deployment is to make the sensor nodes move from their original positions to new positions in order to form the ideal deployment layout. Optimal deployment patterns for k-coverage with l-connectivity maintenance have been studied [21–23]. Bai et al. [24] have improved k-coverage of mobile sensor networks using improve PSO algorithm. In this paper, we study the 1-coverage self-deployment problem. An ideal deployment grid structure for 1-coverage is show in Figure 1. Equilateral triangle grid (hexagonal placement structure) has the smallest overlapping area, hence this deployment requires the least number of sensors for area full coverage [25]. There is no coverage "hole" exists in an ideal deployment sensor network. The ideal distance D_{th} between sensor node and its nearest neighbor should be $\sqrt{3}R_s$, and the angle formed by one node and its two adjoined neighbors should be $\pi/3$ [26]. If $R_c/R_s > 3$, the distance between sensor node and its physical neighbors may be larger than D_{th}.

3.3. Virtual Force-Based Approach. All of the virtual physics approaches for self-deployment are similar with the framework of virtual force, which combines the ideas of potential field with circle packing [4] by modeling the sensor node to be a particle in the potential field. The potential field exerts forces on the nodes nearby. For node i and j, it is useful to write the force as the negative gradient of the potential field. We can construct a potential function $\mathbf{V}_{ij}$. So the control input of a node is the force as

$$\mathbf{F}_{ij} = -\nabla \mathbf{V}_{ij}. \tag{1}$$

Sensor nodes move towards the required placement by these virtual forces. The force may be either attractive force or repulsive force. If two sensor nodes are placed closer than the threshold distance (ideal distance D_{th}), repulsive forces are exerted on each other. Alternatively, attractive forces are exerted if two sensor nodes are farther apart than the threshold distance. The attractive force is to keep a certain density of sensor nodes without coverage holes, and the repulsive force is to make sensors sparse enough without too much redundant coverage.

The law of updating position exploits either step method by iteration [17] or sampling time method [8, 19]. Normally, both of these methods will achieve the same results. In this paper, we choose sampling time method for analyzing. The control law for each sensor node is

$$\ddot{\mathbf{x}}_i = \sum_{j \in G_i} \mathbf{F}_{ij} - k_d \dot{\mathbf{x}}_i, \tag{2}$$

where $\ddot{\mathbf{x}}_i$ is node's acceleration, $\dot{\mathbf{x}}_i$ is node's velocity, G_i is the set of node's neighbors (in original VFA, G_i is the set of all nodes except i), k_d is the positive damping coefficient.

According to the traditional VFA, the force law is given as follows:

$$\overrightarrow{F_{ij}} = \begin{cases} \left(\omega_A\left(d_{ij} - D_{\mathrm{th}}\right), \alpha_{ij}\right), & \text{if } d_{ij} > D_{\mathrm{th}}, \\ 0, & \text{if } d_{ij} = D_{\mathrm{th}}, \\ \left(\omega_R d_{ij}^{-1}, \alpha_{ij} + \pi\right), & \text{if } d_{ij} < D_{\mathrm{th}}, \end{cases} \tag{3}$$

where ω_A and ω_R represent the virtual force attractive and repulsive coefficients, respectively. And d_{ij} is the Euclidean distance between sensor nodes i and j, α_{ij} is the orientation of the line segment from nodes i to j.

The sensor moves to its new position under the control law in (2). The total force $\mathbf{F}_i$ exerted on node i is decided by the summation of all forces contributed by its all neighbors:

$$\mathbf{F}_i = \sum_{j \in G_i} \mathbf{F}_{ij}. \tag{4}$$

4. Analysis of Virtual Physics-Based Approach for Self-Deployment

4.1. Stability Analysis. Although small oscillation can be observed in the processing of self-deployment under virtual physics-based approach, all nodes will stop moving finally. In fact, there is always damping effect acting against the motion of each sensor node. It causes the reduction in kinetic energy. Potential energy is conservative because the drive force is defined as the negative gradient of it. Thus, the total energy cannot be increased, and kinetic energy must eventually approach to 0. We can proof the stability with Lyapunov stability theory.

Theorem 1. *In virtual force-based approach for self-deployment, any node will eventually converge to a steady state.*

Proof. Let $\mathbf{x}_i$ be the position vector of node i. The control input of a node is the force $\overrightarrow{F_{ij}}$ exerted on node i by node j.

The virtual potential field is constructed as $\mathbf{V}_{ij}$. From (1) and (4), we obtain the total force from each node which can be described as

$$\mathbf{F}_i = \sum_{j\in G_i} \mathbf{F}_{ij} = \sum_{j\in G_i} -\nabla \mathbf{V}_{ij}. \tag{5}$$

Formally, we consider the following energy function ($\mathbf{\Phi}$) that combines kinetic energy with potential energy as Lyapunov function:

$$\mathbf{\Phi} = \sum_{i=1}^{N} \frac{1}{2}\dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in G_i} \mathbf{V}_{ij}. \tag{6}$$

Because of the symmetry of $\mathbf{V}_{ij}$ and $\mathbf{V}_{ji}$, and $\dot{\mathbf{x}}_i = -\dot{\mathbf{x}}_j$ on the orientation from node i to j. We get the time derivative of $\mathbf{V}_{ij}$ as follows:

$$\begin{aligned}\frac{d}{dt}\mathbf{V}_{ij} &= \frac{\partial \mathbf{V}_{ij}}{\partial \mathbf{x}_i} \cdot \frac{d\mathbf{x}_i}{dt} + \frac{\partial \mathbf{V}_{ij}}{\partial \mathbf{x}_j} \cdot \frac{d\mathbf{x}_j}{dt} \\ &= \nabla \mathbf{V}_{ij} \cdot \dot{\mathbf{x}}_i - \nabla \mathbf{V}_{ij} \cdot \dot{\mathbf{x}}_j \\ &= \nabla \mathbf{V}_{ij} \cdot \left(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j\right) \\ &= 2\nabla \mathbf{V}_{ij} \cdot \dot{\mathbf{x}}_i. \end{aligned} \tag{7}$$

So the derivative of the energy function $\mathbf{\Phi}$ is transferred into:

$$\begin{aligned}\dot{\mathbf{\Phi}} &= \sum_{i=1}^{N} \frac{1}{2}\left(\dot{\mathbf{x}}_i^T \ddot{\mathbf{x}}_i + \ddot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i\right) + \sum_{i=1}^{N}\sum_{j\in G_i} \nabla \mathbf{V}_{ij}\dot{\mathbf{x}}_i^T \\ &= \sum_{i=1}^{N} \dot{\mathbf{x}}_i^T \left(\ddot{\mathbf{x}}_i + \sum_{j\in G_i} \nabla \mathbf{V}_{ij}\right). \end{aligned} \tag{8}$$

According to (2), (8) can be processed as follows:

$$\begin{aligned}\dot{\mathbf{\Phi}} &= \sum_{i=1}^{N} \dot{\mathbf{x}}_i^T \left(\ddot{\mathbf{x}}_i + \sum_{j\in G_i} \nabla \mathbf{V}_{ij}\right) \\ &= \sum_{i=1}^{N} \dot{\mathbf{x}}_i^T \left(\sum_{j\in G_i} \mathbf{F}_{ij} - k_d \dot{\mathbf{x}}_i + \sum_{j\in G_i} \nabla \mathbf{V}_{ij}\right) \\ &= \sum_{i=1}^{N} \dot{\mathbf{x}}_i^T \left(\sum_{j\in G_i} -\nabla \mathbf{V}_{ij} - k_d \dot{\mathbf{x}}_i + \sum_{j\in G_i} \nabla \mathbf{V}_{ij}\right) \\ &= -k_d \sum_{i=1}^{N} \dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i. \end{aligned} \tag{9}$$

For the positive damping coefficient k_d, $\dot{\mathbf{\Phi}}$ is the seminegative definite. Therefore, according to Lyapunov stability theory, the system with virtual force input control law is asymptotically stable. Equality $\dot{\mathbf{\Phi}} = 0$ only when $\dot{\mathbf{x}}_i = 0$. So any node will eventually converge to a stationary state. □

Figure 2 shows the variations of coordinate for movement from self-deployment by 3 nodes with initial positions of S_1 (50, 55), S_2 (55, 58), and S_3 (45, 60). From the graph, it can be seen that each sensor node could eventually converge to a steady state within 1 minute.

Note that the domain of the Lyapunov function $\mathbf{\Phi}$ is not the positions $\mathbf{x}$, but the velocities $\dot{\mathbf{x}}$ of mobile sensor node. Although every node may reach a stable state, it does not mean that the mobile sensor network can achieve the ideal deployment. It is possible for some potential energy to exist even in a static state. We will further discuss it in Section 4.2.

4.2. Formation Analysis. The goal of self-deployment is to make the formation of placement as the ideal equilateral triangle grid. However, Chen et al. [17] and Shucker and Bennett [8] have noticed that attractive force always exists whenever the distance between two sensors is larger than threshold D_{th} by some examples. So VFA cannot always guarantee that the distance between sensors is reached at threshold D_{th}. Moreover, VFA shows significant stacking of nodes in full-connectivity graph or nonplanar connectivity graph. Here, we analyze the drawbacks of VFA.

Theorem 2. *In mobile sensor networks with ideal deployment, all virtual forces exerted on any pair of nodes are equal to 0.*

Proof. From Section 3.2, we know that the distance between each pair of nodes is larger than or equal to D_{th} in an ideal deployment. Therefore, every force exerted on each pair of nodes is either 0 or attractive force. An ideal deployment is a stable deployment, and the total virtual force from neighbors of each node is equal to 0.

We assume there are N sensor nodes in the network topology graph, and there are M ($M \geqslant 1$) virtual attractive forces across M connectivity edges $E(i, j)$, while other forces equal to 0.

For $\forall E(a, b) \in E(i, j) > D_{\text{th}}$, the force through the edge $E(a, b)$ is toward each other for node a and node b. We consider $\overrightarrow{ab}$ as the orientation of x-axis and the position of node a as the coordinate origin. Then we build the Cartesian coordinate system, as shown in Figure 3.

We suppose there are P_a neighbors on the left half plane of node a, while Q_a neighbors on the right half plane. As the total forces on a static node is 0, the sum of virtual forces of node a is

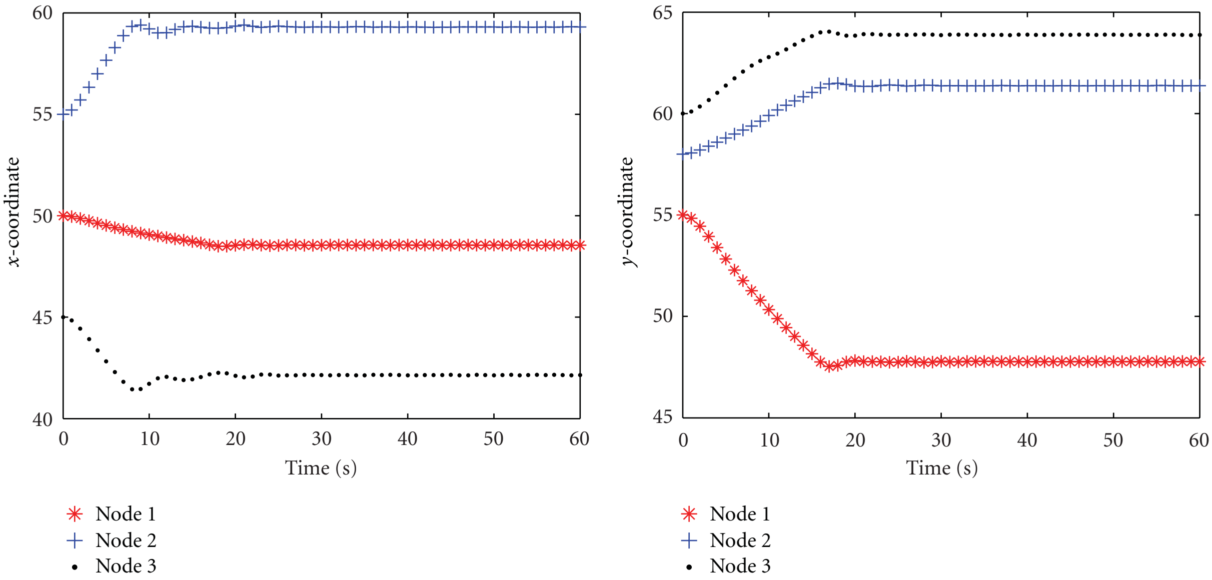
$$\overrightarrow{F_a} = \overrightarrow{F_{ab}} + \sum_{i=1}^{P_a} \overrightarrow{F_{ai}} + \sum_{j=1}^{Q_a} \overrightarrow{F_{aj}} = 0. \tag{10}$$

For each neighbor node of a, the force on $\overrightarrow{ab}$ is

$$\begin{aligned}\overrightarrow{F_{a(\overrightarrow{ab})}} &= \overrightarrow{F_{ab}} + \sum_{i=1}^{P_a}\left(\left|\overrightarrow{F_{ai}}\right| \cos\angle iab\right)\mathbf{u}_{ab} \\ &\quad + \sum_{j=1}^{Q_a}\left(\left|\overrightarrow{F_{aj}}\right| \cos\angle jab\right)\mathbf{u}_{ab} = 0, \end{aligned} \tag{11}$$

where $\mathbf{u}_{ab}$ is the unit vector from node a to node b. If $P_a = 0$, $\cos\angle jab \geq 0$, so

$$F_{a(\overrightarrow{ab})} = \left|\overrightarrow{F_{ab}}\right| + \sum_{j=1}^{Q_a}\left|\overrightarrow{F_{aj}}\right| \cos\angle jab > 0. \tag{12}$$



(a) x-coordinate

(b) y-coordinate

Figure 2: Convergence of VFA for 3 nodes with sensing range 10 m.

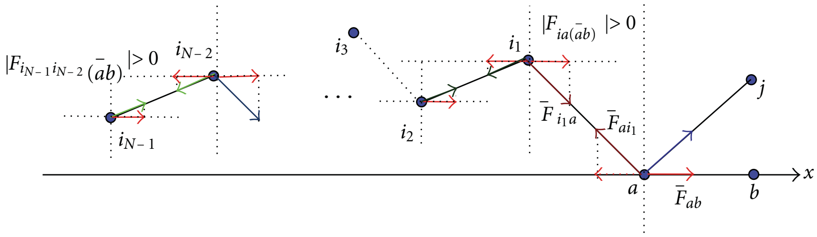


Figure 3: Virtual force for each node should equal to 0.

It is unstable. So, there are P_a ($P_a \geqslant 1$) neighbor nodes on the left half plane of node a, and these P_a nodes satisfing their x-coordinates are negative. We define node a as the top node, and these P_a nodes as level 1 nodes.

As shown in Figure 3, in order to achieve the stable state of these P_a nodes, for the same analysis on node a, there are P_1 ($P_1 \geqslant P_a$) level 2 neighbor nodes on the left half plane of these p_1 nodes. For the same reason, there are P_{N-1} ($P_{N-1} \geqslant P_{N-2}$) level N neighbor nodes on the left half plane of P_{N-2} level $N-1$ nodes. So, in order to keep the equilibrium of force on orientation $\overrightarrow{ab}$, the number of nodes we need is at least:

$$2 + P_a + P_1 + P_2 + \cdots + P_{N-1} + \cdots \geq 2 + P_a N > N. \quad (13)$$

It is obvious that the network cannot reach the ideal deployment if there exists attractive force. Therefore, we conclude that all of the virtual forces across connectivity edges in the topology graph for ideal deployment are equal to 0. □

Lemma 3. *If the connectivity graph of mobile sensor network is nonplanar, the network deployment cannot reach the ideal deployment.*

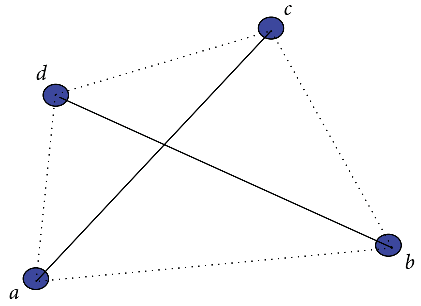
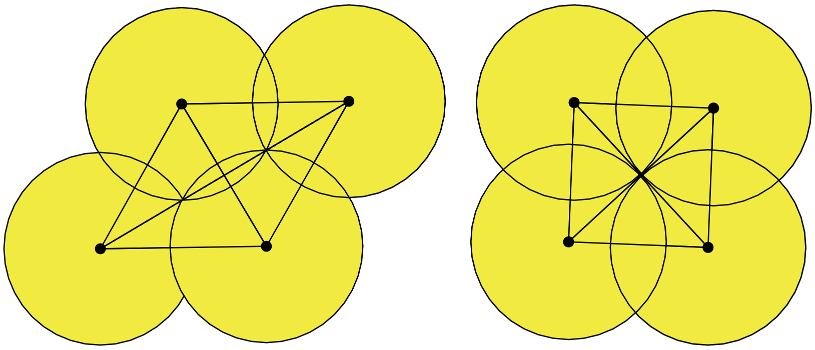


Figure 4: Nonplanar connectivity graph.

Proof. Figure 4 shows a nonplanar connectivity graph of sensor nodes $\{a, b, c, d\}$ with two intersections of connectivity edges $E(a, c)$ and $E(b, d)$.

From Theorem 1, in an ideal deployment, all the connectivity graph edges should equal to ideal distance D_{th}.

So $\mathbf{ac} = \mathbf{bd} = D_{\text{th}}$. For quadrilateral **abcd**, there is at least one angle larger than or equal to $\pi/2$. In right-angled triangle or obtuse-angled triangle, the other two edges of triangle are shorter than the longest angle's diagonal whose length



(a) Ideal deployment

(b) Self-deployment by VFA

Figure 5: Nonplanar connectivity graph cannot reach ideal deployment.

is D_{th}. However, in Section 3.2 we know that in an ideal deployment, the distance between each pair of two nodes is larger than or equal to D_{th}.

Thus, the network deployment cannot reach the ideal deployment if there is (are) intersection(s) of edges in connectivity graph. □

As shown in Figure 5, the connectivity graph of four nodes is nonplanar. Self-deployment by traditional VFA cannot reach the ideal deployment. In fact, the nonplanar graph exhibits more significant stacking under self-deployment if there are more intersections of connectivity edge. So in mobile sensor networks with a large value of R_c/R_s, compact deployment will be reached after self-deployment via VFA. Some algorithms assume the communication range R_c is twice as long as the sensing range R_s in order to decrease the intersections of connectivity edges [9, 17]. However, it may cause coverage hole suffer from discontinuous connectivity. Even if the planar connectivity graph such as MST, GG, RNG, and TDG can be used in avoiding the intersections [8, 19], they cannot guarantee the logical neighbor for each node is 6 for construct the hexagonal formation.

In the following sections, we will make some improvements on virtual physics-based approach in order to solve the above-mentioned problems.

5. Extended Virtual Force Approach for Mobile Sensor Networks

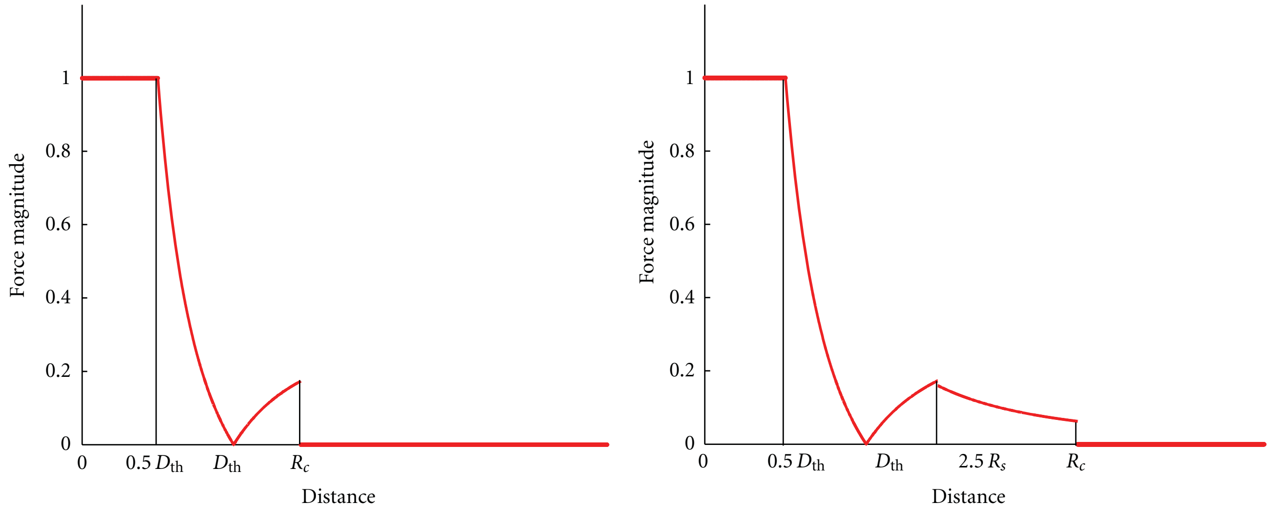
The extended virtual force algorithm is a self-deployment scheme with some novel features which can overcome the drawbacks of traditional VFA. In mobile sensor networks with small value of R_c/R_s, the stacking problem caused by nonplanar connectivity graph is not serious. Thus, connectivity maintenance should be more important than stacking avoidance. In this paper, the algorithm is defined as low-R_c VFA. On the other hand, in mobile sensor networks with large value of R_c/R_s, the stacking problem caused by nonplanar connectivity graph is quite obvious. The algorithm should eliminate the sensor nodes' stacking. We define the algorithm as high-R_c VFA.

5.1. Low-R_c VFA. As shown in Figure 1, under the ideal deployment, the nonplanar connectivity graph can be formed, when R_c/R_s is larger than 3. So in this paper, we consider the mobile sensor network as a low-R_c network, while the value of R_c/R_s is equal to or smaller than 2.5. In low-R_c VFA, distance force pushes nodes away from neighbors if their distances are less than D_{th} while moves nodes towards to neighbors if their distances are larger than D_{th}, and orientation force is added to improve the force model in order to keep the continuous connectivity. Chen et al. [17] have shown that exponential force model can achieve fast convergence. The distance force model is shown in (14) and Figure 6(a):

$$\overrightarrow{F_{ij(d)}} = \begin{cases} \alpha\left(d_{ij}{}^{-\beta} - D_{\text{th}}{}^{-\beta}\right)\mathbf{u}_{ij}, & \text{if } D_{\text{th}} < d_{ij} \le R_c, \\ 0, & \text{if } d_{ij} = D_{\text{th}}, \\ \left|\alpha\left(d_{ij}{}^{-\beta} - D_{\text{th}}{}^{-\beta}\right)\right|\mathbf{u}_{ji}, & \text{if } 0.5D_{\text{th}} < d_{ij} < D_{\text{th}}, \\ 1\mathbf{u}_{ji}, & \text{if } d_{ij} \le 0.5D_{\text{th}}, \end{cases} \tag{14}$$

where $\mathbf{u}_{ij}$ is the unit vector from node i to node j, α and β are constants which can be adjusted according to different situations. Normally, the value of α is equal to $0.25{D_{\text{th}}}^2$, while β is equal to 2. For pair of nodes which distance is less than $0.5D_{\text{th}}$, the limit magnitude of virtual force exerted on them is set as 1 in order to avoid the high acceleration and velocity by the control input saturation.

The distance force can reduce the connectivity degree to be less than or equal to 6 if the initial connectivity degree is larger than 6. The orientation forces are only exerted on the nodes whose neighbors are less than 6. The aim of orientation force is tried to keep the angle formed by one node and its two adjoined neighbors equal to $\pi/3$. Then mobile sensor networks will reach reliable connectivity and eliminate coverage holes caused by continuous connectivity. For node i with connectivity degree m ranging from 2 to 6, we calculate all angles between node i and its two adjoining



(a) Low-R_c VFA

(b) High-R_c VFA

FIGURE 6: Force models.

neighbors as $S(\theta) = \{\theta(i, j, p)\}$. The orientation force model is

$$\overrightarrow{F_{ij(o)}} = \begin{cases} \omega_o\left(\theta(i,j,j_c) - \dfrac{\pi}{3}\right)\mathbf{u}_{jj_c}, \\ \quad \text{if } \theta(i,j,j_{ac}) \text{ is the largest angle}, \\ \omega_o\left(\theta(i,j,j_{ac}) - \dfrac{\pi}{3}\right)\mathbf{u}_{jj_{ac}}, \\ \quad \text{if } \theta(i,j,j_c) \text{ is the largest angle}, \\ \omega_o\left\{\left|\left(\theta(i,j,j_c) - \dfrac{\pi}{3}\right)\right|\mathbf{u}_{jj_c} \right. \\ \quad \left. +\left|\left(\theta(i,j,j_{ac}) - \dfrac{\pi}{3}\right)\right|\mathbf{u}_{jj_{ac}}\right\}, \\ \quad \text{otherwise}, \end{cases} \tag{15}$$

where node j_c is the neighbor of node i on the clockwise orientation under node j, and j_{ac} is the neighbor on the anticlockwise orientation.

Then, in low-R_c VFA, the force exerted on a node can be calculated by

$$\overrightarrow{F_{ij}} = \begin{cases} \overrightarrow{F_{ij(d)}}, & \text{if } m > 6 \text{ or } m = 1, \\ \overrightarrow{F_{ij(d)}} + \sum\limits_{j \in G_i} \overrightarrow{F_{ji(o)}}, & \text{if } 2 \le m \le 6. \end{cases} \tag{16}$$

5.2. High-R_c VFA. We consider the mobile sensor network as a high-R_c networks if the R_c/R_s is larger than 2.5. In high-R_c VFA, connectivity maintenance can be guaranteed. Thus, only distance forces are exerted among the nodes and their neighbors. We obtain the virtual force model in high-R_c VFA as follows:

$$\overrightarrow{F_{ij}} = \begin{cases} \alpha d_{ij}{}^{-\beta}\mathbf{u}_{ij}, & \text{if } D_{\text{th}} > 2.5R_s, \\ \alpha\left(d_{ij}{}^{-\beta} - D_{\text{th}}{}^{-\beta}\right)\mathbf{u}_{ij}, & \text{if } D_{\text{th}} < d_{ij} \le 2.5R_s, \\ 0, & \text{if } d_{ij} = D_{\text{th}} \text{ or } d_{ij} > R_c, \\ \left|\alpha\left(d_{ij}{}^{-\beta} - D_{\text{th}}{}^{-\beta}\right)\right|\mathbf{u}_{ji}, & \text{if } 0.5D_{\text{th}} < d_{ij} < D_{\text{th}}, \\ 1\mathbf{u}_{ji}, & \text{if } d_{ij} \le 0.5D_{\text{th}}. \end{cases} \tag{17}$$

In high-R_c VFA, virtual force varies exponentially with distance between each pair of mobile sensor nodes. As shown in (17) and Figure 6(b), the virtual forces between node and its faraway neighbors are decreasing with the farther distance. This feature can diminish nodes stacking effectively.

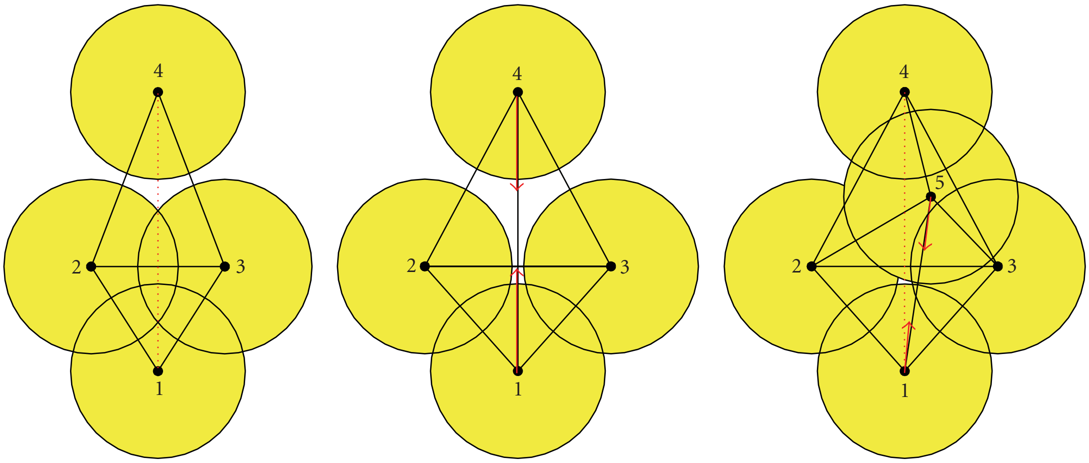
There always exist virtual attractive forces between nodes and its faraway neighbors in the traditional VFA. Therefore, stacking possibly occurs in high-R_c mobile sensor networks. In high-R_c VFA, besides the exponential force model, we consider a judgment of distance force between node and its faraway nodes to prevent node stacking from nonplanar connectivity. The scheme of judgment is described as Algorithm 1.

As shown in Figure 7, $d_{14} > D_{\text{th}}$. $\angle 213 < \pi/3$ in (a), while $\angle 213 > \pi/3$ in (b). Node 2 and node 3 are the neighbors of node 1, where $d_{1j} < D_{\text{th}}$, and also are the neighbors of node 4, where $d_{4j} \le R_c$. According to Algorithm 1, $\overrightarrow{F_{14}} = 0$ in the deployment of (a), while $\overrightarrow{F_{14}}$ in (b) can be calculated by (17). And in Figure 7(c), node 5 is located in quadrilateral 1234, so $\overrightarrow{F_{14}} = 0$ in this deployment.

Both in low-R_c VFA and high-R_c VFA, the total force exerted on node i is decided by adding all forces contributed

```
$G_i$: the set of neighbors of node $i$ satisfy $d_{ij} \le R_c$;
$S_i$: the set of neighbors of node $i$ satisfy $d_{ij} < D_{th}$;
$N_i$: the number of nodes belongs to $G_i$;
$n_i$: the number of nodes belongs to $S_i$;
for $1 \le j \le N_i$
    if $d_{ij} \le D_{th}$
        $\overrightarrow{F_{ij}}$ is calculated by (17)
    end if
    if $d_{ij} > D_{th}$
        Find the neighbors which belong to both $S_i$ and $G_j$. $m$ is the number of nodes satisfy $d_{jm} < R_c$ and belong to $S_i$;
        if $m < 2$
            $\overrightarrow{F_{ij}} = 0$
        end if
        if $m \ge 2$
            Compute the angle $\theta$ formed by node $i$, $j_c$, and $j_{ac}$, where $\overrightarrow{ij_c}$ is on the clockwise side of $\overrightarrow{ij}$, while $\overrightarrow{ij_{ac}}$ is on the anti-clockwise side. Node $j_c$, and $j_{ac}$ belongs to $S_i$ and $G_j$.
            if $\theta \le \pi/3$ and there are not any nodes located in quadrilateral $\mathbf{ij\ j_c\ j_{ac}}$
                $\overrightarrow{F_{ij}} = 0$
            else
                $\overrightarrow{F_{ij}}$ is calculated by (17)
            end if
        end if
    end if
end for
```

Algorithm 1: The scheme of virtual force between node i and its neighbor nodes.



(a) $\theta < \pi/3$ (b) $\theta > \pi/3$ (c) $\theta > \pi/3$

Figure 7: Virtual force between node and its faraway node.

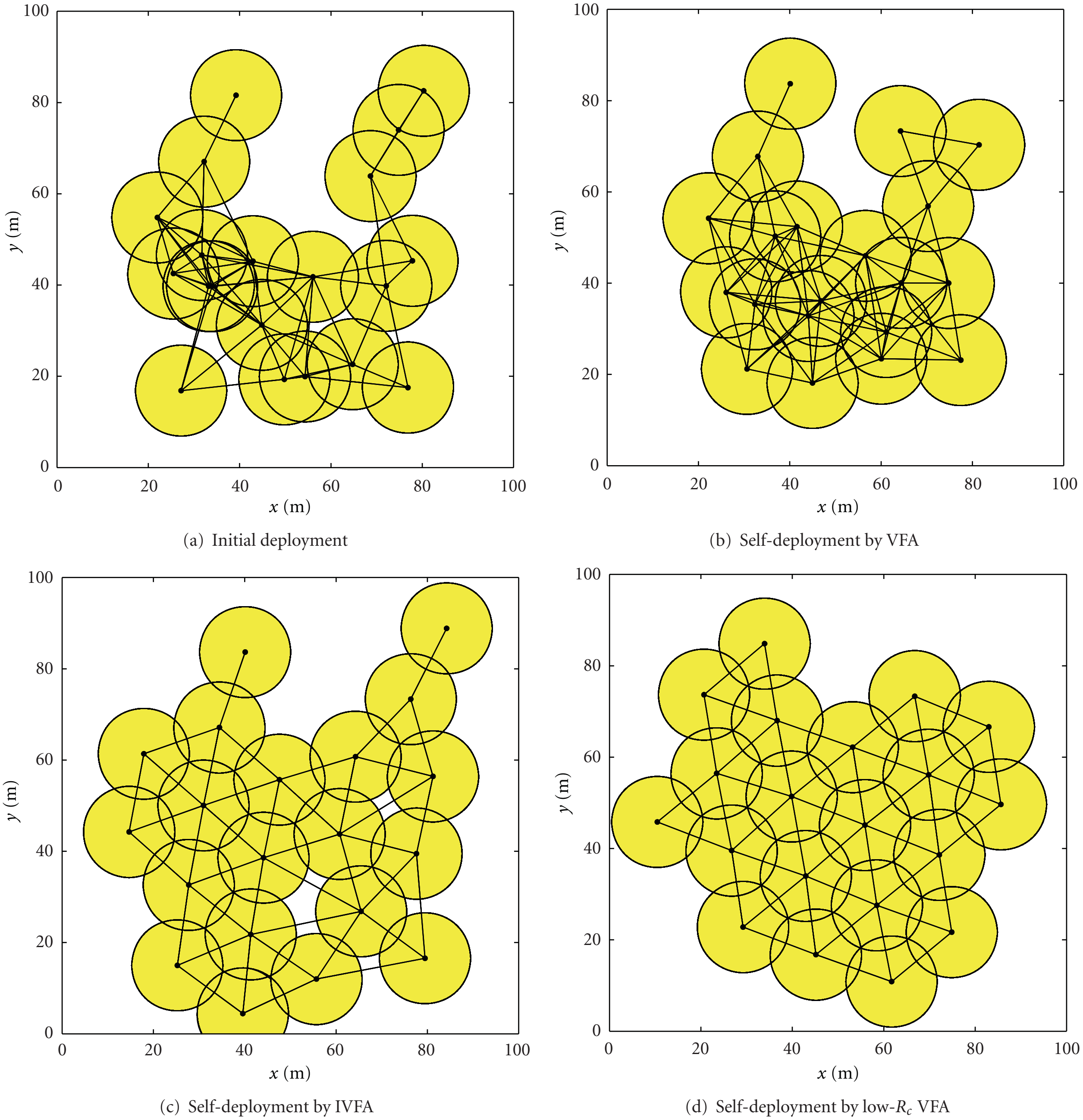
by its neighbors as (4). Each node is driven for moving by the control law in (2).

5.3. Performance Evaluation

5.3.1. Coverage Rate. The coverage rate is a measure of the coverage quality for sensor networks. It was originally introduced by Gage [14]. In blanket coverage problem, coverage is defined by the ratio of the union of covered areas of each node to the complete area of interest. In this paper, the covered area of each node is defined as the circular area with sensing range R_s. The value of coverage rate C is as

$$C = \frac{\bigcup_{i=1}^{n} A_i}{A}, \quad (18)$$

where A_i is the area covered by the node i, n is the total number of mobile sensor nodes, and A is the total area of ROI.



(a) Initial deployment

(b) Self-deployment by VFA

(c) Self-deployment by IVFA

(d) Self-deployment by low-R_c VFA

FIGURE 8: Self-deployment results in $R_c/R_s = 2.5$.

5.3.2. Uniformity of Distance. Under the ideal deployment of sensor nodes, the formation effectiveness of a deployment can be measured by distance uniformity and connectivity uniformity. Distance uniformity is defined as the average of the local standard deviation of the distances between neighboring nodes [6]:

$$U_d = \frac{1}{n}\sum_{i=1}^{n} U_{di},$$
$$U_{di} = \left(\frac{1}{n_i}\sum_{j\in G_i}^{n}\left(d_{ij} - D_{\text{th}}\right)^2\right)^{1/2}, \qquad (19)$$

where U_{di} is the local distance uniformity, G_i is the set of node's neighbors, and n_i is the number of neighbors for node i. However, (19) is inappropriate for showing the uniformity in nonplanar connectivity graph. Here, we define U_i as

$$U_{di} = \left(\frac{1}{n_i}\sum_{j\in(G_i, d(i,j)<2.5R_s)}^{n}\left(d_{ij} - D_{\text{th}}\right)^2\right)^{1/2}. \qquad (20)$$

5.3.3. Uniformity of Connectivity. Under the ideal deployment of sensor nodes, each node has the same number of neighbors except the boundary nodes. Uniformity of connectivity is defined as:

$$U_c = \left(\frac{1}{n} \sum_{i=1}^{n} (\mathrm{CN}_i - \mathrm{CN}_{\mathrm{th}})^2 \right)^{1/2}, \qquad (21)$$

where CN_i is the number of neighbors for node i, where $d_{ij} \leq 2.5R_s$, and $\mathrm{CN}_{\mathrm{th}}$ is the ideal number of neighbors. Particularly, $\mathrm{CN}_{\mathrm{th}}$ is equal to 6 in this paper.

6. Simulation Results

In this section, we present simulation results for the extended virtual force approach including low-R_c VFA and high-R_c VFA.

6.1. Performance of Low-R_c VFA. Figure 8 shows the self-deployment results from stochastic initial distribution of 20 mobile sensor nodes in a 100 m × 100 m area. In this simulation, the communication range is 25 m, while the sensing range is 10 m. The sampling time is 0.1 sec, and the damping coefficient k_d is 0.5. Figure 8(a) is the initial deployment and its unit disk graph (UDG). Figure 8(b) shows the self-deployment result by VFA. The final deployment is compressed because the nonplanar connectivity graph. In Figure 8(c), IVFA shows the better result than VFA for avoiding node stacking caused by intersection of connectivity edges. However, the continuous connectivity is not perfect because of significant coverage holes in IVFA. Self-deployment by low-R_c VFA is shown in Figure 8(d). We can see that an ideal deployment is achieved.

Figure 9 shows the performance comparison of coverage rate among VFA, IVFA, and low-R_c VFA. IVFA can reach the highest coverage because the discontinuous connectivity. For the stacking of nodes, VFA cannot guarantee the enhancement of the coverage rate. All three of these algorithms have achieved a convergence state in about 30 sec. Moreover, the performance comparison of distance uniformity among VFA, IVFA, and low-R_c VFA is shown in Figure 10. It is noticed that after self-deployment with 60 s, the distance uniformity of VFA, IVFA, and low-R_c VFA are 3.01, 1.42, and 0.13 respectively. The performance of distance uniformity shows that low-R_c VFA has reached the most regular ideal deployment.

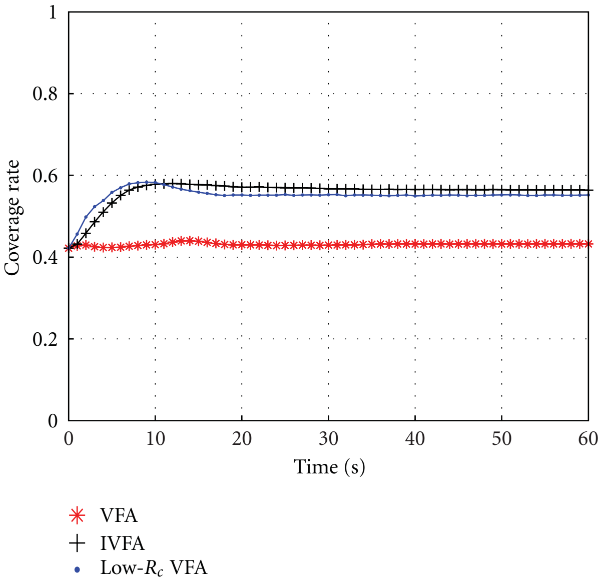


Figure 9: Coverage rate comparison among VFA, IVFA, and low-R_c VFA.

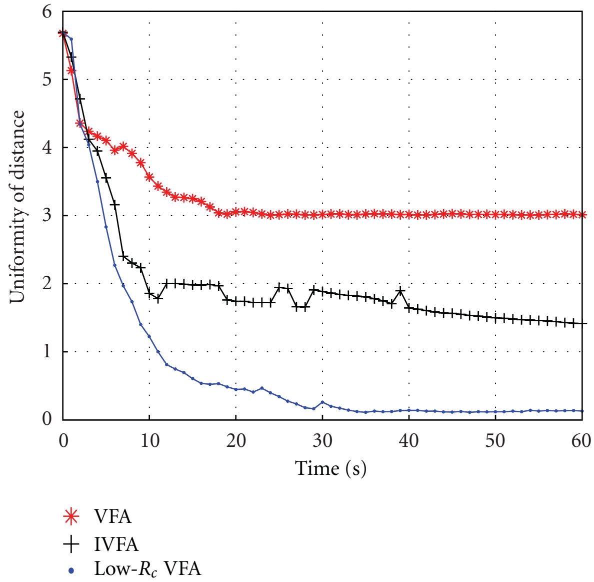
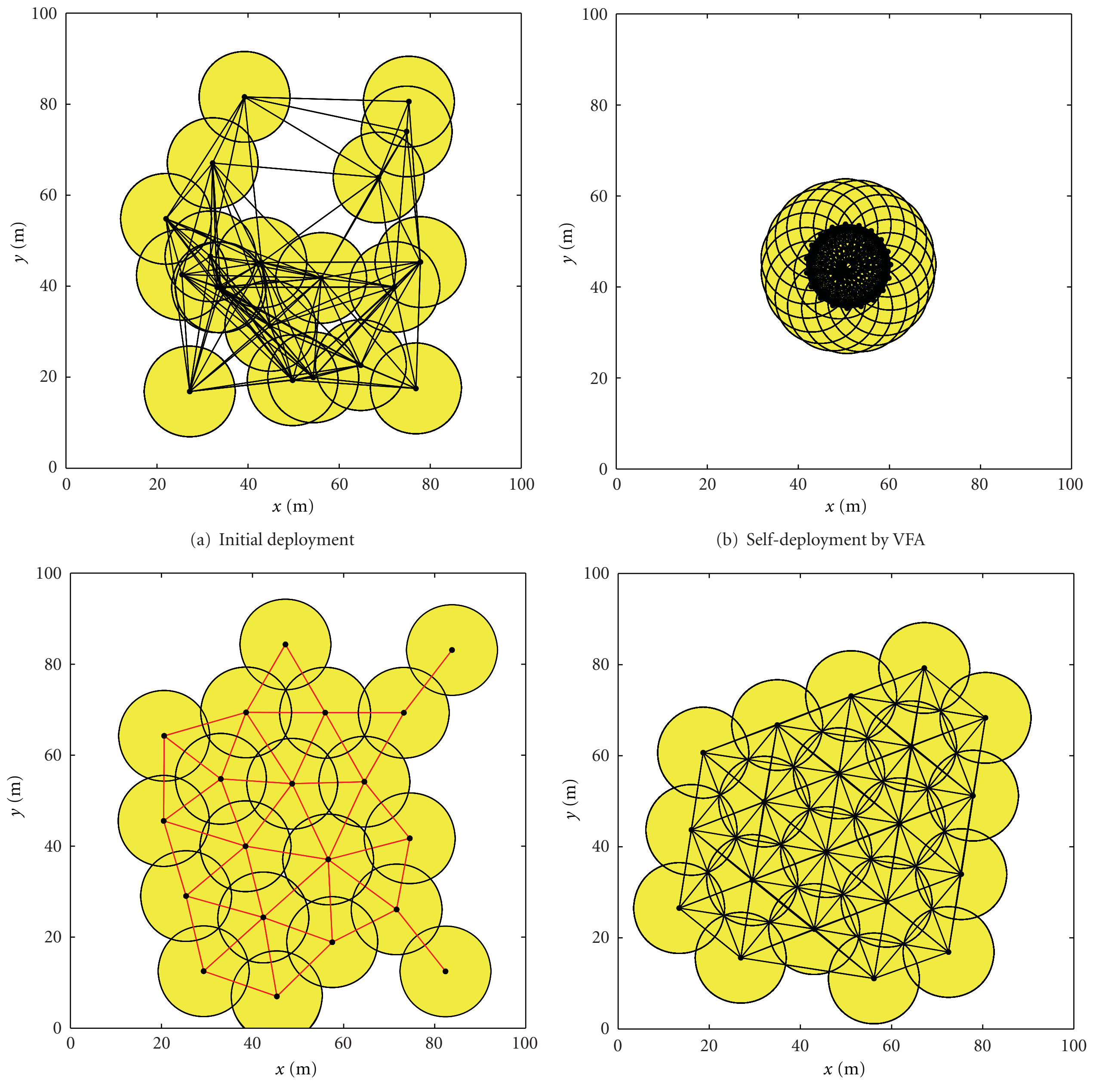


Figure 10: Uniformity comparison among VFA, IVFA, and low-R_c VFA.

6.2. Performance of High-R_c VFA. Figure 11 shows the self-deployment from stochastic initial distribution in Figure 8(a), when R_c/R_s is equal to 4. In this simulation, the communication range is 40 m, while the sensing range is 10 m. The sampling time is 0.1 sec, and the damping coefficient k_d is 0.5. Here we investigate self-deployment results under VFA, Gabriel Graph (GG, same as acute-angle test mesh in [8]) and high-R_c VFA.

Figure 11(a) is the initial deployment and its unit disk graph (UDG). It can be seen that more intersections of connectivity edges have been found compared to the network in Figure 8(a) due to the larger R_c. In Figure 11(b), a fully connectivity graph and conspicuous nodes stacking are shown in self-deployment results by VFA. Self-deployment by GG mesh has obtained a result without stacking. Figure 11(c) shows the GG mesh graph after self-deployment. Although GG is a planar graph, it does not guarantee the hexagonal formation. Self-deployment by high-R_c VFA is shown in Figure 11(d). Clearly, it shows that a regular ideal deployment has been attained in this circumstance.

In order to demonstrate self-deployment of large-scale mobile sensor networks, 100 mobile nodes are randomly placed for a concentrated deployment, as shown in



(a) Initial deployment

(b) Self-deployment by VFA

(c) Self-deployment by GG

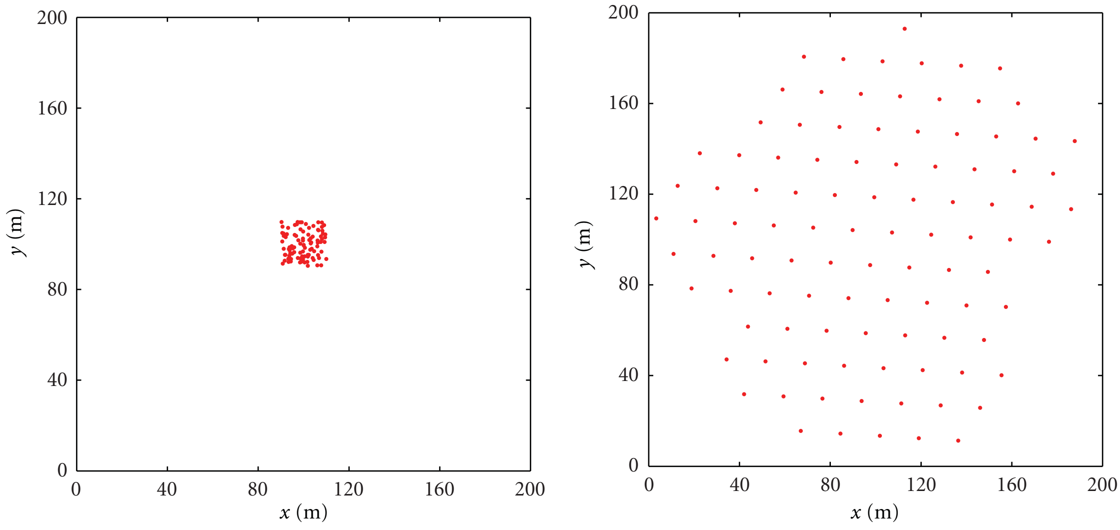
(d) Self-deployment by high-R_c VFA

Figure 11: Self-deployment results in $R_c/R_s = 4$.

Figure 12(a). The communication range is 40 m, while the sensing range is 10 m. The sampling time is 0.1 sec, and the damping coefficient k_d is 0.5. The simulation result of self-deployment by high-R_c VFA is shown in Figure 12(b). The number of iterations is 6000 (10 minutes). It is obvious that an asymptotically ideal deployment has been reached.

The impact of network size on self-deployment is shown in Figure 13. Figure 13(a) shows the change of distance uniformity with different nodes' numbers. In both IVFA and GG, distance uniformity deteriorates as the number of nodes is increased, while high-R_c VFA provides almost constant uniformity. And we can also observe that high-R_c VFA obtains the best performance uniformity from Figure 13(b).

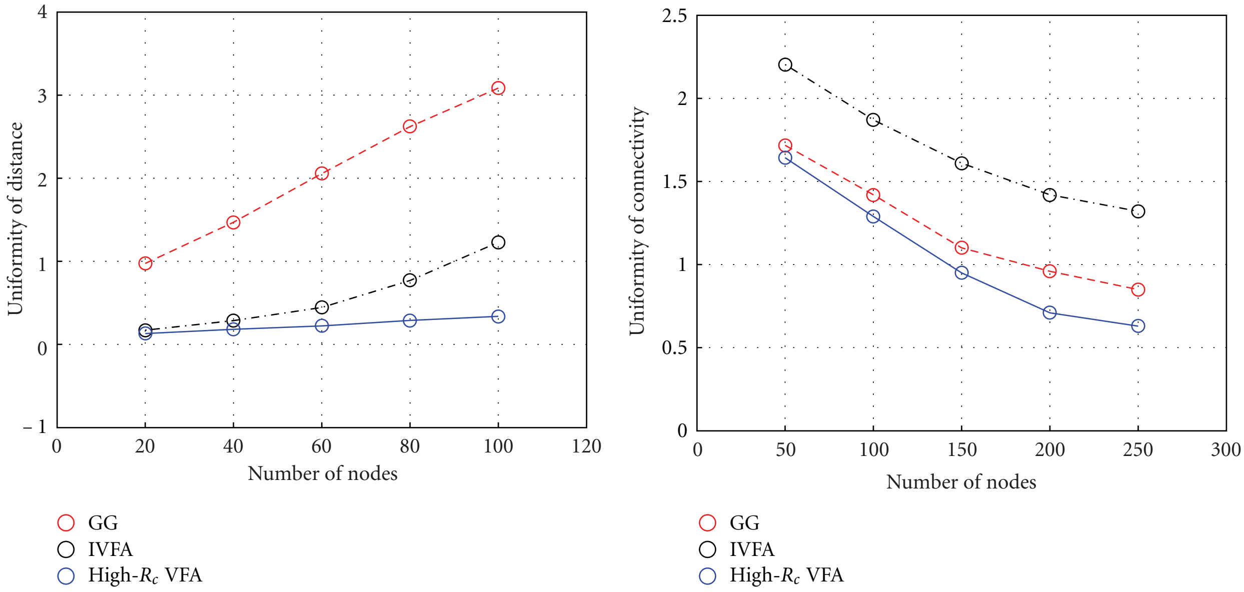
6.3. Impact of Control Coefficients. The control coefficients influence the algorithm's performance. In the proposed extended virtual force-based approach, we focus on the sampling time and damping coefficient in this section. Equation (2) models the movement of self-deployment as a continuous system. Therefore, more accurate control can be obtained, when the sampling time becomes shorter. Figure 14(a) shows the distance uniformity comparison of different sampling time under high-R_c VFA. It can be seen that uniformity oscillation is obvious, when sampling time is 1 sec, while convergence state and low uniformity are shown when the sampling time is 0.1 sec and 0.01 sec. However, a short sampling time needs lots of communication cost,



(a) Concentrated initial deployment

(b) Self-deployment by high-R_c VFA

Figure 12: Self-deployment in large-scale concentrated deployment.



(a) Uniformity of distance

(b) Uniformity of connectivity

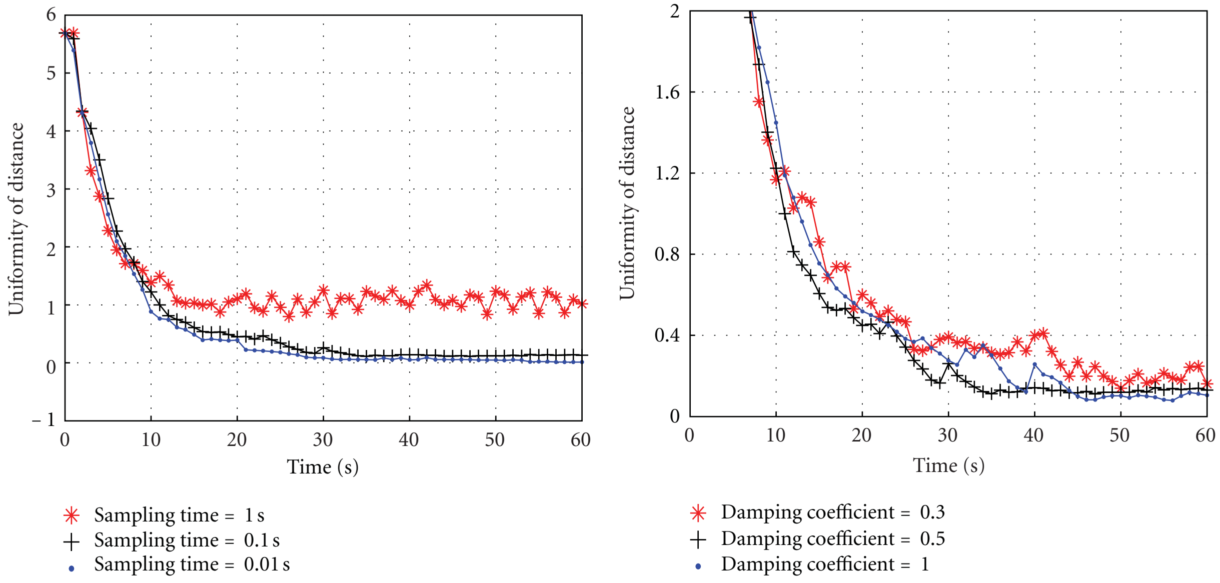
Figure 13: Impact of nodes' number under IVFA, GG, and high-R_c VFA.

and it also requires the process time should be shorter than sampling time. These requirements may be limited by the hardware of sensor nodes. So, 0.1 sec is a reasonable sampling time in this paper.

The damping coefficient also strongly influences self-deployment. Shucker and Bennett [8] has shown that convergence time increases when the damping coefficient is higher. We conduct simulations with damping coefficient of 0.3, 0.5, and 1 in this paper. As shown in Figure 14(b), the lower damping coefficient takes a faster moving at the beginning of self-deployment. However, it gets a higher value of distance uniformity with oscillation. It means that we should choose a higher damping coefficient to reach the more regular formation. Thus, we consider damping coefficient varying from 0.5 to 1 is reasonable value.

7. Conclusions

In this paper, we analyze the stability and formation of self-deployment with virtual physics-based methods. We argue the connectivity maintenance and nodes stacking problems of the existing VFA algorithms. To solve these



(a) Uniformity with different sampling time

(b) Uniformity with different damping coefficient

Figure 14: Impact of control coefficients and the nodes' number.

problems, we proposed an extended virtual force-based approach which can achieve the ideal deployment under self-deployment and can be applied into networks with the different ratio of communication range to sensing range. In low-R_c VFA, the orientation force is introduced to guarantee the continuous connectivity. While in high-R_c VFA, we propose the judgment of distance force between node and its faraway nodes in order to prevent node stacking from nonplanar connectivity.

Simulation results confirm the efficiency of the proposed extended virtual force approach in coverage rate, distance uniformity, and connectivity uniformity, respectively. In low-R_c mobile sensor networks, ideal deployment with continuous connectivity is achieved under low-R_c VFA. Regular ideal deployment can be obtained in high-R_c mobile sensor networks by high-R_c VFA, though the connectivity graph is nonplanar. Future work involves testing the extended virtual force approach in heterogeneous mobile sensor networks with unequal communication (sensing) range and studying k-coverage requirement.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61104086), the National Defense Advanced Research Project of China (Grants nos. 104080, 40405020401), and Excellent Young Scholars Research Fund of Beijing Institute of Technology (Grant no. 2011CX04031).

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.

[2] Y. R. Tsai and Y. J. Tsai, "Sub-optimal step-by-step node deployment algorithm for user localization in wireless sensor networks," in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, pp. 114–121, June 2008.

[3] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, April 2001.

[4] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems*, pp. 299–308, Springer, 2002.

[5] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1293–1303, 2003.

[6] J. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 35, no. 1, pp. 78–92, 2005.

[7] R. S. Chang and S. H. Wang, "Self-deployment by density control in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1745–1755, 2008.

[8] B. Shucker and J. Bennett, "Virtual spring mesh algorithms for control of distributed robotic macorsensors," Tech. Rep. CU-CS-996-05, University of Colorado, Boulder, Colo, USA, 2005.

[9] M. L. Lam and Y. H. Liu, "ISOGRID: an efficient algorithm for coverage enhancement in mobile sensor networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1458–1463, October 2006.

[10] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, "Redundancy and coverage detection in sensor networks," *ACM*

Transactions on Sensor Networks, vol. 2, no. 1, pp. 94–128, 2006.

[11] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[12] C. H. Wu, K. C. Lee, and Y. C. Chung, "A Delaunay Triangulation based method for wireless sensor network deployment," in *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS '06)*, vol. 1, pp. 253–260, July 2006.

[13] W. Li, Y. I. Kamil, and A. Manikas, "Wireless array based sensor relocation in mobile sensor networks," in *Proceeding of the 5th ACM International Wireless Communications and Mobile Computing Conference (IWCMC '09)*, pp. 832–838, June 2009.

[14] D. W. Gage, "Command control for many-robot systems," in *Proceedings of the 19th Annual AUVS Technical Symposium*, June 1992.

[15] R. Simmons, D. Apfelbaum, W. Brugard et al., "Coordination for multi-robot exploration and mapping," in *Proceedings of the 7th National Conference on Artificial Intelligence*, 2000.

[16] T. Balch and M. Hybinette, "Behavior-based coordination of large-scale robot formations," in *Proceedings of the 4th International Conference on Multiagent Systems*, pp. 363–364, July 2000.

[17] J. Chen, S. Li, and Y. Sun, "Novel deployment schemes for mobile sensor networks," *Sensors*, vol. 7, no. 11, pp. 2907–2919, 2007.

[18] G. Tan, S. A. Jarvis, and A. M. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, Article ID 4770105, pp. 836–848, 2009.

[19] K. Derr and M. Manic, "Extended virtual spring mesh (EVSM): the distributed self-organizing mobile ad hoc network for area exploration," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, pp. 5424–5436, 2011.

[20] S.-C. Huang, "Ion-6: a positionless self-deploying method for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 940920, 10 pages, 2012.

[21] Z. Miao, L. Cui, B. Zhang, and J. Li, "Deployment patterns for *k*-Coverage and *l*-Connectivity in wireless sensor networks," in *Proceedings of the IET International Conference on Wireless Sensor Network*, pp. 73–77, 2010.

[22] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 131–142, May 2006.

[23] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, "Complete optimal deployment patterns for full-coverage and K-connectivity ($k \leq 6$) wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 401–410, May 2008.

[24] X. Bai, S. Li, and J. Xu, "Mobile sensor deployment optimization for k-Coverage in wireless sensor networks with a limited mobility model," *IETE Technical Review*, vol. 27, no. 2, pp. 124–137, 2010.

[25] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Wireless Ad Hoc and Sensor Networks*, vol. 1, no. 1-2, pp. 89–123, 2005.

[26] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 7, pp. 946–958, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 686172, 12 pages
doi:10.1155/2012/686172

Research Article

Rendezvous Data Collection Using a Mobile Element in Heterogeneous Sensor Networks

Junzhao Du, Hui Liu, Longfei Shangguan, Luo Mai, Kai Wang, and Shucong Li

Software School and State Key Lab of ISN, Xidian University, Shaanxi 710071, China

Correspondence should be addressed to Junzhao Du, dujz@xidian.edu.cn

Received 11 January 2012; Accepted 20 February 2012

Academic Editor: Mo Li

We study the rendezvous data collection problem for the Mobile Element (ME) in heterogeneous sensor networks where data generation rates of sensors are distinct. The link quality is instable in our network model and the sensory data cannot be aggregated when transmitting. The Mobile Element is able to efficiently collect network wide data within a given delay bound; meanwhile the network eliminates the energy bottleneck to prolong its lifetime. For case study, we consider the trajectory planning for both Mobile Relay and Mobile Sink on a tree-shaped network. In the Mobile Relay case where the ME's trajectory must pass through a sink to upload sensory data for further processing, an $O(n \lg n)$ algorithm named RP-MR is proposed to approach (1) the optimal Rendezvous Points (RPs) to collect global sensory data; (2) the optimal data collection trajectory for the Mobile Relay to gather the cached data from RPs. In the Mobile Sink case where the Mobile Element can process the sensory on its motion, we develop an $O(n \lg^2 n)$ algorithm named RP-MS to recursively investigate the optimal solution. Both the theoretical analysis and extensive simulations verify the correctness and effectiveness of proposals.

1. Introduction

Since the emergence, Wireless Sensor Networks (WSNs) undertake intensive data collection during the sensing period. However, the limited energy storage of tiny sensors involved in a WSN harshly confines their ability of transmitting the acquired data. As a result, the premature death of several sensors always perturbs the connectivity of the network. To address this issue, some researchers [1–3] introduced a novel data gathering technique that combines controlled mobility and rendezvous-based data collection, where sensors first forward their sensing data to several Rendezvous Points (RPs), and then, tell a Mobile Element (ME) to pick up the cached data for further processing. A critical issue in such method is conserving limited energy and maximizing network lifetime.

To date, there is a broad range of literatures on improving the utility of this mobility-based strategy, managing to reduce the total data transmission load taken by the whole WSN. Although many of these prior efforts were able to present some optimal or near optimal solutions, most of these works based on several impractical assumptions. That is, the link quality is assumed fair enough; the data generation rates of sensors (In this paper, the words sensor and sensor node are used interchangeably.) are identical, and data can be fused when transmitting.

Generally, such assumptions cannot offer a holistic view of typical properties in WSNs for three reasons. First of all, medium quality and paths self-interfere can trigger packets loss and retransmission that consequently affect the link quality. As a result, the wireless link cannot stay in a stable condition, and data packet cannot be successfully delivered to the destination in each transmission. Secondly, collecting data like images from distant area of battlefield cannot be fused when transmitting. Therefore, the energy consumption of data transmission cannot be calculated as in previous works [1–3]. Thirdly, data generation rates on distinct sensors are usually diverse depending on the varying monitoring objectives in the network. For example, in monitoring environmental habitat, some sensors maybe indicated to observe the vibration frequency of the ground, while others are allocated to probe the local humidity. Apparently, the rates on these devices are not the same.

Basically, such properties which we term heterogeneous feature among sensor nodes mirror dynamic properties of

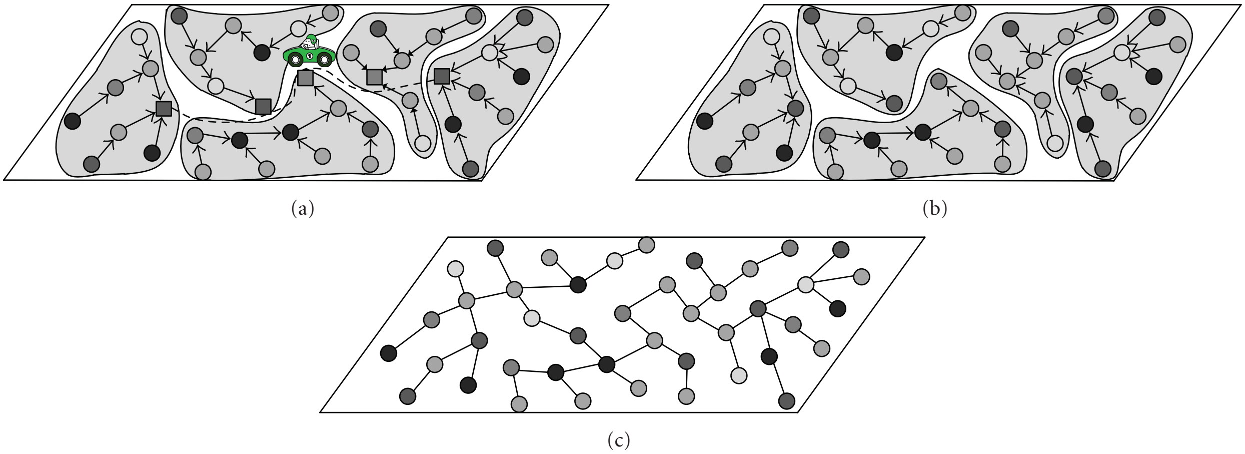


Figure 1: An example of Rendezvous Point Selection Problem. The circle represents sensor nodes and the square stands for Rendezvous Point. Notice that the darkness of circles varies with data generation rate.

WSNs, therefore requires rational handling. The significance of heterogeneous feature among sensor nodes lies in that, it captures the uncertain property of WSNs, which assists us to conduct in-depth analysis on mobility-based techniques, that is, locating RP in the "hot space" where data generation rate is high. As the basic assumptions in previous efforts are revealed to be impractical, the introduction of heterogeneous feature will trigger a big change in the optimization structure like maximizing network lifetime or minimizing the cost of communication, and so, many previous algorithms, which assumed sensor have the same data generation rate, cannot be mapped into this typical scenario.

In this paper, we are focusing on rendezvous planning in WSNs by taking the heterogeneous feature into consideration. In our network model, we adopt ETX [4] metric to measure the link quality considering the lossy link and the interference from hidden terminal. The data type and generation rate here are disparate from sensor to sensor; therefore the data cannot be fused when transmitting. Our goal is to find a ME moving trajectory and corresponding RPs, so as to minimize total energy consumption on in-network data communication. (In-network data communication represents the total data transmission cost for gathering global sensory data.) We formulate such an optimization problem as Minimum-Energy Rendezvous Point Selection Problem (shown in Figure 1). For case study, we consider two cases where the Mobile Element is Mobile Relay and Mobile Sink (Mobile Sink is also called Mobile Base Station in some literatures.), respectively. In the Mobile Relay case, the sensing data are first forwarded to several RPs for caching. Then, a Mobile Relay reaches each RP to download the cached data and then upload them at the Sink's site. In the Mobile Sink case, the Mobile Sink does data processing while it downloads the sensing data from each RP, with no need for uploading data at fixed point.

Our contribution can be summarized as follows.

(1) We investigate the heterogeneous feature of sensors in WSNs and formulate rendezvous planning problems based on practical assumptions. In our network model, the link quality is measured by ETX; the data generation rates are disparate and cannot be fused when transmitting.

(2) For case study, we consider two cases where the ME is Mobile Relay and Mobile Sink, respectively. In the Mobile Relay case, we propose an $O(n \lg n)$ algorithm named RP-MR to achieve the optimal solution in tree-shaped topology. In the Mobile Sink case, we propose an $O(n \lg^2 n)$ algorithm to achieve the optimal solution under the same settings.

(3) To the best of our knowledge, our work is the first attempt to consider heterogeneous feature of sensor nodes, which will trigger different data generation rate, and undermine the mechanism, aggregation. Ignoring such considerations will possibly lead to a higher energy burden on in-network data transmission.

The rest of paper is organized as follows. Section 2 reviews relevant work about rendezvous planning. Section 3 gives a brief description of our problems. Section 4 and Section 5 investigate the rendezvous planning problem in two cases, Mobile Relay and Mobile Sink, respectively. We evaluate the performance of our proposed algorithms in Section 6 and conclude our paper in Section 7.

2. Related Work

Typically, the Mobile Element is either Mobile Relay or Mobile Sink. The mobility of the ME is uncontrollable or controllable.

The basic idea of the uncontrollable case has been proposed by Shah et al. in their work Data Mules [5, 6]. The mobile relays, called MULEs, are used to collecting data from the sensors via single-hop transmission. Kim et al. [7] proposes protocol, SEAD, to find a tradeoff between the packet transmission delay and the energy consumption of reconfiguring the tree topology. The mobile sinks randomly access the tree from specified sensors (Access Point). As mobile sinks move freely, the communication between mobile sinks and Access Point can be multihop. Luo et al. [8] investigate the approach that makes use of a mobile sink for balancing the traffic load and in turn improving network lifetime. They engineer a routing protocol, MobiRoute, to effectively support sink mobility. Tian et al. [9] design protocol, AVRP, to adopt Voronoi scoping plus dynamic anchor selection to handle free and unpredictable moving of mobile sink, and protocol, TRAIL, to use the trail of mobile sink for guiding packet forwarding. The use of mobile elements with predictable pattern has been addressed in [10–16]. As the movements of the mobile elements in these works are fully controllable, the trajectories can be planed and routing protocol can be designed using the prediction.

The sensors first send their data to rendezvous point [1, 2], then the mobile sink visits these RPs to collect data via single-hop transmission. As the multihop routing between the sensors to the stationary sink is reduced to a smaller number of hops, the energy therefore gets reserved. Our previous work [17] falls into this category, which only considers the Mobile Relay case. To additively update the routing structure of data collection for mobile sink, Li et al. [18] propose protocol to only perform a local modification to update the routing structure, while the routing performance is bounded and controlled compared to the optimal performance.

3. Basic Assumptions and Network Model

In this section, we give our basic assumptions and present the network model.

3.1. Basic Assumptions

(1) We assume that wireless communication is the dominating energy-consuming factor and hence omit other energy consuming functions such as sensing and processing.

(2) We assume that the ME roams in the network to collect all those sensory data by visiting a series of RPs, with a required data collection delay bound D. The delay bound can be measured by the maximum distance L that the mobile sink is allowed to move. The location of sensor nodes can be calculated by localization algorithm [19].

(3) We assume that each transmission (regardless of success or failure) costs equivalent energy. Thus, the energy consumption for delivering a data packet from a node to its neighbor is proportional to the transmission count until the neighbor receives the packet successfully.

(4) We assume that the storage capacity of sensors is large enough to buffer the sensing data during a collection period. Several recent sensor network platforms [20] are equipped with 32 512MB MAC or NAND flash with ultralow power consumption.

(5) We assume that data cannot be fused at any intermediate node before it arrives at the ME. Data gathering without aggregation has been applied in many scenarios like battlefield surveillance, habitat monitoring, which is dominated by the heterogeneous data.

3.2. Network Model. We consider a WSN with a set of $\aleph$ nodes, where $|\aleph| = n$. A link $l(i, j) \in Ł$ exists if and only if the Euclidean distance between the sensor nodes is shorter than the transmission range R. Let $r(i)$ be the data generation rate (packet/min) at node i, and $c_l(i, j)$ represents the ETX of link $l(i, j) \in Ł$.

Initially, we assume each sensor has equivalent energy E, and each sensor cost identical energy e to transmit a data packet in each link. Thus, the energy consumption for transmitting one data packet from node i to its neighbor node j is $e \cdot c_l(i, j)$. Denote $\iota^{\text{out}}(i)$ and $\iota^{\text{in}}(i)$ the sets of outgoing and incoming links at node i. On link l, denote $r_l(i)$ the data generated from node i that is attributed to link l. The network lifetime T here is defined as the time when the first node dies.

4. Problem Formulation

In this section, we first present the description and formal definition of our problem and then give the formulation and list the characteristics of the problem.

4.1. Problem Definition. In our problem, a plenty of sensors must deliver their cached data to a Mobile Element within time interval D. The data generation rate on distinct sensors is different. Our objective is to find a data collection trajectory for the ME, such trajectory consists of several RPs between which the ME moves and at which they download data. On the other hand, RPs cache the global data of the network and upload them to the ME via short-range transmission when it passes by. The energy consumption for global data transmission should be minimized under the constraint that the ME must cover its trajectory before it depletes energy or the time interval D. We name this problem as Minimum-Energy Rendezvous Point Selection Problem (MRPSP). The formal definition can be given as follows.

Definition 1 (Minimum-Energy Rendezvous Point Selection Problem). Given a set of sensors $\aleph$ and the maximum allowable trajectory length $L = \nu_{\text{ME}} \cdot D$ of ME, we are aiming to find (1) a set of RPs that cache the global data of the network, so as to minimize the in-network transmission cost and (2) a length-constrained data collection trajectory (Here the length-constrained data collection trajectory stands for the trajectory whose length is shorter than the given bound L.) that consists of these RPs.

To present the optimization problem intuitively, we give the mathematical model in the following. Let $c_p(i)$ denote the

ETX sum of the link between node i and corresponding RP. We define the total in-network transmission cost C as

$$C(\Re) = \sum_{i\in\aleph} r(i)\cdot c_p(i)\cdot\lambda. \tag{1}$$

With above definition, our optimization model can be formulated as

$$\begin{aligned} \text{OPT}\ \min\ \ & C(\Re) = \sum_{i\in\aleph} r(i)\cdot c_p(i)\cdot\lambda \\ \text{s.t.}\ \ & \sum_{l\in\iota^{\text{out}}(i)} r_l(i) = r(i), \end{aligned} \tag{2}$$

$$\sum_{l\in\iota^{\text{in}}(i)} r_l(i) + r(i) = \sum_{l\in\iota^{\text{out}}(i)} r_l(i), \tag{3}$$

$$\left(\sum_{l\in\iota^{\text{out}}(i)} r_l(i)\cdot c_l(i,j)\right)\cdot e \le E, \tag{4}$$

$$Y(\Re) \le L. \tag{5}$$

Then we explain each constraint corresponding to the flow conservation equations for multihop routing mentioned above.

(1) If node i is the source node which only needs to relay its own data packet, then it fulfills (2).

(2) If node i is the intermediate node which generates and relays others' data packet, then it fulfills (3).

(3) On the sensor node i, we have the energy consumption (4).

(4) The length of the data collection trajectory which composed of $\Re$ should be constrained within L in the form of (5).

The detailed notations can be found in Table 1.

By this formulation, we implicitly assume that the data rate between any two nodes i and j (i.e., $l(i,j)$) is feasible under the corresponding link capacity. Our problem has an important characteristic: we should jointly consider two metrics (shown in Figure 2), Euclidean distance and ETX, in our problems. The former metric is to measure the distance covered by the ME, while the latter one is to weigh the transmission cost on each link. By doing so, we could satisfy the constraint on the length of data collection trajectory and obtain the optimization objective simultaneously.

We convert the graph topology into a tree network through constructing a minimum spanning tree on the initial graph. The significance of tree network is threefold.

First, the existence of multipaths between a pair of nodes in the network always results in the problem that data packets might be trapped in a loop and cannot be delivered to the designated points [21]. We use Minimum Spanning Tree (or Steiner Tree) to avoid such problem. Second, such tree can be regarded as an initial optimization for network communication cost for it reduces the total count of ETX in terms of routing paths. Third, many applications [1, 22] are configured with fixed reporting trees when they are deployed.

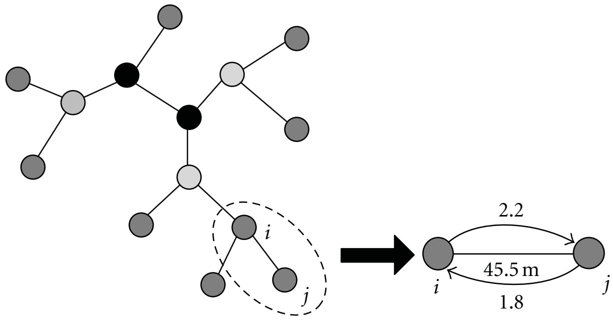


Figure 2: Two metrics on one link: euclidean distance and ETX, respectively.

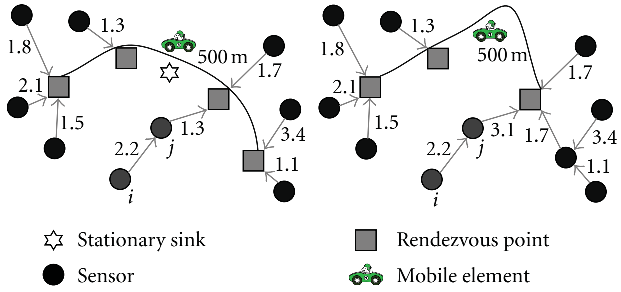


Figure 3: Illustration of two scenarios: Mobile Relay versus Mobile Sink.

And such conversion can guarantee our approach to be applied in a more general case.

For case study, we take the type of Mobile Element into consideration. The intrinsic difference between these two cases is as follows: the sink is fixed at a stationary point in the first problem, but can move freely in the second problem. For the former case, the ME can be a Mobile Relay without any processing ability, and it only caches the sensing data and sends them to the stationary sink for further processing. For the latter one, sink can move freely, so the ME can be such Mobile Sink itself. Both problems have practical utilizations. For the sake of delay tolerant WSNs, employing a stationary sink and a mobile collector is relatively economical since equipping the sink with mobility results in much higher manufacturing cost and shorter allowable traversal. On the opposite, in the networks where the sensory data are urgent, it is necessary for us to adopt a Mobile Sink to satisfy with a tighter latency constraint. Figure 3 shows the difference among these two cases.

5. Case 1: Rendezvous Point Selection with a Mobile Relay

In most cases, equipping the sink with mobility seems a "good" idea but lacks practical meanings, because such improvement will inevitably result in higher or even unbearable manufacturing cost and power consumption. Instead, network designers more want to let the sink locate in a stationary place and wait for the global sensing data. We facilitate this stationary sink with a Mobile Relay to collect

Table 1: Notation used.

$\aleph$	The set of sensor nodes.
Ł	The set of links.
R	The communication radius of sensor.
$\mathfrak{R}$	The set of RPs.
$\mathfrak{R}_{\text{opt}}$	The optimal set of RPs with minimum data transmission cost.
λ	The energy consumption parameter.
s	The stationary point where the Sink is located.
$C(\mathfrak{R})$	The total in-network communication cost.
D	The maximum data collection delay.
T	The network lifetime.
E	The initial energy of each sensor.
L	The maximum length of ME's trajectory.
G	The adjacent matrix of the tree-shaped topology.
$Y(\mathfrak{R})$	The length of trajectory composed of $\mathfrak{R}$.
e	The energy consumption for transmitting one data packet from a node to its neighbor.
$p(i)$	The multihop transmission link between the node i and nearest RP.
p_{opt}	The optimal data collection trajectory with minimum data transmission cost.
$r(i)$	Data generation rate (packet/min) at node i.
T_v	The subtree which rooted at node v.
v_{ME}	The average velocity of the Mobile Element.
$p_{i,j}$	The path between node i and node j.
$l(i, j)$	Wireless link between node i and node j.
$c_l(i, j)$	The ETX of link $l(i, j) \in$ Ł.
$\iota^{\text{out}}(i)$	The sets of outgoing links at node i.
$\iota^{\text{in}}(i)$	The sets of incoming links at node i.
$c_p(i)$	The ETX sum of the link between node i and corresponding RP, $c_p(i) = \sum_{l \in p(i)} c_l$.
κ_v	The sum of the data rate at all the decedent nodes of $v, \kappa_v = \sum_{i \in T_v} r(i)$.
$c_{\text{save}}(p_{u,v})$	We define $c_{\text{save}}(p_{u,v}) = C(u) - C(\mathfrak{R})$, where $\mathfrak{R} = \{u, v\}$. This equation can be paraphrased as how many transmission count can be saved after we extend the path from node u to node v.

data from RPs along a designated path, and then, the Mobile Relay uploads these data to the sink for further processing.

5.1. An Optimal Solution: RP-MR. In this part, we present an $O(n \lg n)$ algorithm named Rendezvous Planning with a Mobile Relay (RP-MR) to find the optimal locations of RPs and the corresponding length-constrained data collection trajectory. The basic idea of RP-MR is to investigate which qualified trajectory (within L, passes s) retrenches the largest data transmission count of forwarding global sensing data to RPs. It can be easily proved that the path with largest data transmission count savings is the one with minimum in-network communication cost. To facilitate our algorithm, we first give three lemmas that help to construct the infrastructure of RP-MR.

Lemma 2. *Given a directed tree rooted at r, let v denote any descendent node of r, one has*

$$c_{\text{save}}(p_{r,v}) = c_{\text{save}}\left(p_{r,\text{parent}(v)}\right) + k_v \cdot c(v, \text{parent}(v)). \quad (6)$$

Proof. According to the definition of $C_{\text{save}}(p, (r, v))$, we can develop the left part of (6) as

$$c_{\text{save}}(p_{r,v}) = C(r) - C(\mathfrak{R}_1), \quad (7)$$

where $\mathfrak{R}_1 = \{r, v\}$.

Also, we develop the right part of (6) as

$$\begin{aligned} & c_{\text{save}}\left(p_{r,\text{parent}(v)}\right) + k_v \cdot c(v, \text{parent}(v)) \\ & \quad = C(r) - C(\mathfrak{R}_2) + k_v \cdot c(v, \text{parent}(v)), \end{aligned} \quad (8)$$

where $\mathfrak{R}_2 = \{r, \text{parent}(v)\}$.

By (7)-(8), we have

$$\begin{aligned} & C(\mathfrak{R}_2) - C(\mathfrak{R}_1) - k_v \cdot c(v, \text{parent}(v)) \\ & \quad = k_v \cdot c(v, \text{parent}(v)) - k_v \cdot c(v, \text{parent}(v)) \\ & \quad = 0, \end{aligned} \quad (9)$$

where $\mathfrak{R}_1 = \{r, v\}$, $\mathfrak{R}_2 = \{v, \text{parent}(v)\}$. □

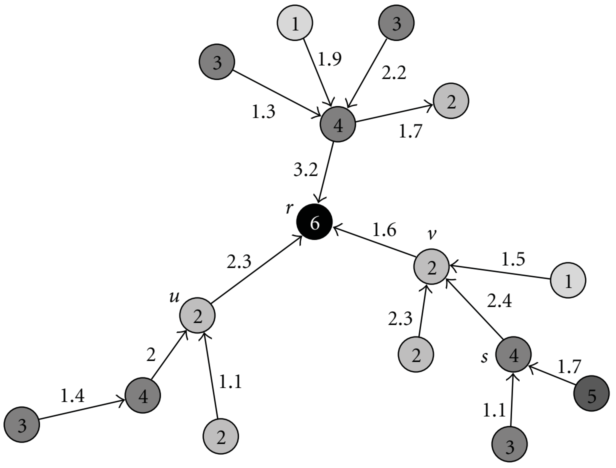


FIGURE 4: Illustration of a directed tree which rooted at node r, node v is derived from r.

Example 3. In Figure 4, the numbers attached to each edge indicate the ETX of links, and the numbers on the nodes denote the data generation rate. According to the definitions of $c_{\text{save}}(p_{r,\text{parent}(v)})$ and $c(v, \text{parent}(v))$, we can calculate the following values from Figure 4:

$$\begin{gathered} c_{\text{save}}\left(p_{r,\text{parent}(v)}\right) = C(r) - C(\Re') = 177 - 149.8 = 27.2, \\ k_s \cdot c(s, \text{parent}(s)) = 2.4 \times (4 + 3 + 5) = 28.8. \end{gathered} \tag{10}$$

Also, we can get

$$c_{\text{save}}(p_{r,s}) = C(r) - C(\Re'') = 177 - 121 = 56. \tag{11}$$

Thus,

$$c_{\text{save}}(p_{r,s}) = c_{\text{save}}\left(p_{r,\text{parent}(v)}\right) + k_s \cdot c(s, \text{parent}(s)) \tag{12}$$

holds.

Lemma 4. *Let z denote any intermediate node on path $p_{u,v}$, then one has*

$$C(\Re_1) = C(z) - c_{\text{save}}(p_{u,z}) - c_{\text{save}}(p_{z,v}), \tag{13}$$

where $\Re_1 = \{u, v\}$.

Proof. We first develop the left part of (13):

$$C(\Re_1) = C(u) - c_{\text{save}}(p_{u,v}). \tag{14}$$

Then, we develop the right part of (13):

$$\begin{aligned} C(z) &- c_{\text{save}}(p_{u,z}) - c_{\text{save}}(p_{z,v}) \\ &= C(z) - [C(z) - C(\Re_2)] - [C(z) - C(\Re_3)] \\ &= C(\Re_2) + C(\Re_3) - C(z), \end{aligned} \tag{15}$$

where $\Re_2 = \{u, z\}$, $\Re_3 = \{z, v\}$.

Substituting the $C(\Re_3)$ by $C(z) - c_{\text{save}}(p_{z,v})$ in (15), we have

$$\begin{aligned} C(\Re_2) &+ C(\Re_3) - C(z) \\ &= C(\Re_2) + C(z) - c_{\text{save}}(p_{z,v}) - C(z) \\ &= C(\Re_2) - c_{\text{save}}(p_{z,v}). \end{aligned} \tag{16}$$

By (14)–(16), we have

$$\begin{aligned} C(u) &- c_{\text{save}}(p_{u,v}) - C(\Re_2) + c_{\text{save}}(p_{z,v}) \\ &= C(u) - C(\Re_2) - c_{\text{save}}(p_{u,v}) + c_{\text{save}}(p_{z,v}) \\ &= c_{\text{save}}(p_{u,z}) - c_{\text{save}}(p_{u,v}) + c_{\text{save}}(p_{z,v}) \\ &= 0. \end{aligned} \tag{17}$$

Therefore, Lemma 4 holds. □

Lemma 5. *Let v denote any node on the tree-shaped topology, then one has*

$$C(v) = C(\text{parent}(v)) + c(v, \text{parent}(v)) \cdot (\kappa_s - 2 \cdot \kappa_v). \tag{18}$$

Proof. This equation is easy to understand intuitively. And the proof is quite straightforward, therefore it is omitted here. □

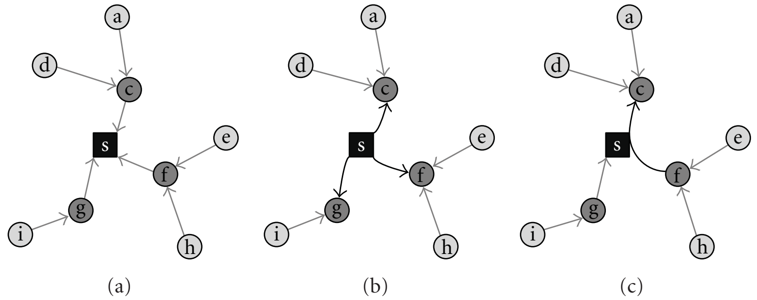
5.2. Algorithm. In the algorithm, as all the trajectories that originate from the roots to the other vertices are different in length, we have to maintain all the trajectories starting from the root to other vertices. In the combination process (steps 7 and 8), if we do not order the paths by their length first, the time needed for finding the optimal trajectory containing the roots with length at most L is $O(n^2)$. That is because we have to combine all pairs of paths originated from the root. But if we order the trajectories before we execute the combination process, we could find the optimal trajectory containing the roots in $O(n)$ time. To reduce the time complexity of our algorithm, here we adopt *quicksort* [23] to order the trajectories in $O(n \lg n)$.

Figure 5(a) illustrates the initial tree-shaped network. At step 2, we orient tree into the directed form with the root s. From step 3 to step 5, the algorithm computes $C(v)$ for each node v; at step 6, the algorithm investigates all the length-constrained (within L) trajectories; that is, in Figure 5(b), we have three trajectories within L, $s \to c$, $s \to g$, and $s \to f$, respectively. At step 7, we select two trajectories found at step 6 and integrate them into a new trajectory. For instance, we operate $s \to c$ and $s \to g$, achieve the new path $c \to s \to g$ or $g \to s \to c$. If the length of this new trajectory is shorter than L, we regard it as a candidate path otherwise we drop it. After determining all the possible trajectories at step 6, we choose the one with maximum $c_{\text{save}}()$ from all the candidate trajectories. Then, we output the $\Re_{\text{opt}}$ and C_{opt}. Hence, the optimal path in Figure 5(c) is $f \to s \to c$.

5.3. Theoretical Analysis of RP-MR. In this part, we state some properties of RP-MR algorithm (see Algorithm 1).

```
(1) procedure RP-MR $(G, L/2, s)$
(2)     Orient tree-shaped topology into a directed case with the root $s$;
(3)     Compute $\kappa_v$ for each node $v$ using a bottom-up fashion;
(4)     Compute $C(\mathfrak{R})$ in a bottom-up fashion, where $\mathfrak{R} = \{s\}$;
(5)     Compute $C(v)$ for each node $v$ by using Lemma (6) in a top-down fashion;
(6)     Calculate the $c_{\text{save}}()$ for the trajectories (shorter than $L$) that originate from node s by using Lemma (6), respectively;
(7)     Combine each pair of trajectories found in Step (6). Calculating the $c_{\text{save}}()$ in the same way;
(8)     Computing the in-network data transmission cost $C(\mathfrak{R}_{\text{opt}})$ for the path with maximum $c_{\text{save}}$ from the compound paths set;
(9)     return $p_{\text{opt}}$ and $C(\mathfrak{R}_{\text{opt}})$;
(10) end procedure
```

Algorithm 1: RP-MR algorithm.



(a) (b) (c)

Figure 5: Case study 1: an illustration of Rendezvous Point Selection with a Mobile Relay.

Theorem 6. *The data collection trajectory p_{opt} found by RP-MR is optimal.*

Proof. This theorem is easy to understand intuitively. As we have enumerated all the length-constrained trajectories and restored them in the candidate set, the optimal trajectory in the candidate set is the global optimal trajectory. □

Theorem 7. *RP-MR can be done in $O(n \lg n)$.*

Proof. First, it takes $O(n)$ time to orient the tree into directed form via top-down fashion. Then we get κ_v for each node v, and calculate $C(s)$ through a bottom-up traversal. In the following step, through top-down fashion, we compute the $C()$ value for each node and get the c_{save} for the length-constrained trajectory by retraversing the tree in the top-down fashion. Next, we adopt quick-sort to help find the optimal trajectory with the maximum c_{save}, this can be accomplished in $O(n \lg n)$. Finally, calculating the optimal $C(\mathfrak{R}_{\text{opt}})$ can be done in $O(1)$. Therefore, the time complexity of RP-MR is $O(n \lg n)$. □

Theorem 8. *The data collection trajectory found by RP-MR always heads to the "hot area" in which the sensors' data generation rates are pretty high.*

Proof. This theorem can be proved by contradiction. Let $T_{h\,a}$ denote the subtree with maximum sum of data generation rates (hot area). Suppose the optimal data collection trajectory (p_{opt}) does not head to $T_{h\,a}$. Then considering the endpoint contained in this optimal trajectory, we remove trajectory between the endpoint p^* and the root of sub-tree $T_{h\,a}$, and prolong the trajectory into $T_{h\,a}$ with same length to guarantee that the length is shorter or equal to L. According to Lemma 2, the c_{save} of the new trajectory is greater than that of p_{opt}. This contradicts the fact that the p_{opt} is the optimal one. Therefore, the data collection trajectory found by RP-MR always heads to the "hot area". □

6. Case 2: Rendezvous Point Selection with a Mobile Sink

In this section, firstly we state the basic idea and present the pseudocode of RP-MS algorithm. Then, an execution example of RP-MS is illustrated. Finally, we evaluate the performance of RP-MS with three theorems.

Sometimes, the Sink can move inherently, that is, the Airborne Warning and Controlling System collect information from a battlefield, or the rangers with a handheld device monitors habitat area. The mobility of the sink inspires us to propose an efficient approach to deal with the case with a Mobile (unfixed) Sink.

6.1. An Optimal Solution: RP-MS. Generally, for a specific node v, the optimal trajectory either passes through node v, or lay in one of the sub-trees by removing the node v from the tree. Based on this observation, we design a recursive algorithm to find the optimal trajectory. First, the algorithm finds the optimal length-constrained trajectory which must pass through a stationary point called *median* (the node v with minimum $C_{\mathfrak{R}}$, where $C_{\mathfrak{R}} = \{v\}$), and then, by "removing" the *median*, the algorithm finds the current

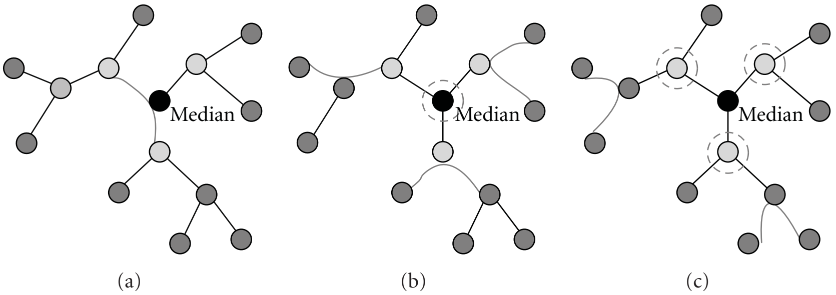


Figure 6: Illustration of Splitting tree technique.

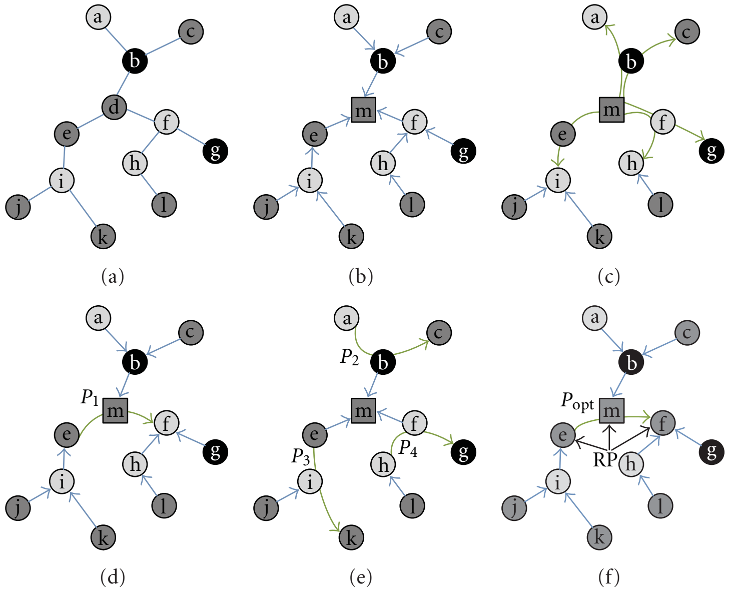


Figure 7: Case study 2: an illustration of Rendezvous Point Selection with a Mobile Sink.

optimal trajectory in the sub-trees which are rooted at one of median's children. In the same way, the algorithm keeps on "splitting" each sub-tree. For each sub-tree, we find the optimal length-constrained trajectory which passes through its root until every sub-tree has been examined, restoring these optimal trajectories' candidates into the candidate set. Finally, we get the global optimal one by examining the candidate set. An illustration of splitting tree technique can be found in Figure 6. In the following, we present the pseudocode of our optimal solution named Rendezvous Planning with a Mobile Sink (RP-MS) algorithm.

6.2. Algorithm. Figure 7(a) shows the initial tree-shaped network. At step 2, RP-MS calculates C_v for each node v to determine the median m. Then, it orients the network into a directed tree rooted at m, shown in Figure 7(b). Figures 7(c) and 7(d) illustrate the first iteration of finding the optimal trajectory which passes through m. The result of this process is $e \rightarrow m \rightarrow f$ (denote as P_1). At step 5, the tree splits into three sub-trees. From step 5 to step 9, it executes the optimal path in each sub-tree P_2, P_3, P_4, respectively. Finally, we choose the global optimal path P_{opt} from P_1, P_2, P_3, P_4, and the result is shown in Figure 7(f).

6.3. Theoretical Analysis of RP-MS. In this part, we state some properties of RP-MS algorithm (see Algorithm 2).

Theorem 9. *Finding the median can be done in $O(n)$.*

Proof. We divide the process of finding the median into four substeps. At step 1, we arbitrarily select a node as the root r and then orient the tree-shaped topology into a directed one. This can be achieved in $O(n)$ through Breadth-First Traversing. At step 2, we compute $C(\mathfrak{R})$ in $O(n)$ with the bottom-up fashion, where $\mathfrak{R} = \{r\}$. At step 3, with top-down fashion, we achieve $C(\mathfrak{R})$ for each node v in $O(n)$ as well, where $\mathfrak{R} = \{r\}$. Finally, we select the node with minimize $C(\mathfrak{R})$ as the median in $O(n)$. Hence, the execution time to find the median is $O(n)$. □

Theorem 10. *The data collection trajectory p_{opt} found by RP-MS is optimal.*

Proof. For a given tree-shaped topology, the optimal data collection trajectory either passes through the median or be fully contained in one of the sub-trees by removing the median from the tree. As for the former case, we can easily achieve the global optimal solution in the first iteration.

```
procedure RP-MS $(G, L/2)$
    Find $median(m)$ of the tree-shaped topology;
    Orient tree-shaped topology into directed one which is rooted at $m$;
    push $m$ into queue $q$;
    while $q \neq NULL$ do
        Find the optimal trajectory $p_{\mathrm{opt}}$ by calling RP-MR $(G, L, v)$, where $v = q \cdot \mathrm{pop}()$;
        Insert $p_{\mathrm{opt}}$ into the candidate set $SET$;
        Insert the children of $v$ into queue $q$;
    end while
    Find the global optimal trajectory with minimum $C(\Re_{\mathrm{opt}})$ from the candidate set $SET$;
    return $p_{\mathrm{opt}}$ and $C(\Re_{\mathrm{opt}})$;
end procedure
```

Algorithm 2: RP-MS algorithm.

While for the latter case, the recursive process guarantees the optimal solution could be found. Hence, the data collection trajectory found by RP-MS is optimal. □

Theorem 11. *The algorithm RP-MS finds the optimal RPs and corresponding trajectory in $O(n \lg^2 n)$.*

Proof. As the pseudocode shows, the algorithm executes in iterative fashion. For a given level of the recursion, suppose there are k sub-trees to be examined, let n_i denote the number of nodes of the ith problem. Consequently, we need $O(n_i \lg n_i)$ time to find the current optimal trajectory. As $\sum_{i \in K} n_i \leq n$, we obtain each level of recursion in $O(n \lg n)$. Notice that the depth of the tree is $O(\lg n)$; therefore we get the optimal trajectory in $O(n \lg^2 n)$. □

Theorem 12. *The data collection trajectory found by RP-MS always heads to the "hot area" as well.*

Proof. This theorem can be proved by contradiction as in Theorem 8. The proof is quite simple and is omitted here. □

7. Performance Evaluation

In previous sections, we described the design techniques and important properties of our proposed algorithms. In this section, we validate the applicability and feasibility of such algorithms via comprehensive and large-scale simulations.

We evaluate the performance of RP-MR and RP-MS algorithms, and compare them with other two well-known algorithms, GREEDY and RD-VT. GREEDY is a heuristic algorithm, where the Mobile Element always moves to the nearest neighbor until the length of its path exceeds L. RD-VT is introduced by [1], where the Mobile Element's trajectory is formed by a preorder traversal of sensor nodes in the network.

7.1. Simulation Settings. We randomly deploy various number of sensor nodes on a 200 m × 200 m square area. The transmission range of sensor is set to be 100 m, and every sensor can be chosen as a RP here. As the energy consumption of the data transmission is proportional to the ETX of corresponding link, we evaluate the energy consumption performance of various algorithms by comparing the total ETX of resulting data routing paths. All the evaluation results are averaged based on 10 different runs.

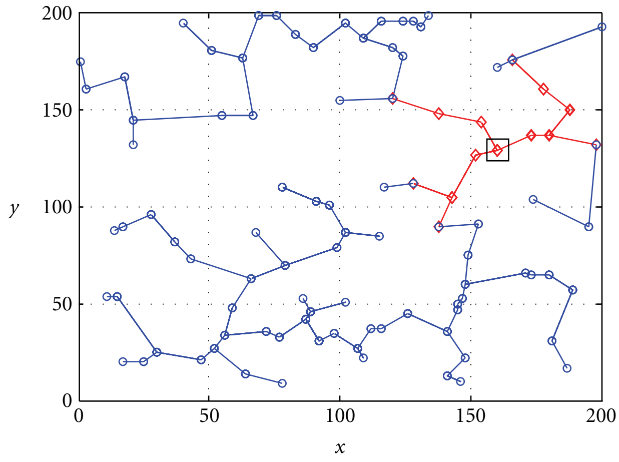


Figure 8: Data collection trajectory found by GREEDY algorithm.

7.2. Simulation Results. We first give the snapshot of ME's trajectory found by four algorithms in a stationary tree-shaped network, where 100 sensors are formed into a Minimum Spanning Tree. The maximum length (L) of data collection trajectory is set to be 400 m here. Figures 8, 9, 10, and 11 show the results found by GREEDY, RD-VT, RP-MR, and RP-MS, respectively. The stationary sink here is denoted by a rectangle. Clearly, the data collection trajectory in Figures 8 and 9 is limited in the top right corner in the field where less sensors are deployed. As a result, a great portion of sensors have to deliver their data to the nearest RPs via long distance transmission, which in turn, harshly increases the total in-network communication cost. In the opposite, the trajectory found by RP-MR and RP-MS is prolonged and oriented to the biggest brunch in the field, that is, the brunch in the top left and bottom left corner. Therefore, the data transmission count from each sensor to RP get fair

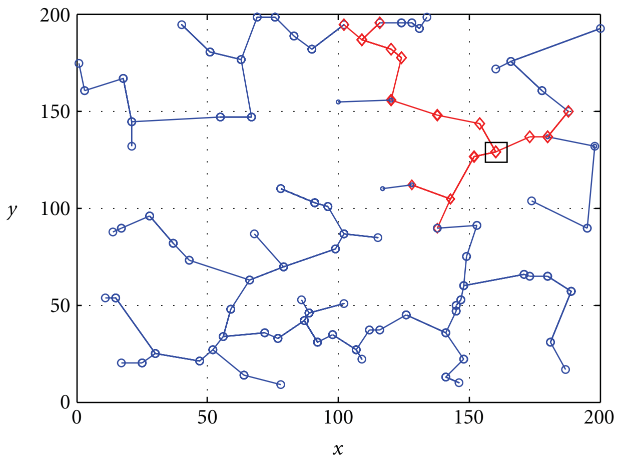


Figure 9: Data collection trajectory found by RD-VT algorithm.

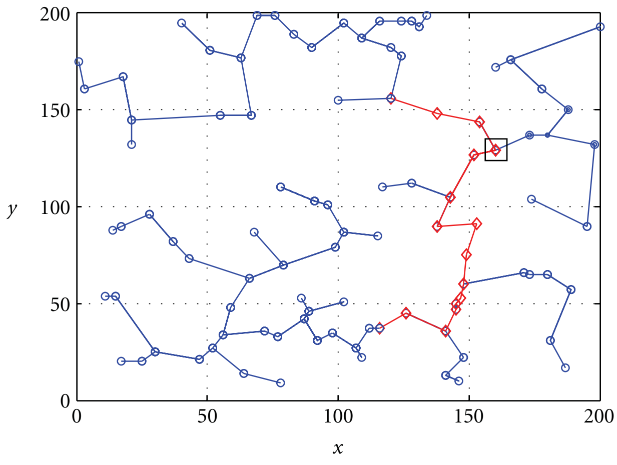


Figure 10: Data collection trajectory found by RP-MR algorithm.

enough, which in turn apparently reduced the in-network communication cost and verified our theorem that the data collection trajectory found by RP-MR and RP-MS always head to the hot area.

Figure 12 illustrates the relationship between the in-network communication cost of data routing paths and the length of Mobile Element's trajectory. 300 sensor nodes are randomly deployed in the field. All algorithms' performance becomes better when the ME mobility increases. Consistent with the result in Figure 12, our algorithms are superior to other two competitors. Specifically, the gap between RP-MR and RP-MS decreases with the rise of the length of the trajectory. This implies that when the length of data collection trajectory is chosen properly, the result yield by RP-MR stays in the same level of the result acquired by RP-MS in terms of in-network transmission cost. In contrast, GREEDY and RD-VT achieves similar result which almost triple the result of RP-MR and RP-MS under the same settings, which implies that the network can enjoy a longer lifetime in RP-MR and RP-MS compared with other two algorithms.

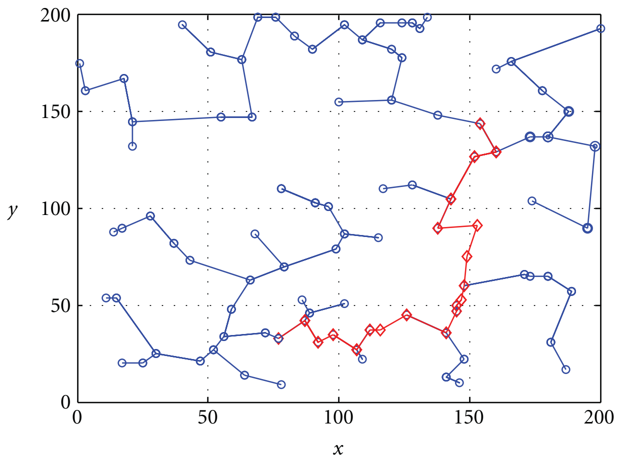


Figure 11: Data collection trajectory found by RP-MS algorithm.

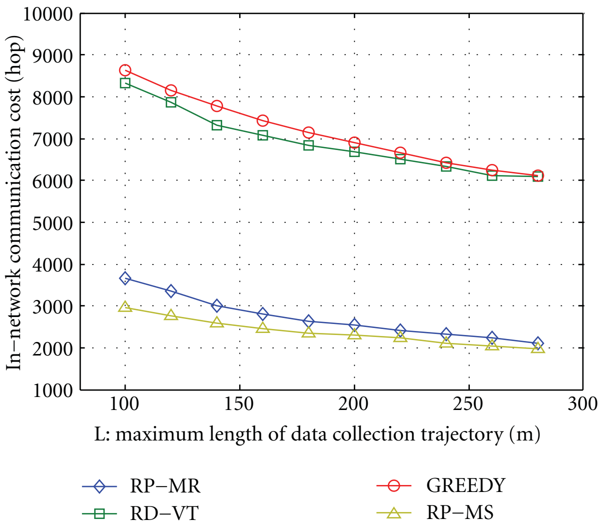


Figure 12: In-network communication cost versus length of data collection trajectory *L*.

Figure 13 shows the relationship between the maximum work load among RPs and the length of the ME's trajectory. Similar to Figure 12, 300 sensor nodes are randomly deployed in the field as well. According to Figure 13, the maximum work load incurred by four algorithms all decrease with the increase of *L*. More specifically, when the ME trajectory is short, the maximum work load is extremely heavy in GREEDY (2300), RD-VT (2000), RP-MR (1100), and RP-MS (1000). However, as *L* increases, RP-MR and RP-MS rapidly decreases its maximum work load, and the gaps between them are largely narrowed. This implies that both of our proposed algorithms can effectively mitigate the work load on the bottleneck RP. Although the same trend can be found in the result achieved by GREEDY and RD-VT, they still stay in a high level with respect to the work load, which implies that the network cannot be prolonged effectively.

Figure 14 compares the in-network communication cost of resulting data routing paths formed by different algorithms with the number of source nodes. The mobile sink

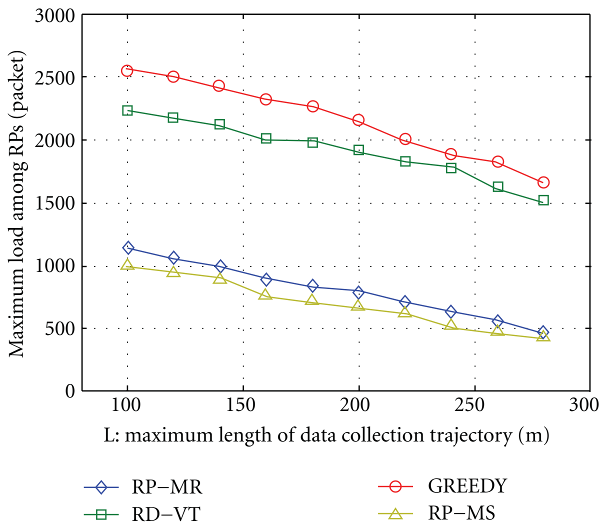


Figure 13: Maximum load among RPs versus length of data collection trajectory L.

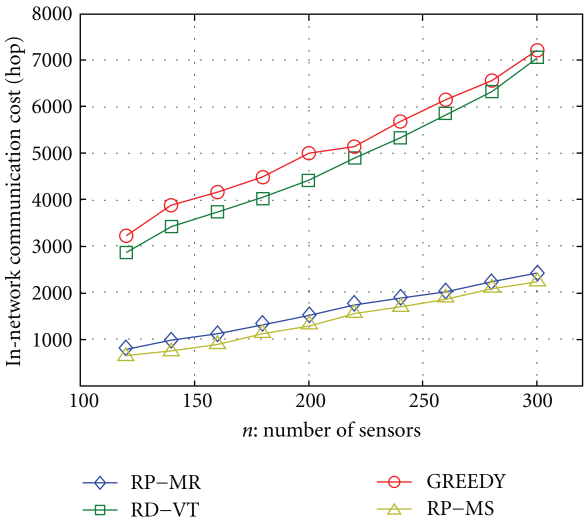


Figure 14: In-network communication cost versus number of sensor nodes.

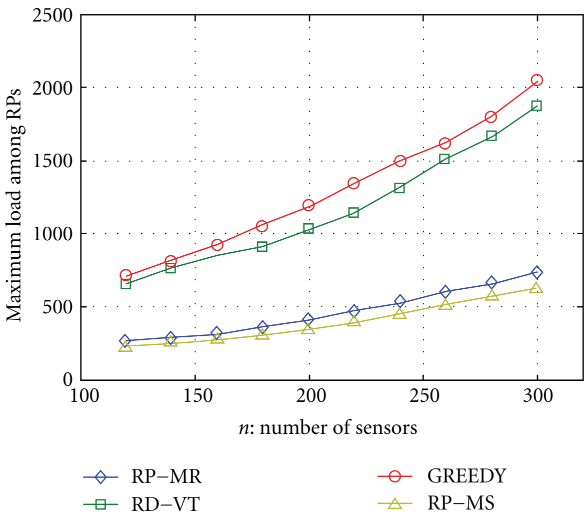


Figure 15: Maximum load among RPs versus number of sensor nodes.

trajectory length L is 400 m. Figure 14 illustrates that RP-MR and RP-MS both outperform GREEDY and RD-VT. Compared with the RP-MS where the Mobile Element can move freely, its counterpart RP-MR also works well, with only a slight performance distortion in terms of the data transmission cost. In addition, the gap from GREEDY or RD-VT to RP-MS or RP-MR increases as more sensor nodes are involved. It indicates that RP-MR and RP-MS can effectively reduce the energy consumption by taking advantage of the property that the optimal trajectory always heads to the largest brunch, that is, “hot area”.

Figure 15 compares the maximum work load among RPs incurred by different algorithms with the number of sensor nodes. Same to Figure 14, the ME trajectory length L is set to be 400 m as well. According to Figure 15, all of the maximum work load among RPs generated by the four algorithms increase with the number of sensors. Specifically, the curves of GREEDY and RD-VT change rapidly with the increase of sensor nodes, while the curves composed of RP-MR and RP-MS increase smoothly. This is an indication of the effectiveness of RP-MR and RP-MS to prolong the network lifetime since the RP always drain its energy first among the sensor nodes and paralyze the network. Besides, the gap between RP-MR and RP-MS aggrandized gradually, which indicates that RP-MS is superior to RP-MR in terms of prolonging the network lifetime.

8. Conclusion

In this paper, we study the rendezvous data collection problem for the Mobile Element (ME) in heterogeneous sensor networks where data generation rates of sensors are distinct. Essentially different from previous works, the link quality is assumed instable and the sensory data cannot be aggregated in our paper. Such assumptions highlight the dynamic properties of WSNs and tend to be more practical. We introduce to optimize the energy consumption on gathering the global data. By doing so, the Mobile Element is able to efficiently collect network-wide data within a given delay bound meanwhile the network eliminates the energy bottleneck to prolong its lifetime. Then we give two algorithms for dealing with different rendezvous data collection scenarios. In the Mobile Relay scenario where the ME’s trajectory must pass through a given point to upload sensory data, an $O(n\lg n)$ algorithm named RP-MR is proposed to approach the optimal solution: (1) the optimal Rendezvous Points (RPs) to collect global sensory data; (2) the optimal data collection trajectory for the Mobile Relay to gather the cached data from RPs and upload to a fixed point, that is, the sink, for further processing. In the latter case where the Mobile Element can process the sensory on its motion, that is, Mobile Sink, we develop an $O(n\lg^2 n)$ algorithm named RP-MS to recursively investigate the optimal solution.

We prove the correctness of our proposals both in theoretical analysis and extensive simulations.

A possible future work is to further optimize the data collection trajectory. In the case studies of this work, we implicitly assume that the Mobile Element must walk along the edges of the tree. Actually, instead of traversing each RP, the ME needs only one step in the data transmission range of the RPs. By doing so, the length of data collection trajectory is retrenched. Therefore we could find more RPs and further minimize the in-network communication cost than the approaches before.

Acknowledgments

This work is supported by the NSFC under Grants no. 60803152, 60903206, and 61100075, the Fundamental Research Fund for the Central Universities (K50510230004), Open Funds of ISN National Lab under Grant no. ISN-9-09, Research on Key Technologies of Electromagnetic Spectrum Monitoring Based on Wireless Sensor Network under Grant No. 2010ZX03006-002-04.

References

[1] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 231–239, May 2008.

[2] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in mobility-assisted wireless sensor networks," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS '07)*, pp. 311–320, December 2007.

[3] Y. Tirta, Z. Li, Y. H. Lu, and S. Bagchi, "Efficient collection of sensor data in remote fields using mobile collectors," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN '04)*, pp. 515–519, October 2004.

[4] D. S. J. de Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 134–146, September 2003.

[5] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy-efficient data collection in sensor networks," in *Proceedings of the IEEE Workshop on WiOpt*, 2004.

[6] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks".

[7] 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 ACM International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 193–204, November 2003.

[8] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J. Hubau, "Mobiroute: routing towards a mobile sink for improving life-time in sensor networks," in *Proceedings of the Distributed Computing in Sensor Systems (DCOSS '06)*, 2006.

[9] K. Tian, B. Zhang, K. Huang, and J. Ma, "Data gathering protocols for wireless sensor networks with mobile sinks," in *Proceedings od the IEEE Global Telecommunications Conference (Globecom '10)*, 2010.

[10] P. Baruah, R. Urgaonkar, and B. Krishnamachari, "Learning-enforced time domain routing to mobile sinks in wireless sensor fields," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 525–532, November 2004.

[11] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks".

[12] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–15, 2005.

[13] Y. Tirta, Z. Li, Y.-H. Lu, and S. Bagchi, "Efficient collection of sensor data in remote fields using mobile collectors," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN '04)*, pp. 515–519, October 2004.

[14] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proceedings of the IEEE Military Communications Conference (MILCOM '03)*, pp. 688–693, October 2003.

[15] P. Venkitasubramaniam, S. Adireddy, and L. Tong, "Sensor networks with mobile access: optimal random access and coding," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1058–1068, 2004.

[16] M. Li, W. Cheng, K. Liu, Y. H. Liu, X. Y. Li, and X. Liao, "Sweep coverage with mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1534–1545, 2011.

[17] L. Shangguan, L. Mai, J. Du, W. He, and H. Liu, "Energy-efficient heterogeneous data collection in mobile wireless sensor networks," in *Proceedings of the PMECT of IEEE ICCCN*, 2011.

[18] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 2246–2254, 2011.

[19] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[20] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 374–381, April 2006.

[21] W. Stallings, *Data and Computer Communications*, Higher Education Press, Beijing, China, 7th edition, 2006.

[22] X. Li, A. Nayak, and I. Stojmenovic, "Exploiting actuator mobility for energy-efficient data collection in delay-tolerant wireless sensor networks," in *Proceedings of the 5th International Conference on Networking and Services (ICNS '09)*, pp. 216–221, April 2009.

[23] C. A. R. Hoare, "Partition: algorithm 63, quicksort: algorithm 64, and find: algorithm 65," *Communications of the ACM*, vol. 4, no. 7, pp. 321–322, 1961.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 153912, 12 pages
doi:10.1155/2012/153912

Research Article

Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems

Li Lu,[1] Jinsong Han,[2] Lei Hu,[3] and Lionel M. Ni[4]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[3] *State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing 100049, China*
[4] *Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China*

Correspondence should be addressed to Li Lu, lulirui@gmail.com

Received 14 December 2011; Accepted 14 February 2012

Academic Editor: Mo Li

The objective of private authentication for Radio Frequency Identification (RFID) systems is to allow valid readers to explicitly authenticate their dominated tags without leaking the private information of tags. In previous designs, the RFID tags issue encrypted authentication messages to the RFID reader, and the reader searches the key space to identify the tags. Without key-updating, those schemes are vulnerable to many active attacks, especially the compromising attack. We propose a strong and lightweight RFID private authentication protocol, SPA. By designing a novel key-updating method, we achieve the forward secrecy in SPA with an efficient key search algorithm. We also show that, compared with existing designs, (SPA) is able to effectively defend against both passive and active attacks, including compromising attacks. Through prototype implementation, we demonstrate that SPA is practical and scalable for current RFID infrastructures.

1. Introduction

The proliferation of RFID applications [1–5] raises an emerging requirement—protecting user privacy [6] in authentications. In most RFID systems, tags automatically emit their unique serial numbers upon reader interrogation without alerting their users. Within the scanning range, a malicious reader thus can perform bogus authentication on detected tags to obtain sensitive information. For example, without privacy protection, any RFID reader is able to identify a consumer's ID via the serial number emitted from the tag. In this case, a buyer can be easily tracked and profiled by unauthorized entities. Nowadays, many companies embed RFID tags with produced items. Those tags indicate the unique information of the items they are attached to. Thus, a customer carrying those items might easily get subject to silent track from unauthorized entities in a much larger span. Some sensitive personal information would thereby be exposed: the illnesses she suffers from indicated by the pharmaceutical products; the malls where she shops; the types of items she prefers, and so on. To prevent such unexpected leakage of private information, a secure RFID system must meet two requirements. First, a valid reader must be able to successfully identify the valid tags; on the other hand, misbehaving readers should be isolated from retrieving private information from these tags.

To address the above issue, researchers employ encryptions in RFID authentication. Each tag shares a unique key with the RFID reader and sends an encrypted authentication message to the reader. Instead of identifying the tag directly, the back-end database subsequently searches all keys that it holds to recover the authentication message and identify the tag. For simplicity, we will denote the reader device and back-end database by the "reader" in what follows. Two challenging issues on the reader side must be addressed in the key storage infrastructure and search algorithm: the search efficiency and the security guarantee. Searching a key should be sufficiently fast to support a large-scale system, while the maintained keys should be dynamically updated to meet security requirements.

Many efforts have been made to achieve efficient private authentication. To the best of our knowledge, the most efficient protocols are tree based [7, 8]. They provide efficient key search schemes with logarithm complexity. In those approaches, each tag holds multiple keys instead of a single key. A virtual hierarchical tree structure is constructed by the reader to organize these keys. Every node in the tree, except the root, stores a unique key. Each tag is associated with a unique leaf node. Keys in the path from the root to the leaf node are then distributed to this tag. If the tree has a depth d and branching factor δ, each tag contains d keys and the entire tree can support up to $N = d^{\delta}$ tags. A tag encrypts the authentication message by using each of its d keys. During authentication, the reader performs a depth-first search in the key tree. In each hierarchy, the reader can narrow the search set within δ keys. Thus, the reader only needs to search $d\delta$ keys for each tag's authentication. Therefore, the key search complexity of identifying a given tag from N tags is logarithmic in N.

The tree-based approaches are efficient, nevertheless, not sufficiently secure due to the lack of a key-updating mechanism. Most tree-based approaches do not update keys of tags dynamically. Since the storage infrastructure of keys in tree-based approaches is static, each tag shares common keys with others. Consequently, compromising one tag will reveal information of other tags. To address this problem, we need to provide a dynamic key-updating mechanism to such approaches. The major challenge of dynamic key-updating in tree-based approaches is consistency. If a single tag updates its keys, some other tags have to update their keys accordingly. Till now, consistent and dynamic key-updating mechanisms have scarcely been seen in the literatures.

In this paper, we propose a strong and lightweight RFID private authentication protocol, (SPA), which enables dynamic key-updating for tree-based authentication approaches. Besides consistency, SPA also achieves forwarding secrecy without degrading key search efficiency. We also show that SPA outperforms existing designs in defending against both passive and active attacks, including the compromising attack.

The rest of this paper is organized as follows. We introduce related work in Section 2. We present the SPA design in Section 3. In Section 4, we analyze the security guarantee of SPA. We evaluate the performance of SPA via a prototype implementation in Section 5. We conclude this paper in Section 6.

2. Related Work

Many approaches have been proposed to achieve private authentication in RFID systems. Weis et al. [9] proposed a hash-function-based authentication scheme, HashLock, to avoid tags being tracked. In this approach, each tag shares a secret key k with the reader. The reader sends a random number r as the authentication request. To respond to the reader, the tag generates a hash value on the inputs of r and k. The reader then computes $h(k, r)$ of all stored keys until it finds a key to recover r, thereby identifying the tag. The search complexity of HashLock is linear to N, where N is the number of tags in the system. Most subsequent approaches in the literature are aimed at reducing the cost of key search. Juels [10] classifies these approaches into three types.

Tree-Based Approaches. Tree based approaches [7, 8, 11] improve the key search efficiency from linear complexity to logarithmic complexity. Molnar and Wagner proposed the first tree-based scheme, which employs a challenge-response scheme [8], which achieves *mutual* authentication between tags and readers. The protocol uses multiple rounds to identify a tag and each round needs three messages. Since it requires $O(\log N)$ rounds to identify a tag, the exchanged messages incur relatively large communication overhead. In [7], the authors provide a more efficient scheme which performs the authentication via one message from the tag to the reader and no further interactions. However, the tree-based approaches are often vulnerable to the *Tag Compromising Attack*. Because tags share keys with others in the tree structure, compromising one tag results in compromising secrets in other tags.

Synchronization Approaches. Synchronization approaches [12] make use of an incremental counter to enhance the authentication security. When successfully completing an authentication, the counter of a tag augments by one. The reader can compare the value of a tag's counter with the record in the database. If they match, the tag is valid and the reader will synchronize the counter record of this tag. However, incomplete authentications lead the tag's counter larger than the one held by the reader. To solve this problem, the reader keeps a window for each tag. Such a window limits the maximum value of the counter held by the tag. If a tag's counter is larger than the record held by the reader but within the window, the reader still regards this tag as valid. Such schemes are vulnerable to the *Desynchronization Attack*. In such an attack, an invalid reader can interrogate a tag many times so that the counter of this tag exceeds the window recorded in the valid reader. In [13], the authors proposed a protocol to mitigate the impact from desynchronization attacks by allowing tags to report the number of incomplete authentications since the last successful authentication with the reader. Dimitriou proposed a scheme in [14], in which a tag increases its counter only after successful mutual authentications. Those protocols, however, degrade the anonymity of tags. An attacker is still able to interrogate a given tag enough times so that the tag will be immediately recognized when replying with unchanged responses.

Time-Space Tradeoff Approaches. Avoine converted the key search problem to an attempt at breaking a symmetric key [15]. In [16], Hellman studied the key-breaking problem and claimed that recovering a symmetric key k from a ciphertext needs to precompute and to store $O(N^{2/3})$ possible keys. Accordingly, the key search complexity is $O(N^{2/3})$ in key-breaking-based approaches. Obviously, those approaches are not efficient compared with tree-based approaches.

3. SPA Protocol

In this section, we first introduce the challenging issues of static tree-based private authentication approaches. We then present the design of SPA.

3.1. Challenges of Tree-Based Approaches. Existing tree-based approaches [7, 8, 11] construct a balanced tree to organize and store the keys for all tags. Each node stores a key and each tag is arranged to a unique leaf node. Thus, there exists a unique path from the root to this leaf node. Correspondingly, those keys on this path are assigned to the tag. For example, tag T_1 obtains keys k_0, $k_{1,1}$, $k_{2,1}$, and $k_{3,1}$, as illustrated in Figure 1. When a reader authenticates a tag, for example, T_1, it conducts the identification protocol shown in Figure 2. $h(k, r)$ denotes the output of a hash function h on two input parameters: a key k and a random number r. The identification procedure is similar to traversing a tree from the root to a leaf. The reader R first sends a nonce r to tag T_1. T_1 encrypts r with all its keys and includes the ciphertexts in a response. Upon the response from T_1, the reader searches proper keys in the key tree to recover r. This is equal to marking a path from the root to the leaf node of T_1 in the tree. At the end of identification, if such a path exists, R regards T_1 as a valid tag. Usually, the encryption is employed by using cryptographic hash functions.

From the above procedure, we see that tags will, more or less, share some nonleaf nodes in the tree. For example, T_1 and T_2 share $k_{2,1}$, while T_1, T_2, T_3, and T_4 share $k_{1,1}$. Of course all tags share the root k_0. Such a static tree architecture is efficient because the complexity of key search is logarithmic. For the example in Figure 1, any identification of a tag only needs $\log_2(8) = 3$ search steps. On the other hand, if the adversary tampers with some tags; however, it obtains several paths from the root to those leaf nodes of the corrupt tags, as well as the keys on those paths. Since keys are not changed in the static tree architecture, the captured keys will still be used by those tags which are not tampered (for simplicity, we denote those tags as *normal* tags). Consequently, the adversary is able to capture the secret of normal tags.

A practical solution is to update keys for a tag after each authentication so that the adversary cannot make use of keys obtained from compromised tags to attack normal ones. However, the static tree architecture makes it highly inflexible to provide consistent key-updating. Suppose we update the keys of T_1 in Figure 1, we have to change k_0, $k_{1,1}$, $k_{2,1}$, and $k_{3,1}$ partially or totally. Note that $k_{1,1}$ is also used by T_2, T_3, and T_4, and $k_{2,1}$ is used by T_2. To keep the updating consistent, the keys of all influenced tags must be updated and re-distributed. A challenging issue is that if the position of a key is close to the root, the key-updating would influence more tags. For example, updating $k_{1,1}$ would influence half of all tags in the system (T_1, T_2, T_3, and T_4). One intuitive idea is to periodically recall all tags and update the keys simultaneously. Unfortunately, such a solution is not practical in large-scale systems with millions or even hundreds of millions of tags. Another solution is collecting those influenced tags only and updating their keys. This is also difficult because we have to collect a large span of tags even though there is only one tag updating its keys.

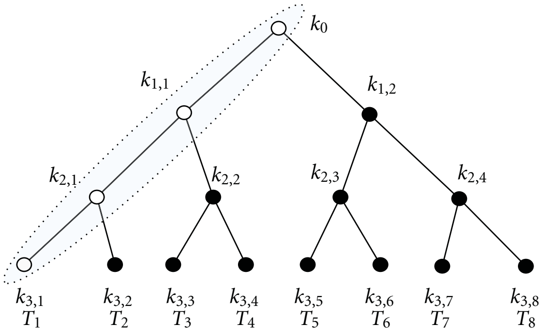


Figure 1: A binary key tree with eight tags.

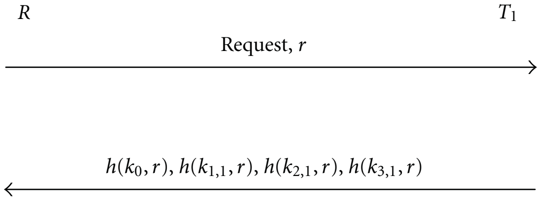


Figure 2: A basic RFID authentication procedure.

This problem motivates us to develop a dynamic key-updating algorithm for private authentication in RFID systems. This is where our proposed SPA enters the picture.

3.2. SPA Overview. SPA comprises four components: *system initialization*, *tag identification*, *key-updating*, and *system maintenance*. The first and second components are similar to tree-based approaches such as [7] and perform the basic identification functions. The key-updating is employed after a tag successfully performs its identification with the reader. In this procedure, the tag and the reader update their shared keys. This key-updating procedure will not break the validation of keys used by other tags. SPA achieves this using temporary keys and state bits. A temporary key is used to store the old key for each nonleaf node in the key tree. For each nonleaf node, a number of state bits are used in order to record the key-updating status of nodes in the subtrees. Based on this design, each nonleaf node will automatically perform key-updating when all its children nodes have updated their keys. Thus, SPA guarantees the validation and consistency of private authentication for all tags. SPA also eases the system maintenance in high dynamic systems where tags join or leave frequently by the fourth component.

3.3. System Initialization. For the simplicity of discussion, we use a balanced binary tree to organize and store keys, as shown by an example in Figure 3. Let δ denote the branching factor of the key tree (e.g., $\delta = 2$ when the key tree is a binary tree). We assume that there are N tags T_i, $1 \leq i \leq N$, and a reader R in the RFID system. The reader R assigns the N tags to N leaf nodes in a balanced binary tree S. Each nonleaf

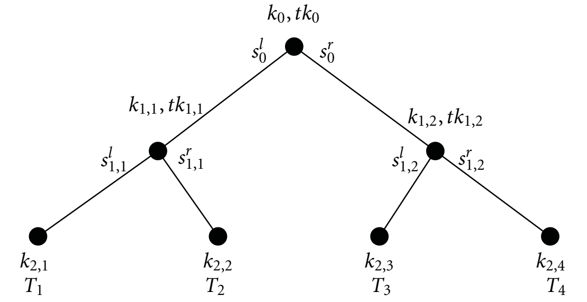


FIGURE 3: A key tree of a system with four tags ($N = 4$).

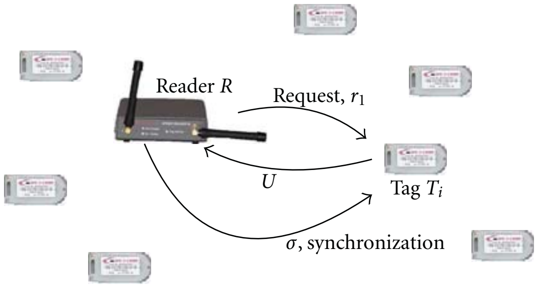


FIGURE 4: Authentication procedure in SPA. After receiving U, Reader R's operations are Step 1, identifying T_i and computes σ; Step 2, sending σ to T_i and key-updating. T_i also updates its keys after checking σ.

node j in S is assigned with two keys, a working key k_j and a temporary key tk_j. The usage of tk_j will be illustrated in Section 3.5. Initially, each key is generated independently randomly by the reader, and $tk_j = k_j$ for all nonleaf nodes.

When a tag T_i is introduced into the system, the reader distributes the $(\lceil \log N \rceil + 1)$ keys to T_i. Those keys are corresponding to the path from the root to tag T_i (for a nonleaf node j at the path, if $tk_j \neq k_j$, tag T_i is assigned with k_j). For example, the keys stored in tag T_1 are k_0, $k_{1,1}$ and $k_{2,1}$, as illustrated in Figure 3. Hereafter, we use d to denote the depth of the tree and $(k_0^i, k_1^i, \ldots, k_d^i)$ to denote the secret keys distributed to T_i.

3.4. Tag Identification. The basic authentication procedure between the reader and tags includes three rounds, as illustrated in Figure 4. In the first round, R starts the protocol by sending a "*Request*" and a random number r_1 (a nonce) to tag T_i, $1 \leq i \leq N$. In the second round, upon receiving the request, T_i generates a random number r_2 (a nonce) and computes the sequence $(h(k_0^i, r_1, r_2), \ldots, h(k_d^i, r_1, r_2))$, where $h(k, r_1, r_2)$ denotes the output of a hash function h on three inputs: a key k and two random numbers r_1 and r_2. T_i replies R with a message $U = (r_2, h(k_0^i, r_1, r_2), \ldots, h(k_d^i, r_1, r_2))$. For simplicity, we denote the elements in U as $u, v_0, \ldots, v_d$. R identifies T_i according to U.

R executes the basic identification procedure to identify T_i, represented as Step 1 in Figure 4. From the root, the reader first encrypts r_1 by using k_0 and compares the result with $h(k_0, r_1, r_2)$ from T_i. If they match, R invokes a recursive algorithm, Algorithm 1, as illustrated in Algorithm 1 to identify T_i. For the key tree in Figure 3, the reader starts from the root and encrypts r_1 by using $k_{1,1}(or\ tk_{1,1})$ and $k_{1,2}$ (or $tk_{1,2}$). Having the results, the reader compares them with received $h(k_1^i, r_1, r_2)$. If $h(k_1^i, r_1, r_2)$ is equal to the result computed from $k_{1,1}$ (or $tk_{1,1}$), the tag belongs to the left sub-tree; otherwise, it belongs to the right sub-tree.

Level by level, R figures out the path of T_i originated from the root by invoking Algorithm 1. Suppose the path reaches an intermediate node j at level l $(1 \leq l \leq d)$ in the tree. At this point, R computes all hash values $h(k_{l+1}, r_1, r_2)$ and $h(tk_{l+1}, r_1, r_2)$ by using the keys of node j's children, then compares them with v_l. Note that v_l is in the authentication message U received from T_i. If there is a match, T_i must belong to the sub-tree of the matched j's child node. Therefore, R extends the path to that node and continues the identification procedure until reaching a leaf node.

Identifying a tag is similar to traversing from the root to a leaf in the key tree. The path is determined by using Algorithm 1. Instead of using only one key for each node, Algorithm 1 uses both the working key k and the temporary key tk.

3.5. Key-Updating. After successfully identifying T_i, R invokes the Key-updating algorithm in Step 2, as shown in Figure 4.

When generating new keys, SPA still makes use of the hash function h. Let k_j be the old key of node j. The reader computes a new key k_j' from the old key k_j as $k_j' = h(k_j)$. The key-updating algorithm for the key tree is shown in Algorithm 2. To remain consistent, the nonleaf node j uses temporary key tk_j to store j's old key. In this way, the key-updating of a tag will not interrupt the authentication procedures of other tags belonging to j's sub-tree.

Two important issues must be addressed when updating keys. First, R should update the keys of the identified tag T_i without interrupting the identification of other tags. This is because the keys stored in nonleaf nodes are shared by multiple tags. Those keys should be updated in a consistent manner. Second, each nonleaf node should automatically update its keys when all its children have updated their keys.

To address the two issues, SPA introduces a number of state bits to each nonleaf node. The basic idea behind this mechanism is that each nonleaf node uses these bits to reflect the key-updating status of its children. Once a child has updated its key, the corresponding bit is set to 1. Each node updates its own key when all its state bits become 1.

Without losing generality, we still use balanced binary key tree S to illustrate this mechanism. Each nonleaf node j in S is assigned with two state bits, denoted as s_j^l and s_j^r, s_j^l, $s_j^r \in \{0, 1\}$, where s_j^l (s_j^r) represents the state whether or not the left (right) child of node j has updated its keys. When initializing the key tree S, $s_j^l = s_j^r = 0$ for all nonleaf nodes. At any time, if the key of node j's left (right) child is updated, SPA sets s_j^l (s_j^r) to 1.

```
Fix $d \leftarrow \log N$;
SUCCEED ← false;
$l \leftarrow$ DepthofNode($n$);
    if ($v_l = h(k_n, r_1, r_2) \vee v_l = h(tk_n, r_1, r_2)$)
        if ($l \neq d$)
            if $v_l = h(tk_n, r_1, r_2)$
                Record $n$ in Synchronization
                Message;
          for $i = 1$ to $\delta$
                $m \leftarrow$ FindChildren($n$, $i$);
                Identification ($U$, $m$);
        else if $l = d$
          SUCCEED ← true;
 if (¬SUCCEED)
        Fail and output 0;
else Accept and output 1;
```

Algorithm 1: Identification (U, node n). Tree-based identification.

```
if $n$ is a nonleaf node
      Store the old key $tk_n \leftarrow k_n$;
Generate a new key $k_n \leftarrow h(k_n)$;
$m \leftarrow$ FindParent($n$);
if $n$ is the left child of $m$
     Set $s_m^l \leftarrow 1$;
else if $n$ is the right child of $m$
     Set $s_m^r \leftarrow 1$;
if ($s_m^l = 1 \wedge s_m^r = 1$)
    Reset $s_m^l$ and $s_m^r$ to 0, and record $m$ in
    Synchronization message;
if $m$ is not the root node
     $n \leftarrow m$;
    Key-updating ($n$);
```

Algorithm 2: Key-updating (node n). Tree-based key-updating.

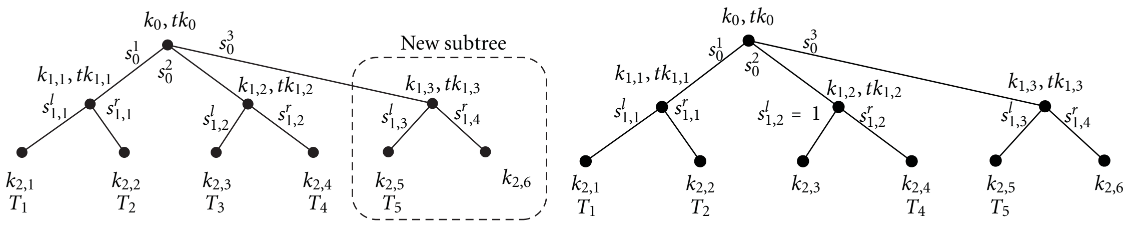
When R finishes key-updating, it sends a message $\sigma = h(k_d^i, r_1, r_2)$, as shown in Figure 4, and a *synchronization* message to T_i. The former one is used by T_i to authenticate R. The latter one includes the information of the levels on which the nodes have updated their keys in the key tree. After receiving these messages, T_i first verifies whether or not $\sigma = h(k_d^i, r_1, r_2)$. If yes, T_i updates its keys according to the synchronization message. For example, in Figure 3, suppose that R has updated keys $k_{1,1}$ and $k_{2,2}$ at levels 1 and 2 after identifying T_2. The synchronization message is (1, 2). Accordingly, T_2 updates $k_{1,1}$ as $k'_{1,1} = h'(k_{1,1})$ and $k_{2,2}$ as $k'_{2,2} = h(k_{2,2})$, respectively. This algorithm guarantees that the key-updating is consistent and feasible under arbitrary tag access patterns.

The key-updating algorithm is suitable for an arbitrary balanced tree with $\delta > 2$. In such a tree, there are δ state bits maintained in each nonleaf node to indicate the key-updating states of δ children.

3.6. System Maintenance. If a tag T_i is added to the RFID system, R checks whether or not there exists an empty leaf node in the key tree S. If yes, T_i is assigned to an empty leaf node. T_i's keys are then predistributed according to the path from the leaf node to the root of S. If there is no any empty leaf node in the tree S, R creates a new balanced tree S' with the branching factor δ and depth $d - 1$. R then initializes S' by employing the system initialization component described in Section 3.3.

After initialization, R connects the root of S to the root of S'. In this way, S' becomes a sub-tree of S. Hereby, T_i is assigned to an empty leaf node in S' and T_i's keys are pre-distributed according to the path from the leaf node to the root of S. For example, in Figure 5(a), R has 4 tags and each leaf node is occupied in R's key tree. If a new tag T_5 is added to the RFID system, R creates a new sub-tree and assigns a leaf node in this sub-tree to T_5. T_5's keys are k_0, $k_{1,3}$, and $k_{2,5}$.

For any empty leaf node i in the key tree, SPA lets the i's parent node j to set the corresponding state bit s_j to 1. Further, s_j is locked until the node i is assigned to a new tag T_i. This constraint is for protecting the key-updating of other tags from being interrupted. Indeed, since the node i is



(a) A new tag T_5 joins the RFID system

(b) A tag T_3 leaves the RFID system

Figure 5: Tag joining and leaving.

empty, no tag will trigger the change of s_j. Therefore, if s_j is 0, it will never change, such that node j will never update keys. To avoid such an interruption, SPA sets the s_j to 1 to allow other tags update their keys continuously and consistently.

If a tag leaves, R sets the leaf node of the leaving tag to be empty. For example, in Figure 5(b), if the tag T_3 leaves, R empties the leaf node distributed to T_3 and let $s_{1,2}^l = 1$.

3.7. An Example of Key-Updating. For the ease of understanding, we use an example to explain the key-updating algorithm. Here we show one reader R and four tags a, b, c, d (i.e., four leaf nodes in the key tree). We assume the sequence of tag authentication is (a, b, c, d, a). The original state of the key tree is shown in Figure 6(a). When tag a is identified, R sets the corresponding state bit of a's parent to 1. Meanwhile, R generates a new key of the leaf a as shown in Figure 6(b).

When tag b is identified, the corresponding state bit of b's parent is set to 1, and R updates the keys of the leaf node b and b's parent as shown in Figure 6(c).

When tag c is identified, the situation is similar to that of tag a. Since all state bits of the parent node of a and b are set to 1, R clears the state bits (i.e., reset the state bits to 0) in the key-updating operation as illustrated in Figure 6(d).

When tag d is identified, all state bits of d's parent and the root are set to 1. Thus, R updates the keys of the path from the leaf node of d to the root as shown in Figure 6(e).

Since tag a does not know that the keys of a's parent and the root have been updated, it will still use the old keys for identification. As indicated in the description of Algorithm 2 (Algorithm 2 in Section 3.5), each node stores the old key as the temporary key. After identifying tag a, R informs tag a to update its keys, according to the new keys from the leaf node of a to the root in the tree, as shown in Figure 6(f).

4. Discussion

In this section, we first discuss the security requirements for designing private authentication protocols in RFID systems. To evaluate the security of SPA, we propose an attack model to represent existing attacking scenarios. We then demonstrate the ability of SPA to meet those requirements and to defend against attacks.

4.1. Security Requirements. A private authentication protocol should meet the following security requirements [7].

Privacy. The private information, such as tag's ID, user name, and other private information should not be leaked to any third party during authentication.

Untraceability. A tag should not be correlated to its output authentication messages; otherwise, it may be tracked by attackers.

Cloning Resistance. Attackers should not be able to use bogus tags to impersonate a valid tag. Also, the replay attack should be resisted.

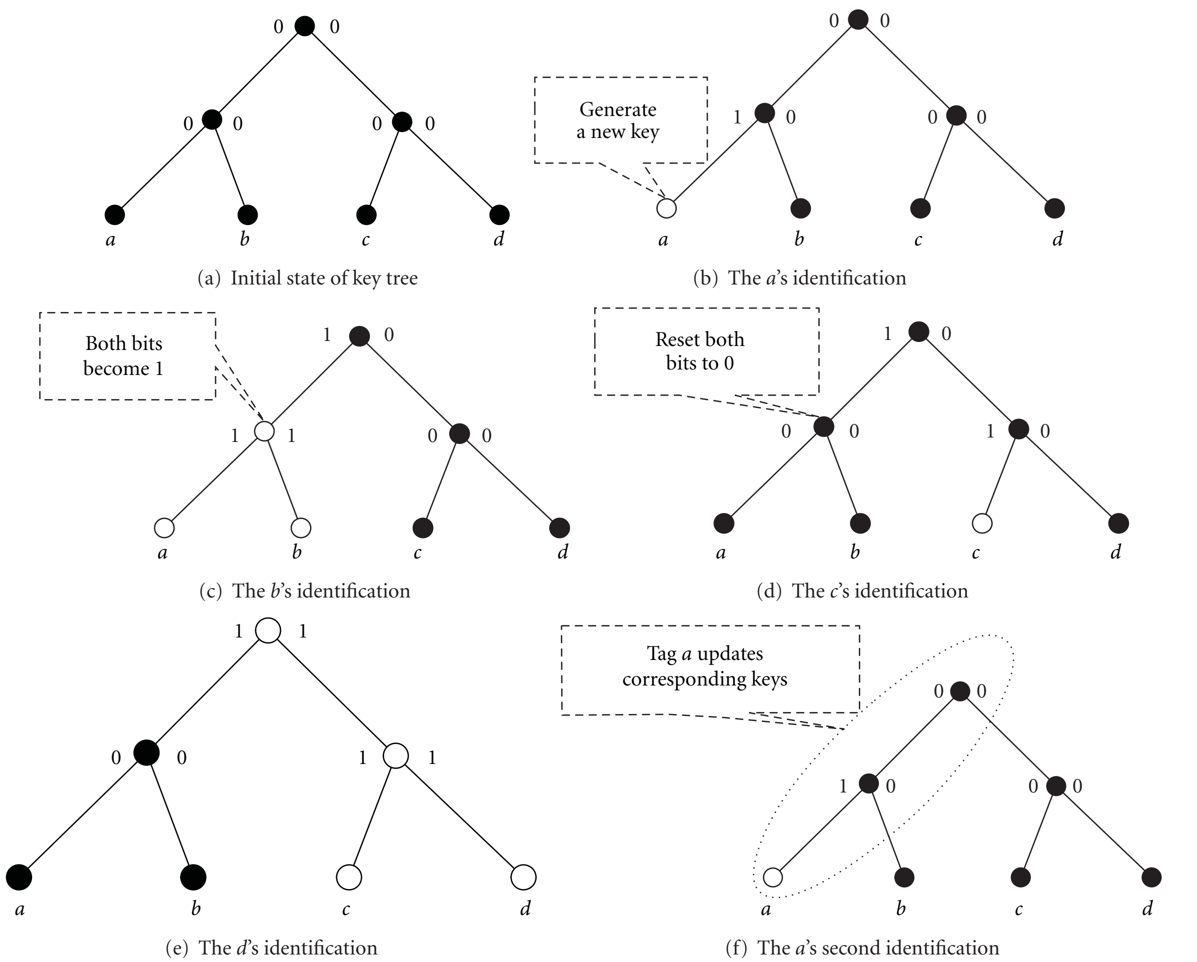
Forward Secrecy. Attackers can compromise a tag to obtain the keys stored in it. In this case, those keys should not reveal the previous outputs of the captured tag.

Compromising Resistance. The privacy of normal tags is threatened if they share some keys with compromised tags. Thus, the number of affected tags should be minimized after a successful compromising attack.

Existing private authentication approaches are able to defend against passive attacks (i.e., eavesdropping), but are vulnerable to active attacks (i.e., cloning and compromising attacks). Therefore, our discussion will focus on how SPA protects tags from active attacks. From the attacker's perspective, two metrics are important for evaluating the capability of SPA in defending against active attacks: (a) *past-exposing probability,* the probability of successfully identifying the past outputs of a compromised tag—this metric reflects the forward secrecy property of an authentication scheme; (b) *correlated-exposing probability,* the probability of successfully tracing a tag when some other tags in the system are compromised.

4.2. Attack Model. Avoine [17] proposes an attack model for RFID system and introduces the concept *untraceability*. The model well-reflects the attack behaviors and impacts on the authentication protocols. Our discussions are mainly based on this model with slight modification as follows.

In our model, the attackers and the RFID system are abstracted into two participants: the *Adversary A* (the attackers) and the *Challenger C* (the RFID system). So, the model is like a game between A and C. A first informs C that A will start to attack. C then chooses two tags to perform SPA protocols. If A can successfully distinguish any



(a) Initial state of key tree

(b) The *a*'s identification

(c) The *b*'s identification

(d) The *c*'s identification

(e) The *d*'s identification

(f) The *a*'s second identification

Figure 6: An example of the key-updating procedure.

tag from another one based on their outputs by performing passive or active attacks on the RFID system, we claim that A successfully compromises the privacy of the system. For simplicity, Let P denote the SPA authentication procedure.

We define four oracles, *Query, Send, Executive,* and *Reveal*, to abstract the attacks on each T or R. Thus, any attack on a given R or T can be represented as A's calling on its oracles.

Query (T, m_1, m_3). A sends a request m_1 to T. Subsequently, A receives a response from T. R then sends the message m_3 to T. Note that m_1 and m_3 represent the messages sent from A in the first round and third round in a SPA authentication procedure, respectively.

Send (R, m_2). A sends a message m_2 to R and receives a response. Note that m_2 represents the message sent from A in the second round in a SPA authentication procedure.

Execution (T, R). A executes an instance of P with T and R, respectively. A then modifies the response messages and relays them to both sides accordingly.

Reveal (T). After accessing this oracle of T, A compromises T, which means A obtains T's keys. Note that A can distinguish any given tag T from other tags if it can obtain T's keys.

We claim that an authentication protocol is resistant to attacks A-O, if the protocol is resistant to A when the adversary has access to the oracles $O \subset \{Query, Send, Executive, Reveal\}$. Based on these oracles, the detailed procedure of a game between A and C is formalized by following steps.

A tells C that the game begins. C chooses two tags T_0 and T_1.

For two tags T_0 and T_1 chosen by C, A accesses the oracles of T_0 and T_1. For T_0 and T_1, let O_{T_0} and O_{T_1} denote the sets of accessed oracles, respectively.

C selects a bit $b \in \{0, 1\}$ uniformly at random, and then provides the oracles of the corresponding tag T_b for A to access. For simplicity, we denote T_b as T. A then accesses T's oracle. Let the set of accessed oracles of T, be O_T.

Based on the results from O_{T_0}, O_{T_1} and O_T, A outputs a bit b'. If $b' = b$, A successfully distinguishes T_0 and T_1; otherwise, A loses. Note that A can access the oracles in O_{T_0}, O_{T_1}, and O_T in polynomial times. Indeed, if A can distinguish T_0 from T_1, it means that A can track all tags in an RFID system.

A passive adversary who can only eavesdrop on the messages delivered by T or R has no access to any oracle. For an active adversary, it can access arbitrary oracles introduced above. For instance, if an adversary applies a cloning attack, it means that it can access the *Query*, *Send*, and *Executive* oracles in this attack model. If an adversary can apply compromising attack, it means that it can access the *Reveal* oracle.

4.3. Security Analysis. In this subsection, we show how SPA achieves the security goals.

Privacy. Due to the pseudorandomness and one-way properties of the cryptographic hash functions, it is safe to claim that the output of the hash function can be seen as a random bit string. Note that the pseudo-randomness of a hash function means no adversary can distinguish the output of the hash function from a real random bit string. Therefore, the messages sent by the reader and tags provide no useful information to an adversary. None of the passive adversaries is able to deduce the original messages based on the output of hash functions, unless it can invert the hash function. It is well known that the probability of inverting a hash function is negligible.

Untraceability. Since the authentication of SPA does not enroll the ID of a tag and all authentication messages are encrypted by a cryptographic hash function, any passive adversary cannot distinguish the tags from others based on their encrypted messages. That is, it cannot track a tag.

Cloning Resistance. In a cloning attack, an adversary monitors a tag, records its messages and resends the message repeatedly [9]. Similar to most previous protocols, in SPA, the reader and the tags generate random numbers r_1 and r_2 to defend against the cloning attack. Since the random numbers r_1 and r_2 are generated uniformly at random, the adversary has to guess them to recover the content of the messages. If the length of r_1 (r_2) is sufficiently long (e.g., 40 bits), the probability of successfully guessing the random numbers is negligible. Thus, SPA is not subject to the cloning attack.

Forward Secrecy. If a tag is captured, the adversary might obtain the tag's current keys. However, the adversary cannot trace back the tag's past communications because the keys are updated by a cryptographic hash function in each authentication procedure. In this way, the adversary still cannot retrieve the past outputs of the tag, unless it is able to invert the cryptographic hash function. Therefore, we can consider that the past-exposing probability of SPA upon the forward secrecy approaches 0. On the contrary, the static key tree protocols [7, 8, 11] cannot update the keys in the system. When an adversary compromised a tag, it can identify the past outputs of the tag from the obtained keys. Thus, the past-exposing probability of the key tree protocols approaches 1.

4.4. Compromising Attack. As we discussed in Section 3.1, a compromised tag may reveal some of the keys of other tags in static tree-based protocols. The adversary is then aware of some paths from the root to the leaf nodes of the compromised tag. Based on those paths, the adversary partially compromises the tree infrastructure. Knowing the "positions" of those nonleaf nodes, the adversary can further identify a sub-tree to which T_i might belong.

Now we use the attack model to discuss the impact of a compromising attack on SPA. The following analysis is based on Avoine's work [18]. The game procedure comprises six phases.

Phase 1. The adversary A has compromised t tags and obtained their complete secret keys.

Phase 2. The system (*challenger*) C chooses two normal tags T_0 and T_1.

Phase 3. A calls oracles O_{T_0} and O_{T_1} (except the *Reveal* oracle), and then receives the results (Note that A cannot compromise T_0 and T_1).

Phase 4. C selects a bit $b \in \{0, 1\}$ uniformly at random, and then provides the oracle O_T (denote T_b as T) to A for accessing (except the *Reveal* oracle).

Phase 5. A calls oracle O_T (except the *Reveal* oracle) and receives the results.

Phase 6. A outputs a bit b', if $b' = b$, A has successfully distinguished T_0 or T_1 from another. Otherwise, A loses.

We assume that A cannot carry out an exhaustive search over the key space. Suppose that A has compromised t tags except T_0 and T_1. Thus, A is aware of several paths from the root to the leaf nodes of those tags, as well as the relevant keys of the nonleaf nodes on those paths. Let M denote the set of those compromised nonleaf nodes in key tree. Let M_i denote the subset of M in which the nodes have the same level i in the key tree. Clearly, $M = \bigcup_{i=1}^{d} M_i$. Correspondingly, let $\overline{M_i}$ denote the set of nodes at level i which have not been compromised by A.

In Phase 5, A impersonates the reader and queries T, T_0, and T_1 with the keys obtained in the RFID system. At this point, there are three possible cases: (1) If neither T_0 nor T_1 has a nonleaf in M, A completely fails. (2) if either T_0 or T_1 (but not both) has a nonleaf node in M, the key subset of T_0 or T_1 including all the keys from the root to this nonleaf node have been compromised. Thus, the adversary can determine which one is T and obtain a correct answer in Phase 6. In this case, A succeeds. (3) If both T_0 and T_1 have an identical nonleaf node in M, A cannot directly distinguish T_0 or T_1 from another. In this case, A can move down to the lower level of the key tree from the current nonleaf node. We denote the keys of T, T_0, and T_1 by $[k_0, \dots, k_d]$, $[k_0^0, \dots, k_d^0]$, and $[k_0^1, \dots, k_d^1]$, respectively, where d is the depth of the key tree. For a given level i, suppose two nodes $n_{i,0}$ and $n_{i,1}$ have an identical parent $n_{i-1,0}$ at the lever $i - 1$, and their keys are

k_i^0 and k_i^1, respectively. Let S_{i-1} denote the sub-tree of key tree S rooted at $n_{i-1,0}$. Thus, $n_{i,0}$ and $n_{i,1}$ are both in S_{i-1}. Let K_i denote the set of keys of the nodes in the interaction of $S_{i-1} \cap M_i$. Let V_i denote the set of the nodes in the interaction of $S_{i-1} \cap \overline{M_i}$. For example, suppose that R maintains a key tree with eight leaf nodes in Figure 7. A has compromised tags T_3, T_5, and T_8. In this case, for the sub-tree S_1, $K_i = \{k_{2,2}, k_{2,3}, k_{2,4}\}$ and $K_i = \{k_{2,1}\}$. Let t_i be the number of keys in K_i, and let δ be the branching factor of the key tree. We also denote a as the number of keys belonging to a nonleaf node (in SPA, any nonleaf node stores two keys k and tk, therefore $a = 2$). We consider the following five cases.

Case 1. If $C_i^1 = ((k_i^0 \in K_i) \wedge (k_i^1 \in V_i))$, then A succeeds.

Case 2. If $C_i^2 = ((k_i^0 \in V_i) \wedge (k_i^1 \in K_i))$, then A succeeds.

Case 3. If $C_i^3 = ((k_i^0 \in K_i) \wedge (k_i^1 \in K_i) \wedge (k_i^0 \neq k_i^1))$, then A succeeds.

Case 4. If $C_i^4 = ((k_i^0 \in V_i) \wedge (k_i^1 \in V_i))$, then A definitively fails.

Case 5. If $C_i^5 = ((k_i^0 \in K_i) \wedge (k_i^1 \in K_i) \wedge (k_i^0 = k_i^1))$, then A fails at level i but it can move to level $i+1$ to continue its attack.

For $1 \le i \le d$, we have

$$\Pr\left[C_i^1\right] = \Pr\left[C_i^2\right] = \frac{t_i}{a\delta}\left(1 - \frac{t_i}{a\delta}\right),$$
$$\Pr\left[C_i^3\right] = \left(\frac{t_i}{a\delta}\right)^2\left(1 - \frac{1}{t_i}\right), \qquad (1)$$
$$\Pr\left[C_i^5\right] = \left(\frac{t_i}{a\delta}\right)^2 \cdot \frac{1}{t_i}.$$

Therefore, $\Pr[C_i^1 \vee C_i^2 \vee C_i^3] = (t_i/(a\delta)^2)(2a\delta - t_i - 1)$.

The probability that A succeeds is given by

$$\begin{aligned}
&\Pr\left[\text{Attack Succeeds}\right] \\
&\quad = \Pr\left[C_1^1 \vee C_1^2 \vee C_1^3\right] \\
&\qquad + \sum_{i=2}^{d}\left(\Pr\left[C_i^1 \vee C_i^2 \vee C_i^3\right] \times \prod_{j=1}^{i-1}\Pr\left[C_j^5\right]\right) \\
&\quad = \frac{t_1}{(a\delta)^2}(2a\delta - t_1 - 1) \\
&\qquad + \sum_{i=2}^{d}\left(\frac{t_i}{(a\delta)^2}(2a\delta - t_i - 1) \times \prod_{j=1}^{i-1}\frac{t_i}{(a\delta)^2}\right).
\end{aligned} \qquad (2)$$

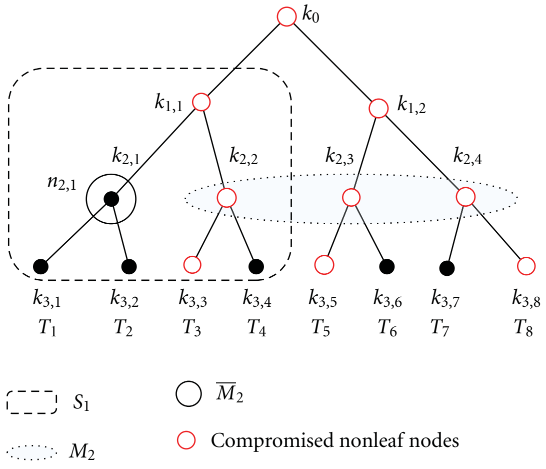


FIGURE 7: An example of the compromising attack.

In (2), t_i, the number of keys known by the adversary at level i, is given by

$$t_1 = \delta\left(1 - \left(1 - \frac{1}{a\delta}\right)^t\right),$$
$$t_i = \delta\left(1 - \left(1 - \frac{1}{a\delta}\right)^{f(t_i)}\right) \quad 1 < i \le d, \qquad (3)$$

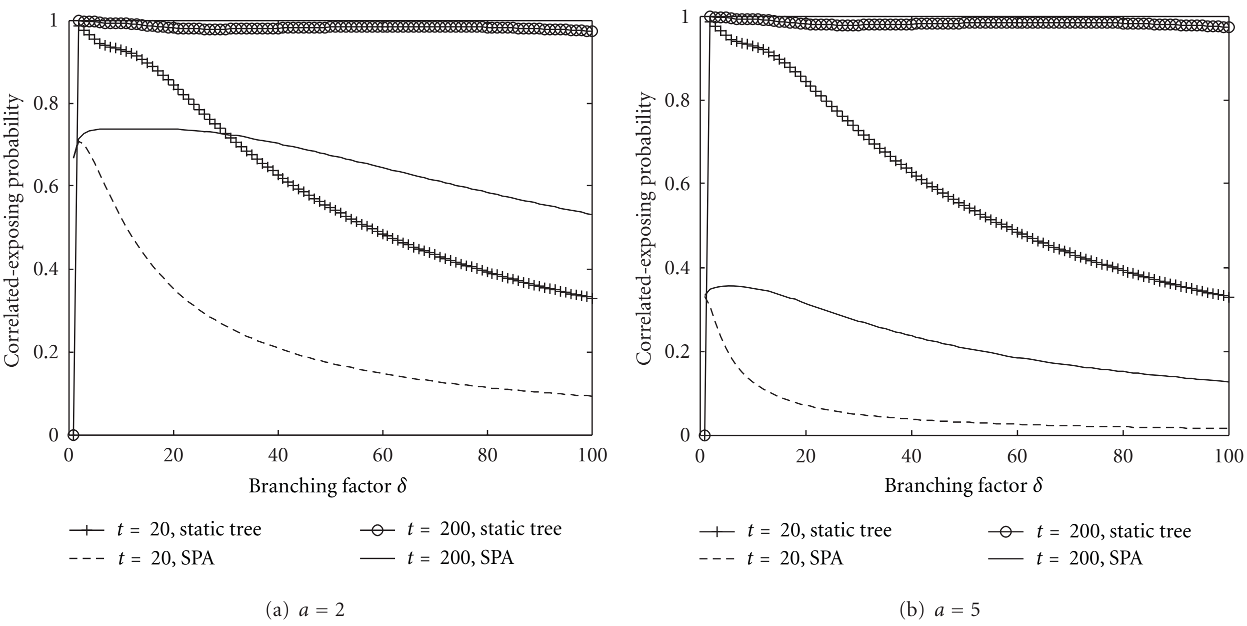
where $f(t_i) = t\prod_{j=1}^{i-1} 1/t_j$.

Equation (2) shows that the correlated-exposing probability is mainly determined by three key parameters: (a) t, the number of compromised tags; (b) δ, the branching factor of the key tree; (c) a, the number of keys belonging to each nonleaf node. Note that if $a = 1$, (2) can also be used to evaluate the security of static tree-based approaches. In Figure 8, we show the theoretical evaluation on the security of SPA in a typical RFID system.

We assume that the system contains 2^{20} tags and the depth of key tree is 20. In the worst case, the adversary A can *simultaneously* compromise t tags at a given time. Then, A immediately starts attacks following the game strategy with challenger C. In addition, we assume there are only T_0 and T_1, which are chosen by C, performing authentication with the reader at this moment. Thus, we can use (2) to compute the correlated-exposing probability for A attacking SPA and static tree-based approaches.

As shown in Figure 8, SPA outperforms static tree-based approaches in defending against compromising attacks. In SPA, although A captures a number of keys shared by some normal tags, those tags are still secure if they update their keys. In contrast, normal tags in static tree-based approaches would be more vulnerable because the keys obtained by A will still be in use. This would ease A's tracking attempts.

In both SPA and static tree-based approaches, the correlated-exposing probability is reduced when enlarging the branching factor δ. This is because enlarging δ leads



(a) $a = 2$

(b) $a = 5$

Figure 8: Defending against the compromising attack.

attackers to capturing fewer keys shared by tags which are not tampered with.

The static tree base approaches are extremely vulnerable to compromising attacks when t is sufficiently large. We find the correlated-exposing probability is close to 1 when $t = 200$ in static tree-based approaches. In this case, enlarging δ does not help much. On the contrary, SPA can decrease the probability by increasing a. The curves of $t = 200$ in Figure 8 show that SPA is more secure under compromising attacks and flexible enough to meet different security concerns.

We here explain why increasing a can enhance the security of SPA. In an RFID system, a part of tags may not be accessed by the reader for a long time. Keys in those tags hereby cannot be updated. The reader must maintain these keys in nonleaf nodes' temporal keys for future use. As discussed in Section 3.2, temporal keys tks store those old keys. The working key k is computed from the temporal keys after these temporal keys have been updated. Therefore, keys of some nonleaf nodes will not be updated if a number of tags are not identified by the reader for a long time. For example, we assume that each nonleaf node has only a single temporal key. If some tags lie in the sub-tree are not accessed by the reader in a period, other tags belong to this node's sub-tree can update keys for only once. That will make the key tree in SPA degenerate into the static tree, thereby alleviate SPA's resilience to compromising attacks. To ease the impact of this problem, we increase the number of temporal keys. We assume that each nonleaf node has one working key k and $a - 1$ temporal keys, $tk_1,\ldots, tk_{a-1}$. Thus, even if a number of tags are not accessed by the reader, keys of other tags in the system can be updated for at least $a - 1$ times. When we enlarge the parameter a, we can enhance the system's capability on resilience to compromising attacks, as shown in Figure 8.

5. Prototype Implementation

In this section, we introduce our experience on SPA prototype implementation. We also evaluate the performance of SPA and compare it with existing private authentication protocols.

5.1. Experiment Setup. We have implemented the SPA protocol on Mantis-series 303 MHz asset tags and Mantis II reader manufactured by RF Code [19]. In terms of its coding, this system is able to support over one trillion tags. A tags' typical transmission range is 300 feet, and the reader can communicate with them at distances of more than 1000 feet depending on the antenna configurations. The back-end database is implemented on the desktop PC with the following configurations: Pentium M 3.2 G dual core CPU, 1 GBytes memory, and 40 G hard disk. We use SHA-1 algorithm as the secure hash function.

In this implementation, the system is able to maintain up to $N = 2^{20}$ tags. For each test, we randomly distribute 40 tags into leaf nodes in the key tree. We perform 1000 independent runs and report the average. We employ a balanced binary tree as the key tree. Each nonleaf node is assigned with two keys, that is, $a = 2$. The length of each key is 64 bit, which is sufficiently long to resist brute-force attacks.

A fundamental concern upon SPA is the latency of key-updating. We use the metric *key-updating Latency* as the time required for the reader to update a tag's keys to evaluate the performance of SPA. On the other hand, the key-updating of SPA should guarantee that the keys are secure enough over their lifetime. We focus on two metrics in the experiments.

(a) Key-Updating Latency. It reflects the computational overhead of key-updating. In SPA, the whole processing

Table 1: Experiment settings.

Parameters	Values
δ	2
a	2
d	20
l	64
p	0.1
a_f	10

overhead for an authentication procedure includes two components: *identification* and *key-updating*. Since the static key tree approaches only perform the identification function, we focus more on the computational overhead caused by key-updating of SPA.

(b) Key Security Degree (KSD). It reflects the possibility of keys being obtained when an adversary attacks a RFID system. Let f denote the tag accessing frequency, which means how many tags interact with the reader per second. Let l be the length of a key, let n be the number of updated keys in one key-updating procedure, and let d be the depth of the key tree. We denote p as the probability of an adversary successfully gaining the n keys, and a_f as the attack frequency (i.e., the number of attacks occurred per second). Thus, the key security degree KSD is defined by

$$\mathrm{KSD} = \frac{f \times n \times l \times d}{a_f \times p}. \tag{4}$$

Because the KSD computed from (4) will be a large value, we use a small real number ε to make the KSD value not too large, where $\varepsilon = 0.0001$ is a system parameter.

KSD reflects the comprehensive ability of a protocol on defending against the active attacks. A higher KSD value represents a more secure protocol. Parameters in our experiments are summarized in Table 1.

5.2. Results. Figure 9 plots the average key-updating latency of SPA. With the increase of the tag accessing frequency, which means how many times a tag is accessed per second, the key-updating latency increases. The processing speed of SHA-1 is 1.73 MByte per second. We find that the latency of key-updating does not exceed 1.7 ms even when the tag accessing frequency approaches 10. Since we construct a tree with the depth of 20 in this experiment, each tag is assigned with 20 keys. Thus, the curve of key-updating is enclosed within two lines: one represents the upper bound (20 keys in a tag are updated) and another represents the lower bound (only one key is updated). The short key-updating latency of SPA enables a reader to support dense access patterns. Due to page limitation, results from other experiments are not reported here.

Figure 10 shows the change of KSD of SPA with different tag access frequencies. The curve of KSD fluctuates when increasing the frequency. In Figure 10, two lines show the upper and lower bounds ($n = 20$ and $n = 1$) of the KSD curve of SPA. We can see that the KSD of SPA increases when enlarging the tag accessing frequency. Clearly, the design of SPA has two advantages. First, tag holders do not need to update the keys of their tags specially. Second, the RFID system is highly secure when tag holders use their tags frequently.

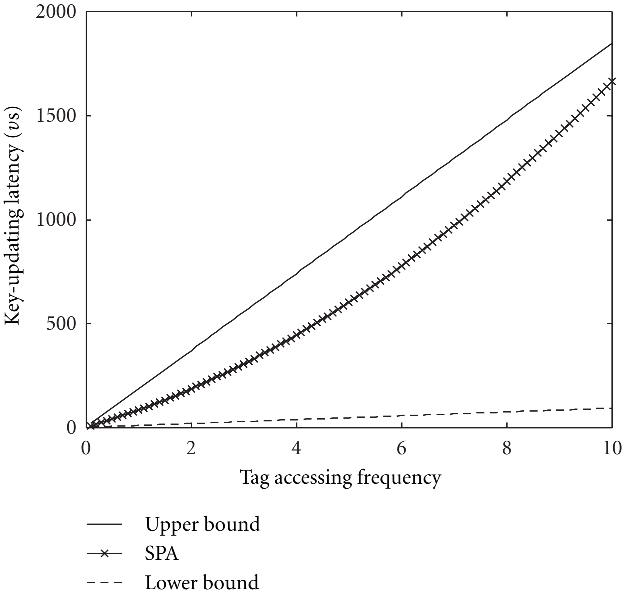


Figure 9: Key-updating latency of SPA.

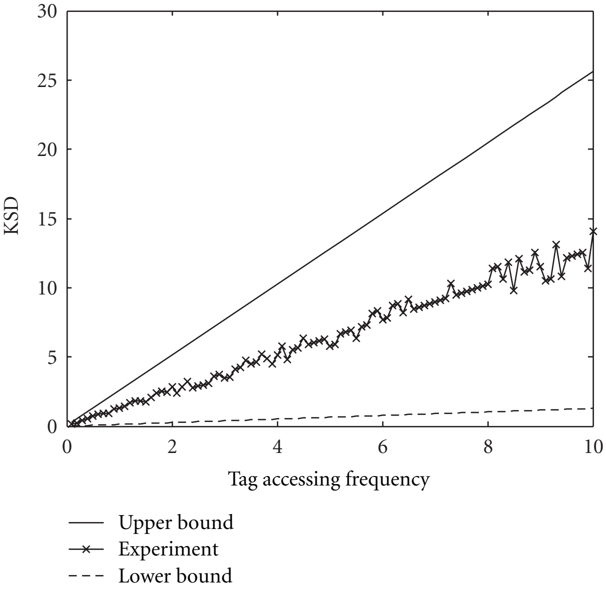


Figure 10: Key security degree of SPA.

In SPA, the main overhead is caused by SHA-1 computations on the side of R. Therefore, the number of SHA-1 computations reflects the computation overhead of SPA. Figure 11 shows the relationship between the computation overhead and KSD in different tag accessing frequencies. In Figure 11, the y-axis refers to the number of

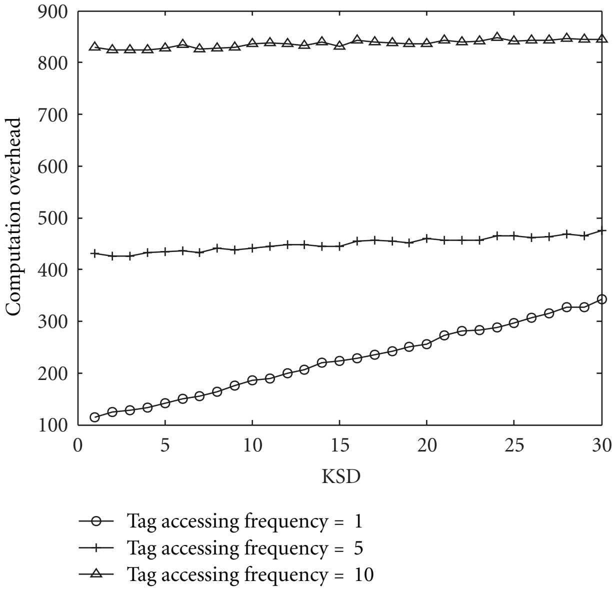


FIGURE 11: Computation overhead versus KSD.

SHA-1 computations. We see that a high tag access frequency results in a smooth curve of overhead. We also find that the computation overhead caused by SPA does not exceed 900 times of SHA-1 even the tag accessing frequency is high, for example, R accessing 10 tags per second, while the PC used in our experiments can perform 86,500 SHA-1 computations per second. It indicates that a significant incensement of KSD only incurs small computation overhead to the system. Hence, SPA is scalable when providing high secure private authentication services.

6. Conclusion

We proposed a privacy-preserving authentication protocol, SPA, to support secure and efficient tag-reader transactions in RFID systems. By using a dynamic key-updating algorithm, SPA enhances the security of existing RFID authentication protocols. SPA is lightweight with high authentication efficiency: a reader can identify a tag within $O(\log N)$ tree walking steps. Compared with previous works, SPA can effectively defend against both passive and active attacks. Through the prototype implementation, we demonstrated that SPA is scalable and practical in large-scale RFID systems.

Acknowledgment

This research was supported by NSFC 60903155 and NSFC 61173171.

References

[1] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proceedings of the1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pp. 407–415, March 2003.

[2] Y. Li and X. Ding, "Protecting RFID communications in supply chains," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS'07)*, pp. 234–241, March 2007.

[3] T. Kriplean, E. Welbourne, N. Khoussainova et al., "Physical access control for captured RFID data," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 48–55, 2007.

[4] C. Qian, H.-L. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'08)*, pp. 30–39, March 2008.

[5] L. Xiao, Y. Liu, W. Gu, D. Xuan, and X. Liu, "Mutual anonymous overlay multicast," *Journal of Parallel and Distributed Computing*, vol. 66, no. 9, pp. 1205–1216, 2006.

[6] P. Robinson and M. Beigl, "Trust context spaces: an infrastructure for pervasive security in context-aware environments," in *Proceedings of International Conference on Security in Pervasive Computing (SPC'03)*, 2003.

[7] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'06)*, pp. 269–274, March 2006.

[8] D. Molnar and D. Wagner, "Privacy and security in library RFID issues, practices, and architectures," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04)*, pp. 210–219, October 2004.

[9] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Proceedings of International Conference on Security in Pervasive Computing (SPC'03)*, 2003.

[10] A. Juels, "RFID security and privacy: a research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[11] D. Molnar, A. Soppera, and D. Wagner, "A scalable, delegatable pseudonym protocol enabling owner-ship transfer of RFID tags," in *Proceedings of the Selected Areas in Cryptography (SAC'05)*, 2005.

[12] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Efficient hash-chain based RFID privacy protection scheme," in *Proceedings of the UbiComp, Workshop Privacy*, 2004.

[13] A. Juels, "Minimalist cryptography for low-cost RFID tags," in *Proceedings of International Conference on Security in Communication Networks (SCN'04)*, 2004.

[14] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm'05)*, pp. 59–66, September 2005.

[15] G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom'05)*, pp. 110–114, March 2005.

[16] M. E. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Transactions on Information Theory*, vol. 26, no. 4, pp. 401–406, 1980.

[17] G. Avoine, "Adversarial model for radio frequency identification," Report 2005/049, 2005, Cryptology ePrint Archive.

[18] G. Avoine, E. Dysli, and P. Oechslin, "Reducing time complexity in RFID systems," in *Proceedings of Selected Areas in Cryptography (SAC'05)*, 2005.

[19] RFCode, Inc., http://www.rfcode.com/Solutions/Asset-Management/Products-Overview.html.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 865184, 9 pages
doi:10.1155/2012/865184

Research Article

Improving Accuracy for 3D RFID Localization

Jinsong Han,[1] Yiyang Zhao,[2] Yan Shun Cheng,[3] Tse Lung Wong,[3] and Chun Hung Wong[3]

[1] *School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2] *School of Software, TNLIST, Tsinghua University, Beijing 100084, China*
[3] *Department of Computer Science and Engineering, HKUST, Kowloon, Hong Kong*

Correspondence should be addressed to Jinsong Han, hanjinsong@mail.xjtu.edu.cn

Received 15 December 2011; Accepted 4 February 2012

Academic Editor: Mo Li

Radio Frequency Identification (RFID) becomes a prevalent labeling and localizing technique in the recent years. Deploying indoor RFID localization systems facilitates many applications. Previous approaches, however, are most based on 2D design and cannot provide 3D location information. The lack of one-dimensional information may lead 2D-based systems to inaccurate localization. In this paper, we develop an indoor 3D RFID localization system based on active tag array. In particular, we employ the geometric mean to filter the explicit 3D location information with high accuracy. The experimental results show that our system is efficient in tracking objects and improving the localization accuracy.

1. Introduction

As a more efficient way of identifying objects than conventional Barcode technique, Radio Frequency Identification (RFID) technology has been popular in many applications, such as surveillance, access control, theft prevention, and movement tracking [1–4]. An RFID system comprises of a number of readers and a large amount of tags. The tag usually stores an ID that is uniquely assigned. When entering or staying in the interrogation area of readers, a tag can report its ID to the reader via RF signals. The reader can retrieve the data sent from the tag with predefined radio frequency and protocols. In practice, the frequency ranges between 135 KHz, 13.56 MHz, UHF (433 MHZ, for active tags, 860 to 960 MHz for passive tags), and 2.45 GHz. There are two main types of tags, passive tags and active tags. Passive tags absorb energy from the RF wave emitted from the reader and hence do not have onboard energy supply. On the contrast, active tags are equipped with onboard power supply, usually using button-cell batteries, so that they can actively operate in a longer communication range than passive tags. The RFID technology offers very attractive advantages. For example, using RFID can achieve a noncontacting and non-line-of-sight processing pattern, which allows the system to automatically collect the data from items with which the tags are attached or embedded.

The RFID can play many important roles in existing applications. In this paper, we are interested in using active RFID tags for 3D localization. Traditionally, if we plan to monitor a large area, an intuitive solution is to deploy many cameras for surveillance. However, the visual-based monitoring system suffers from several drawbacks. A camera could only monitor in a fixed direction. If we are aiming at a full coverage of monitoring over a large region, many cameras may be deployed, which renders high cost. Meanwhile, the detection of objects' complete trajectories is difficult within large areas. Deploying video systems may cause some gaps or holes in the coverage of surveillance, which will produce discontinuous trajectory reporting. Last but not least, the objects' trajectories should be predefined if using video systems. Detecting nonregular or unpredicted moving objects is thereby nontrivial.

As utilizing the RSSI in wireless networks for localization [5, 6], the strength (and its changes) of RF signals could be utilized to locate and hence track objects. On the other hand, the readers and tags emit their RF signal in an omni-directional way, which closes the gap of monitoring coverage. In addition, the deployment pattern of RFID systems is more flexible than that of video-based systems. Redeployment may be not necessary even if the object moves very abnormally or unpredictably.

Many efforts have been made for locating objects based on RSSI or RF signals [2, 3, 7–11]. Most of them, however, focus on two-dimensional (2D) scenarios. In practice, however, many applications are deployed in 3D terrains, such as an inventory management system deployed in a warehouse with many shelves, a structural monitoring network mounted on an aged building, or a surveillance system implemented in a theater or gym, and so forth. Missing any coordinate in the 3D space would seriously constrain the accuracy or completeness of localization.

In this paper, we propose to break above limitation and extend the 2D localization to 3D scenarios. We implement RF-based 3D localization system with active RFID tag array based on a classical 2D localization work, VIRE [7]. VIRE utilizes reference tags for locating a target tag. Although the VIRE algorithm eliminates unlikely positions, it is still computationally inefficient, especially when being extended to 3D implementation. Our approach adopts a simple mechanism, geometric mean, to improve the efficiency of eliminating unlike positions of tracking tags, which also improves the efficiency of 3D localization when using the RF reference tags. The experimental result demonstrates the effectiveness and efficiency of our system.

The rest of the paper is organized as follows. Section 2 briefly discusses the related works. We present our design in Section 3 and implementation in Section 4. We evaluate the performance of our system in Section 5 and conclude in Section 6.

2. Related Work

As a promising wireless identification technology, RFID has been increasingly important in both the research and industry communities [2, 3, 12–15]. The research issues of RFID study in the literature focus on the anticollision, security, and localization.

Collisions significantly lag the identification speed in RFID systems. There are two types of signal collisions: *tag collision*, which may occur when more than one tag responds simultaneously; *reader collision*, which will happen when a region is overlapped by two or more readers' scanning signals. Anti-tag-collision approaches are divided into two categories, Framed-Slotted-ALOHA- (FSA-) based and Binary-Tree- (BT-) based algorithms. Roberts [16] first proposes an FSA-based anticollision scheme. Later, Lee et al. [17] claims that a maximum identification throughput can be achieved when the size of frame equals the number of tags. EPC Gen2 [18], a widely deployed RFID standard, adopts "Q-Adaptive" to adaptively adjust the frame length according to previous slots. Another well-known RFID air protocol, ISO 18000-6 [19], employs the Binary-Tree- (BT-) based RFID identification protocol. Hush and Wood [20] analyze the throughput of BT-based algorithms in [21]. Myung and Lee [22] then propose an adaptive binary splitting (ABS) protocol to reduce collisions. Besides tag collisions, reader collision is also challenging in large-scale RFID deployments. Existing solutions focus on assigning different channels to adjacent readers [23] or scheduling their interrogations into different rounds [24].

Meanwhile, some approaches [25, 26] have been proposed to achieve authentication in RFID applications. To prevent tags from being tracked, Weis et al. propose a hash-function-based authentication scheme, Hash Lock [27]. Hash Lock, however, has a slow authentication speed due to its $O(N)$ key search complexity, where the N is the total number of tags in the system. Some subsequent attempts aim at reducing the cost of key search. Different from the linear key organization, tree-based structures can reduce the search complexity from $O(N)$ to $O(\log N)$ [25]. Choi and Roh propose such a scheme [26]. Lim et al. [28] presents a randomized-bit-encoding scheme to enhance the security protection for RFID tags. The RFID privacy models have been extensively studied in [29].

The employment of radio signal strengths for indoor localization is very popular. Besides the identification, RFID tags can also facilitate the localization by using their RF signals. For the RFID-based localization, LANDMARC is one of the pioneering works. LANDMARC [2] introduces the reference tags and predefines the location map of reference tags to facilitate tracking tags. In LANDMARC, four reference tags that are nearest to the tag attached to the targeted object. The RSSI reported from those tags will be compared to predefined map and hence to approximate the location of the tag tracked. Later, VIRE [7] is proposed as an advance method based on LANDMARC. In VIRE, four reference tags form a grid, and some reference tags will be interpolated in the grid. Then a proximity map is obtained after collecting the results by the RFID readers. The algorithm of VIRE estimates and filters unlikely positions by intersection. Compared with LANDMARC, VIRE is more accurate but with more computational overhead. To our knowledge, there are little efforts focusing on the 3D RFID localization. Utilizing resource-limited wireless devices, for example, the RFID tag, for 3D localization is still difficult.

3. 3D RFID Localization Design

In this section, we present the design of our system. Our system works in following steps: (1) system deployment, (2) RSSI data collection and analysis, (3) tracking targeted tags, and (4) refining the location of targets. First, we establish the tag array in the deploying region. We then collect the strength of RF signals in the initialized system. After an extensive analysis, the strength distribution of RF signals can be retrieved. We design an efficient algorithm for 3D localization based on the signal strength to track targeted tags. Last, we refine the location regions for targets by using a simple but effective condensation process, the geometric mean calculation.

3.1. System Deployment. We employ the active tags, M100, manufactured by RF Code [30], to implement a prototype framework. The active tags work on the 433 MHz and have a transmission range up to 300 feet. Figure 1 shows the deployment of RFID readers and tags in the tag array system.

We deploy a 3D array of RF tags in the monitoring field. The active tag will broadcast a probing message, including its unique ID and status information at each time unit (we term

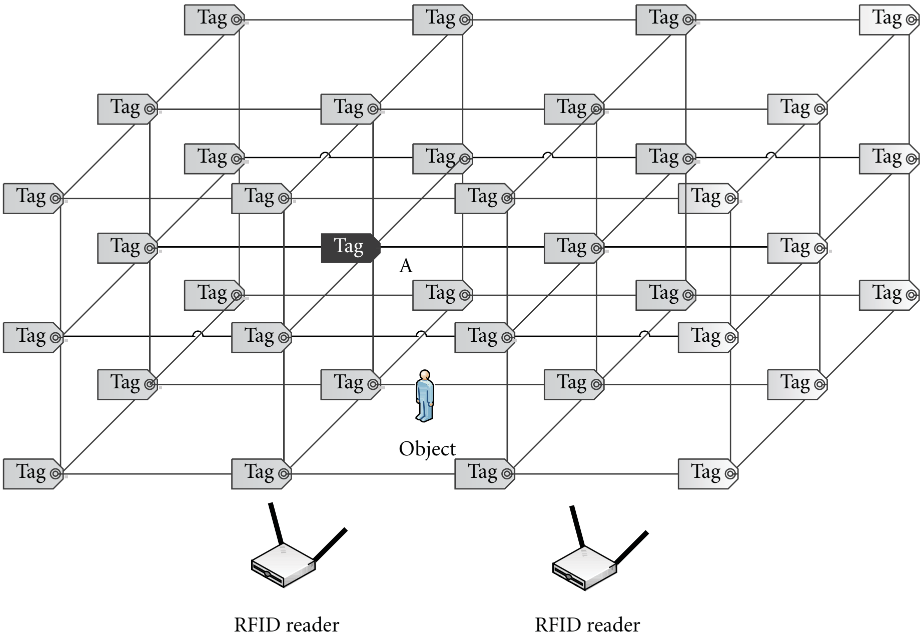


Figure 1: Tag array deployment.

such a unit as *period* in this paper). The beacon rates for these tags range from 2 seconds to 10 seconds, based on the user specification or application requirements. There is no synchronization mechanism among the tags. All the tags compete for the transmission window.

Besides the tags, we also deploy a small number of readers (in our preliminary system, the number is 2) for collecting the signals emitted from tags. Note we do not need to deploy too many readers because the reading range of readers is relatively large. The reader can cover a distance of more than 1,000 feet. In practice, the reading range depends on the antenna configuration. The readers are connected to a server within a LAN. In our system, the reader can collect the essential information about tags, including the tag ID, the reader ID, and RSSI.

3.2. RSSI Data Collection and Analysis. Before performing the localization, the system first reads tags' RSSI from the reader and store in database together with their physical location coordinates. The gathered tags' RSSI values together with the pre-defined reference tags' coordinates, as shown in Figure 2, are stored into our database for later localization and further analysis.

However, the RSSI values collected by the reader are raw and messy. Some of them fluctuate a lot. We filter the raw data to make it more accurate and useful. There are two antennas in each reader. The reader will report that two RSSI values for each tag correspond to the two antennas. Sometimes the signal cannot be detected due to the interference. We adopt the following policy to maintain a good estimation of the RSSI. If both antennas report values, we take the average. If only one of the antennas provides a value, we take it as the resultant value. If both antennas do not provide a value, we take a value of −1 to label that tag as not-detected, as shown in Figure 3. The smaller the RSSI value is, the longer of the distance between the tag and the readers.

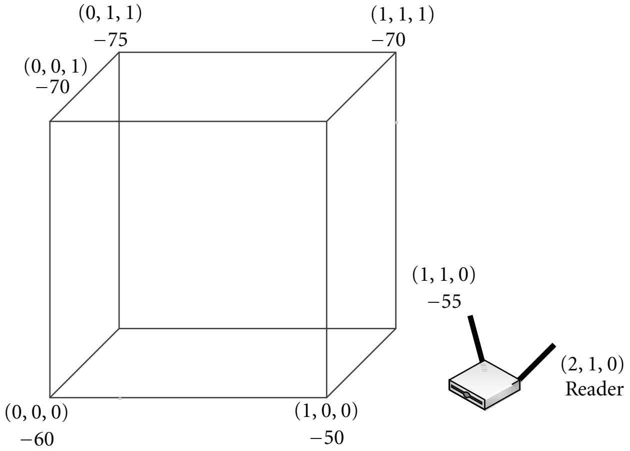


Figure 2: Reference tags' RSSI and predefined coordinates.

3.3. Virtual Reference Tag. We adopt the basic algorithm of VIRE [3] to calculate the virtual tags' RSSI and location to find out tracking tags' location. The VIRE algorithm works as follows.

It can be observed that if other environment factors keep unchanged, the active tag placed in the same position will present similar RSSI [7]. Therefore, one tag's RSSI can be estimated if its position is sufficiently close to a tag whose position is already known. To leverage this phenomenon,

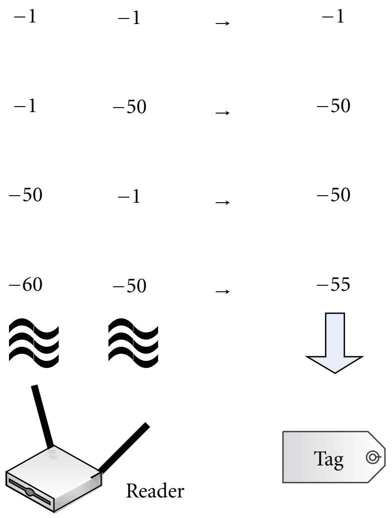


Figure 3: Determining RSSI for each tag.

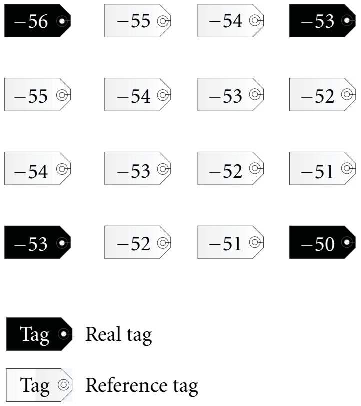


Figure 4: Interpolation of virtual reference tags' RSSI.

LANDMARC and VIRE use the reference tags to track other tags in the deploying area.

In common scenarios, the deployment of reference tags is sparse due to the cost consideration. In order to increase the accuracy, VIRE introduces virtual reference tags without increasing the number of real readers or tags, as shown in Figure 4. The RSSI values and coordinates of virtual reference tags are then calculated based on the real tags. In our approach, we also introduce virtual reference tags. We adopt the major principle of VIRE and then extend it from 2D to 3D, as illustrated in Figure 5.

The determination method of reference tags' RSSI is similar to VIRE, which is calculated by using the linear interpolation algorithm. According to VIRE, $n-1$ virtual reference tags should be equally placed between two adjacent real tags. After performing the linear interpolation algorithm, the RSSIs of virtual tags at both the vertical and horizontal directions can be computed. As shown in Figure 4, if we insert $n-2$ tags between two adjacent real tags, a $n \times n$ tag array can be obtained. To interpolate the RSSI of virtual tags in the 3D deployment, we first perform the VIRE algorithm on the upper and lower planes while treating them as two separate 2D cases. With the results on these two planes, we can easily extend the process to 3D. Finally, we can generate a 3D RSSI distribution of all tags in the deploying area.

3.4. Tracking Targeted Tags. After the 3D virtual reference distribution is retrieved, the RSSI value of each virtual reference tag can be used to track targeted tags. We can measure the RSSI value of the targeted tag via each reader. We then seek the matching between the tracked tag and some reference tags. In our approach, we select the reference tag(s) by using a threshold. The threshold is defined as the maximum acceptable difference between the targeted tags and reference tags. Thus, the positions of those reference tags with a difference smaller than the threshold from the targeted tag are potential locations of the tracked tags. In practice, many positions may be selected as the potential locations given an RSSI value of targeted tag. We employ the proximity map [7] to reflect the possible locations of tracked tags, while our approach extends the map to a 3D grid. Once the reference tags that are located at the potential locations of targeted tag are selected, the reader marks those regions as "1" in its proximity map. In practice, each reader will maintain such a map.

Intuitively, we integrated the proximity maps from the readers deployed in the detecting area. An intersection function is performed and then we filter the most probable regions of target from all readers. The so-called elimination will remove the unlikely positions. Here, determining a proper threshold is very important. The detail setting of threshold and the algorithm of elimination can refer to VIRE [7].

3.5. Refining the Location of Target. However, simply intersecting the proximity maps may be also inaccurate in terms of the scope of potential regions outlined by VIRE. We further improve the localization accuracy by "shrinking" the regions of potential positions. We employ two methods to achieve the refinement. First, we convert the raw 3D proximity maps into a 3D density map. Second, we compute the geometric mean of potential location and its surrounding locations in the 3D density map.

We employ a 3D array to contain the 3D proximity map. It included both the reference tags' locations and virtual tags locations. The array first stores "True" or "False" of the location to log whether the tracked tag is present. In most cases, the "True" location is not unique, as illustrated in Figure 6(a). In this example, we show a $3 \times 3 \times 3$ 3D reference tag array.

The 3D density map can be derived from 3D proximity map. The 3D density map stores the number of "True" locations surrounding one of locations in the 3D proximity map. Note that a location has 26 adjacent locations if this location is fully surrounded by its neighboring locations. For example, in Figure 1, the location of A where the tag is in black color has 26 adjacent locations and its reference tags are in gray color. Our objective is to find out the most densely region of "True" locations. For a given location, if one of

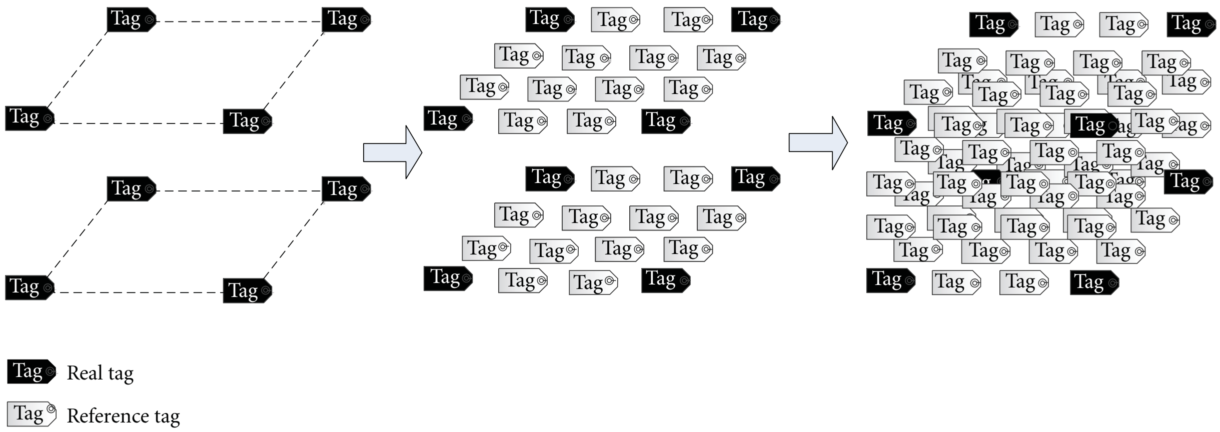
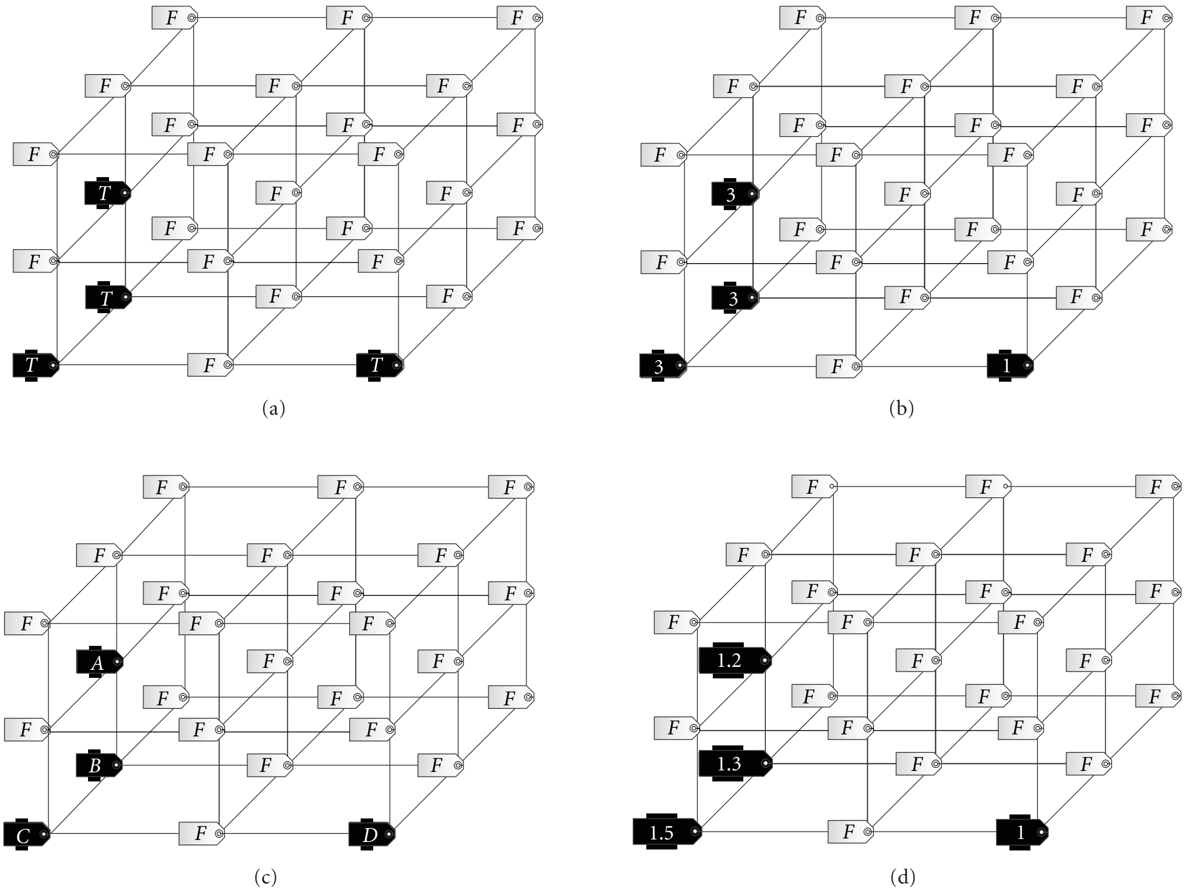


Figure 5: 3D virtual reference tag generation.



(a)

(b)

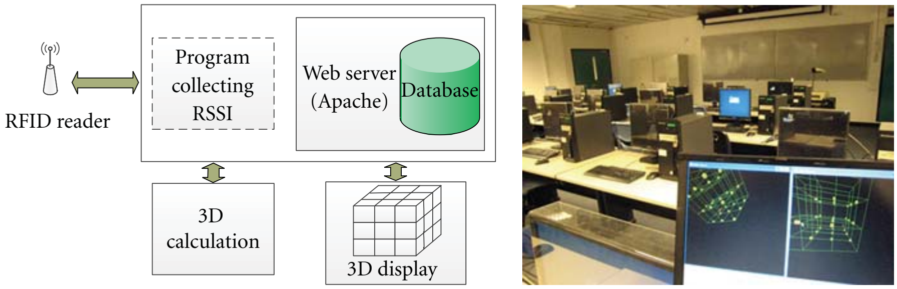
(c)

(d)

Figure 6: One of levels of a 3 × 3 × 3 3D proximity map.

its adjacent locations in the 3D proximity map is "False," no value will be added to this location. Otherwise, if the value is "True," its value will be added by 1. For the example in Figure 6, Figure 6(b) is the 3D density map converted from the 3D proximity map of Figure 6(a). In Figure 6(c), we represent the locations with "True" values in Figure 6(a) as location *A*, *B*, *C*, and *D*, respectively. After the counting process, the values of locations become 3, 3, 3, and 1.

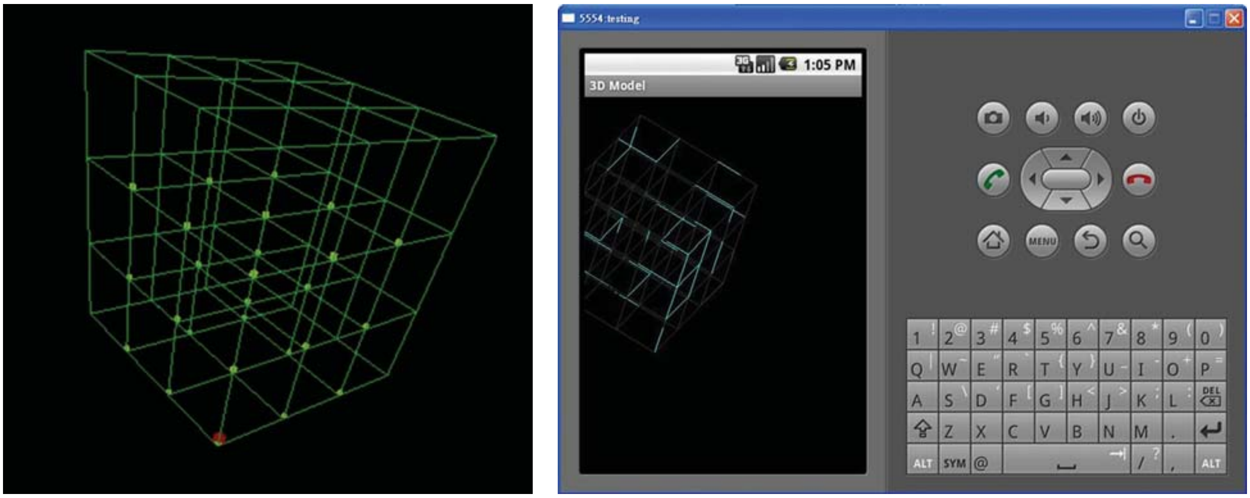
In case that the setting of threshold value is large, the dense region may not be unique in the 3D density map. In order to reduce the region and achieve more accurate location, we conduct a condensation process in the 3D



(a)

(b)

Figure 7: The components of our system and implementation in the indoor environment.



(a)

(b)

Figure 8: 3D effect in GUI and the simulated operating interface in mobile phone.

density map. The principle of this process is to drive the edge locations which have smaller values of density to be closer to the higher density part. We calculate the geometric mean among the given location and its adjacent locations. This process will be continued if the dense region of locations is not unique, until a unique highest density location occurs.

For example, in Figure 6(d), the calculation of four black tags' locations is as follows:

$$\begin{aligned} A &= \sqrt[18]{3^3} \approx 1.2, \qquad B = \sqrt[12]{3^3} \approx 1.3, \\ c &= \sqrt[8]{3^3} \approx 1.5, \qquad D = \sqrt[8]{1} = 1. \end{aligned} \tag{1}$$

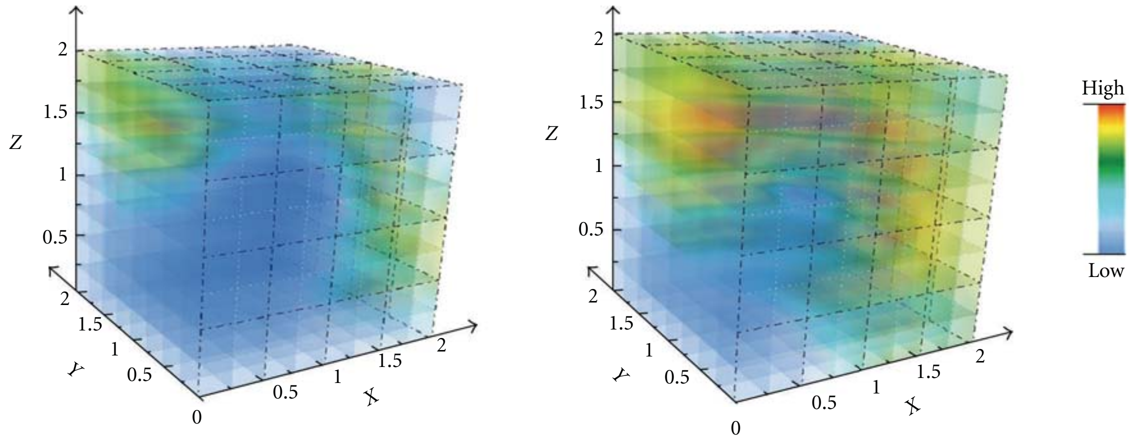
4. Implementation

We deploy the tags and readers in an indoor environment. The distance between two adjacent tags is one meter. The readers are placed in the corners of the sensing area. The location of reference tags and readers are represented in a 3D coordinate system. The beacon interval is 2 seconds.

We develop a 3D model program to display the location of tags on a graphical interface. This GUI can provide the user with both the overlook and detailed information about targeted tags' locations. We show the system deployment in Figure 7 and the prototype GUI of mobile phone in Figure 8.

We use Java to construct the 3D display model. It can be displayed on the mobile phone which supports the Java application. Our 3D model consists of an animation of 30 frames per second and it will continue to update the tag's coordinates of its location.

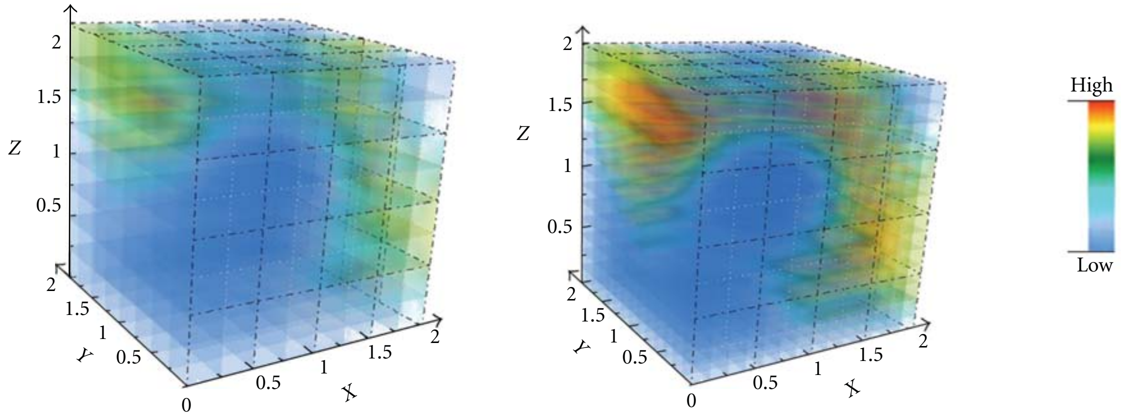
In the experiment, we mainly focus on examining the accuracy of the tracking tags. To simulate the practical application scenarios, we placed our setting in an empty classroom. Due to the limitation of the small size of the classroom, the setting was scaled down. The reference tags were placed in a 3×3×3 3D grid. Based on this grid, we set up a 3-dimensional coordinate system in the deploying room. The reference tags are placed on the floor, chairs, and tables for different levels. The values of z-coordinate of tags placed on the floor, chair, and table are 0, 1, and 2, respectively. The readers are placed at a coordinate of (0, 0, 2) and (1, 0, 2). We introduce 3 targeted tags placed at the position of (0.5, 1.5, 2). We connected 2 readers with LAN.



Threshold: 7
Virtual tag number: 4

Threshold: 10
Virtual tag number: 4

Figure 9: Performance under different threshold values.



Threshold: 7
Virtual tag number: 4

Threshold: 7
Virtual tag number: 7

Figure 10: Performance with different numbers of virtual tags.

5. Evaluations

We examine the performance of localization with two matrixes. We first evaluate the impact of threshold settings. We then study the performance with different numbers of tags inserted into the 3D grid. We also check the diversity effect of tags. We finally investigate the localization errors of our approach.

5.1. Threshold Value. Figure 9 shows the effects on the density map with different threshold values 7 and 10. We can observe that the location of (0.5, 1.5, 2) is with the highest density value and with a very small region as the reported target location. In the contrast, the distribution of possible locations of target tag becomes much larger if the threshold is set as10 than the setting of 7. This indicates that the higher the threshold value is, the higher the error will be. Therefore, it is appropriate to set a small threshold value. In our experiments, the RSSI value changes around 5 units. We then set the threshold value as 5~7. In practice, we suggest that the optimal threshold value can be set according to user specification or the sensitivity of RFID devices.

5.2. Different Virtual Tag Numbers. We then examine the impact of the number of virtual tags and show the result in Figure 10, where the virtual tag numbers are 4 and 7 between two real tags. We can find that the resolution of target location becomes higher when changing the setting from 4 to 7. It can be concluded that setting a larger number of virtual tags tenders more accurate location report. As the number of virtual tags increases, the resulting location will be more precise. However, introducing more virtual tags would cause much more computational workload. In our experiments, we observe that setting the number of virtual tags from 4 to 7 yields acceptable workload to the server while keeping satisfied localization accuracy.

5.3. Comparison on Different Tracked Tags. We also check the performance change when tracking different tags. Figure 11

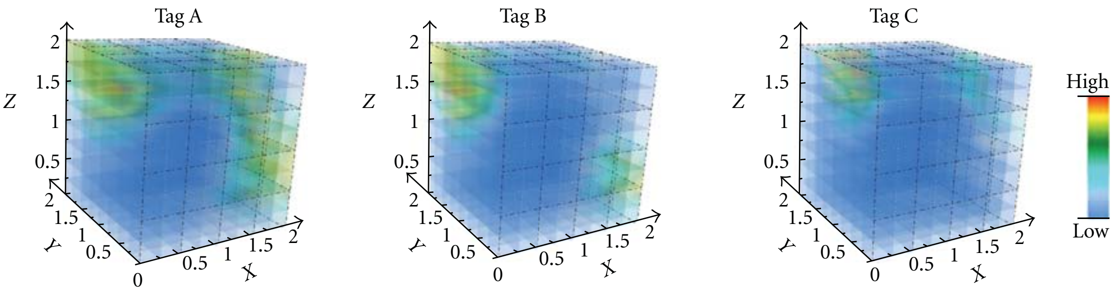


Figure 11: Comparison on different tracking tags.

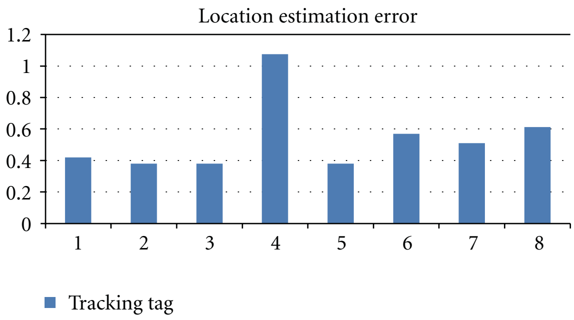


Figure 12: Estimation error.

plots the result of tracking three tags. We adopt the setting of 7 as the threshold and 4 as the number of virtual tags. We randomly select three tags, A, B, and C. We then place them in the same position. We observe that their density maps are similar. The reported possible location regions of three tags are almost same, while the calculated locations of three tags are very close to (0.5, 1.5, 2).

5.4. Location Estimation Error. The location estimation error is the distance between the real position of tracked tag and the computed position. Suppose the former is (x_0, y_0, z_0) and the latter is (x, y, z). To compute the estimation error e, we have the following formula:

$$e = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}. \qquad (2)$$

In order to compute the location estimation errors, we still use the same testbed as above, but the locations of the tracked tags are at the center of the 4 reference tags. We show their coordinates in Table 1.

Table 1: Coordinates of tracked tag.

Tracking tag	x	y	z
1	0.5	0.5	0.5
2	0.5	0.5	1.5
3	1.5	0.5	0.5
4	1.5	0.5	1.5
5	0.5	1.5	0.5
6	0.5	1.5	1.5
7	1.5	1.5	0.5
8	1.5	1.5	1.5

From the formula, the e of 8 tags is shown in Figure 12. The readers are placed at two positions (0, 0, 2) and (1, 0, 2). For Tag-1, Tag-2, and Tag-3 that are placed close to the readers, their estimation errors are around 0.4 m. For Tag-6, Tag-7, and Tag-8, which are placed far away from the readers, the estimation errors are about 0.55 m. In particular, the interference at Tag-4 is so strong that the error approaches 1.1 m at that position. From the result, we learn that the estimation error is higher when the distance between the readers and the tracking tag becomes larger. In addition, the average estimation error of 8 tags is 0.54 m. In the 3D grid, each reference tag is assumed to be placed with a 1-meter distance with its adjacent tags. To summarize, the accuracy of the system is acceptable for the indoor application.

6. Conclusion

In this paper, we propose a 3D RFID localization technology. The work extends the RFID-based localization from 2D to 3D space and refines the location regions from pervious works. The experiment result shows the feasibility and effectiveness of our work. Our future work will focus on the implementation of proposed systems in more real RFID applications.

Acknowledgments

This work is partially supported by the National Key Basic Research and Development Program (973) of China under Project no. 2011CB302705, NSFC under Project no. 61033015 and no. 60903155, and the Fundamental Research Funds for the Central Universities of China.

References

[1] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1441–1454, 2011.

[2] M. N. Lionel, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[3] Y. Liu, L. Chen, J. Pei, Q. Chen, and Y. Zhao, "Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, pp. 37–46, March 2007.

[4] Y. Zheng, M. Li, and C. Qian, "PET: probabilistic estimating tree for large-scale RFID estimation," in *Proceedings of the International Conference on Distributed Computing Systems*, pp. 37–46, Minneapolis, Minn, USA, 2011.

[5] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, Article ID 5280252, pp. 320–332, 2010.

[6] Z. Yang and Y. Liu, "Quality of trilateration: confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, Article ID 5066966, pp. 631–640, 2010.

[7] Y. Zhao, Y. Liu, and L. M. Ni, "VIRE: active RFID-based localization using virtual reference elimination," in *Proceedings of the 36th International Conference on Parallel Processing in Xi'an (ICPP '07)*, September 2007.

[8] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *Journal of Computer Science and Technology*, vol. 25, no. 2, pp. 274–297, 2010.

[9] M. Li, W. Cheng, K. Liu, Y. Liu, X. Li, and X. Liao, "Sweep coverage with mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1534–1545, 2011.

[10] X. Wu, S. Tan, T. Chen, X. Yi, and D. Dai, "Distributed dynamic navigation for sensor networks," *Tsinghua Science and Technology*, vol. 16, no. 6, pp. 648–656, 2011.

[11] L. Wang and W. Liu, "Navigability and reachability index for emergency navigation systems using wireless sensor networks," *Tsinghua Science and Technology*, vol. 16, no. 6, pp. 657–668, 2011.

[12] B. Sheng, C. C. Tan, Q. Li, and W. Mao, "Finding popular categories for RFID tags," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 159–168, Hong Kong, May 2008.

[13] T. Kriplean, E. Welbourne, N. Khoussainova et al., "Physical access control for captured RFID data," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 48–55, 2007.

[14] C. Tan, B. Sheng, and Q. Li, "Efficient techniques for monitoring missing RFID tags," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, Article ID 5475333, pp. 1882–1889, 2010.

[15] D. Zhang, J. Zhou, M. Guo, J. Cao, and T. Li, "TASA: tag-free activity sensing using RFID tag arrays," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 558–570, 2011.

[16] L. G. Roberts, "ALOHA packet system with and without slots and capture," *SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[17] S. R. Lee, S. D. Joo, and C. W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 166–172, July 2005.

[18] EPCGlobal, "EPCGlobal radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz," Technical Report, 2005.

[19] ISO, "Information technology—radio frequency identification for item management—Part 6: parameters for air interface communications at 860 MHz to 960 MHz," Technical Report ISO Standard No. ISO/IEC 18000-6, 2004.

[20] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proceedings of the IEEE International Symposium on Information Theory*, 1998.

[21] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, 1979.

[22] J. Myung and W. Lee, "Adaptive binary splitting: a RFID tag collision arbitration protocol for tag identification," *Mobile Networks and Applications*, vol. 11, no. 5, pp. 711–722, 2006.

[23] J. Ho, D. W. Engels, and S. E. Sarma, "HiQ: a hierarchical Q-learning algorithm to solve the reader collision problem," in *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINT '06)*, pp. 88–91, January 2006.

[24] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, "Slotted scheduled tag access in multi-reader RFID systems," in *Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP '07)*, pp. 61–70, Beijing, China, October 2007.

[25] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '06)*, pp. 269–274, Pisa, Italy, March 2006.

[26] W. Choi and B.-H. Roh, "Backward channel protection method for RFID security schemes based on tree-walking algorithms," in *Proceedings of International Conference on Computational Science and Its Applications*, pp. 279–287, 2006.

[27] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," *Security in Pervasive Computing*, vol. 2802, pp. 201–212, 2004.

[28] T. L. Lim, T. Li, and S. L. Yeo, "Randomized bit encoding for stronger backward channel protection in RFID systems," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08)*, pp. 40–49, March 2008.

[29] L. Lu, Y. Liu, and X. Y. Li, "Refresh: weak privacy model for RFID systems," in *Proceedings of the 29th Conference on Information Communications (INFOCOM '10)*, March 2010.

[30] RF Code, 2011, http://www.rfcode.com/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 758789, 15 pages
doi:10.1155/2012/758789

Research Article

InContexto: Multisensor Architecture to Obtain People Context from Smartphones

Gonzalo Blázquez Gil, Antonio Berlanga, and José M. Molina

Group of Applied Artificial Intelligence, University Carlos III of Madrid, 28270 Colmenarejo, Spain

Correspondence should be addressed to Gonzalo Blázquez Gil, gbgil@inf.uc3m.es

Received 15 July 2011; Revised 29 December 2011; Accepted 3 January 2012

Academic Editor: Mo Li

The way users intectact with smartphones is changing after the improvements made in their embedded sensors. Increasingly, these devices are being employed as tools to observe individuals habits. Smartphones provide a great set of embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone, and camera. This paper aims to describe a distributed architecture, called inContexto, to recognize user context information using mobile phones. Moreover, it aims to infer physical actions performed by users such as walking, running, and still. Sensory data is collected by HTC magic application made in Android OS, and it was tested achieving about 97% of accuracy classifying five different actions (still, walking and running).

1. Introduction

Traditionally, Internet has been accessed from a desktop computer. However, nowadays Internet access is also extended to the mobile phone or commonly called smartphone. The penetration rate of these devices is growing rapidly. For example, in the USA, 27% of mobile phone users had a smartphone at the end of 2010 in some countries of Europe (France, Germany, Italy, Spain, and the UK), smartphone penetration was even larger, reaching 31.1% (comScore 2011 whitepaper, http://www.comscore.com/). By 2011, smartphones sales are projected to overcome desktop computer. Hence, smartphone is becoming increasingly popular as a personal computer, and becoming the main computer and communication device in people's lives.

Indeed, nowadays, smartphones do not only provide internet access; besides, they are provided by a countless number of sensors. Microphones and digital cameras are the most common ones; however, they are being equipped with new sensors: accelerometer, gyroscope, compass, magnetometer, proximity sensor, light sensor, GPS, and so forth [1]. Taking advantage of these features, developers have created new amazing app in order to improve user smartphone experience [2]. The embedded sensors allow the device to adapt to environment conditions, use of battery, lighting conditions, and sound. For example, light sensor controls screen brightness in order to preserve battery life. When the user is using the smartphone in a dark place, the screen brightness is reduced.

Moreover, thanks to smartphone mobile connection over different radio channels it is possible to consider them as a new sensor inside Ambient Intelligence (AmI) Environments. Smartphone ability to act based on sensory information extends user concept. Now the user is provided by a new set of sensory abilities. Smartphones are characterized by multiple sensors retrieving scenario context information in order to recognize inconspicuous activity of individuals and react to their needs.

First of all, in order to determine user needs, it is necessary to know their status and the context where it is located. User status is considered a combination of physical activity and emotional state. Their needs are different if a person runs doing a sport (probably he/she needs to complement his/her activity with music) or he/she runs to avoid a dangerous situation where the essential need is to track your position and advise the emergency services.

Activity recognition aims to perceive which activity is taking place. In these applications, high classification accuracy is always desired. Daily, human beings make ordinary actions such us a cooking, reading or watching TV, chatting with other people or on the phone, and driving [3]. The ability of activity recognition seems so natural and simple for us; however, actually it requires complicated functions of sensing, learning, and inference for computers [4].

Traditionally, activity recognition is carried out through video systems like those described in [5, 6]. However, recent researches in activity recognition show that microelectro mechanical systems (MEMSs) are becoming another way to face this problem [7]. They can return a real-time measurement of acceleration along the x-, y-, or z-axis to be used as a human motion detector.

In general, placing more accelerometers on different body positions improves pattern recognition performance [8]. At the same time, a wearable system must be inconspicuous and operate during long periods of time [9]. However, people are reluctant to wear strange devices over the body. In this case, smartphones are especially well-suited to accomplish this task since they have integrated MEMS and people consider them as friendly devices. Smartphones may obtain and process physical phenomena from embedded sensors (MEMS) and send this information to remote locations without any human intervention [10].

For that reason, it may be possible to consider a smartphone like a nonintrusive device to obtain activity Context from people [10]. Indeed, smartphones experience almost the same physical forces, temperature, and noise of the person who carries them out. If you track their actions, you are tracking people actions.

Although smartphones are considered as a single device, they provide several sources of information, mainly MEMS, internet connection, and human interaction to gather all this information in order to reach better results in activity recognition problem. Information fusion techniques [11] aim to *combine observations from a number of different sensors to provide a robust and complete description of an environment or process of interest*.

However, to handle all the information from the different sensors is pretty costly. In an extreme case, each sensor may have its own processor to manage the local data and cooperate with other sensor nodes. Traditionally, activity recognition system usually employs hard sensor (MEMS) nevertheless, there are other user information sources available in the smartphones. Users daily share their personal information on social networks sites, Facebook, Linkin, Twitter, and so on. These type of sensors are called soft sensors in information fusion researches which are referreded as human observer that provides his/her point of view of something.

Information fusion techniques have been proved in several and complex scenarios [12], but they have not been used in smartphone devices. The principal achievements of information fusion systems are robustness, increased confidence, reduced ambiguity and uncertainty, and improved resolution. For that reason, taking advantages of information fusion techniques, an smartphone architecture has been deployed in order to collect user data and infer user context from smartphones.

In the literature, there are mainly two different ways to obtain user activity using MEMS. Classical techniques just take into account ad hoc accelerometers sensors, for example, in [8] Bao and Intille present a multisensor system wearing six accelerometers around the body, which reaches about 80% of accuracy with different actions. In other research, Barralon et al. [13] describe an activity recognition system for eHealth applications where every patient wears a single sensor on the chest. The final results show that the systems able to differentiate among walk or no walk over 80% of times. Although these systems reach good accuracy, in practice, they are quite uncomfortable, and also they are considered intrusive by the users. On the other hand, recent researches use smartphones to accomplish activity recognition problem. One of the most famous works is Miluzzo et al.'s [10]. The stronghold of CenceMe system is that it sees SNS as a site where you can share user activity information instead of a sensor where you can obtain user information.

Summarizing, this paper is focused on the description of inContexto, an information fusion architecture which retrieves smartphone context information as well as the user who carries it. Besides, inContexto architecture lays the guidelines to collect user information from every provided sensor in the smartphone, whether it is a hard sensor or soft sensor. Finally, inContexto activity recognition module was tested obtaining an overall performance over 97% of accuracy classifying still, walking, running, riding a bike and lying user actions. Besides, a public dataset has been publish with the activity recognition data.

The paper is ordered as follows: Section 2 depicts the actual state-of-art of activity recognition using smartphones and information fusion techniques. Section 3 aims to describe the different user information sources using smartphones. Section 4 presents the proposal architecture according to the context sources, and preliminary results from an initial deployment indicate the potential for accurate, context-aware, and personalized sensing. Results of the chosen activity recognition techniques are shown in Section 5, and finally, Section 6 shows the conclusions and future work.

2. Related Works

Regarding the fields of sensor fusion and activity recognition separately, both are well treated in the scientific literature. In this section, firstly, we will focus on research works that use smartphones to retrieve user activity context and subsequently information fusion architecture is described in order to implement one in our work.

2.1. Information Fusion Architectures. Multisensor fusion architectures are not common in smartphone applications. Nevertheless, there are just a few researches [2] using this information fusion techniques. Ganti et al. presented an architecture for lifestyle monitoring, but it just collects data from sensors in the smartphone, and subsequently, information is sent to a computer desktop for data analysis.

In our case, information fusion is necessary to integrate the data from the different sensors (hard and soft sensors) in

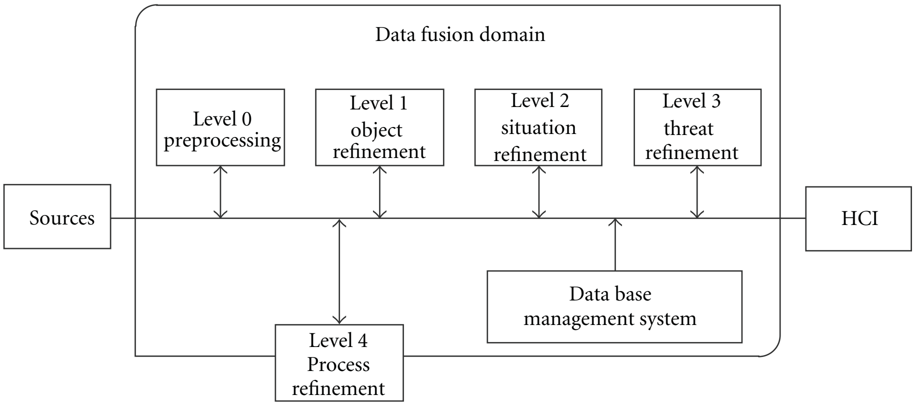

Data fusion domain
Level 0 preprocessing
Level 1 object refinement
Level 2 situation refinement
Level 3 threat refinement
Sources
HCI
Data base management system
Level 4 Process refinement

Figure 1: JDL information fusion model.

order to extract the relevant information on the users. Normally, data fusion architectures are based on an centralized system; however, this algorithm presents high computational cost increasing energy consumption. Thus, in order to prevent this problem, a distributed architecture is designed sharing computational process between the smartphone and cloud servers.

Below, most common general information architectures are described in order to consider pros and cons to use them in a mobile device.

2.1.1. JDL Architecture. Historically, data fusion methods were developed basically for military applications. The military community has developed a layout of functional architectures based on the joint directors of laboratories model for multisensory systems. In recent years, these methods have been applied to civilian applications [14] but never in mobile device.

The JDL model was never intended to decide a concrete order on the data fusion levels. Levels are not alluded to be processed consecutively, and it can also be executed concurrently. Figure 1 depicts JDL data fusion process high level model.

(i) *Level 0.* subobject data assessment is associated with predetection activities such as pixel or signal processing, spatial or temporal registration.

(ii) *Level 1.* At this level, to identify and locate objects is attempted. Hence, the object situation by fusing the attributes from diverse sources is reported. The steps included at this stage are:

(a) alignment: processing of sensor measurement to achieve common time base and a common spatial reference,

(b) association: a process by which the closeness of sensor measurement is completed,

(c) correlation: a decision-making process which employs an association technique as a basis for allocation sensor measurement to the fixed or tracked location of an entity,

(d) correlator-tracker: a process which generally employs both correlation and fusion component processes to transform sensor measurements into states and covariance for entity track,

(e) classification: a process by which some level of identity an entity is established either as a member of a class, a type within a class, or a specific unit within a type.

(iii) *Level 2.* Attempts to construct a picture from incomplete information provided by level 1, that is, to relate the reconstructed entity with an observed event. Entities are associated with environmental, doctrinal, and performance data.

(iv) *Level 3.* It interprets the results from level 2 in terms of the possible opportunities for operation. It analysed pros and cons of taking one action over another one.

(v) *Level 4.* Process refinement is an element of resource management and used to close the loop by retasking resources (e.g., sensors, communications, and processing) in order to support the objectives.

Taking into account that JDL model is considered an abstract model, it is not a guideline to implement information fusion architecture. However, it makes easier to distribute which components should run on the cloud or in the mobile phone.

2.1.2. Waterfall Fusion Architecture. The waterfall IF model was proposed by Markin et al. [15] (see Figure 2). This architecture emphasizes the processing functions on the lower levels. However, waterfall model omits any feedback data flow instead of JDL model in which every level is interconnected. The relationship between waterfall architecture and JDL model is as follows.

(i) Sensing and signal processing correspond to level 0.

(ii) Feature extraction and pattern processing match with level 1.

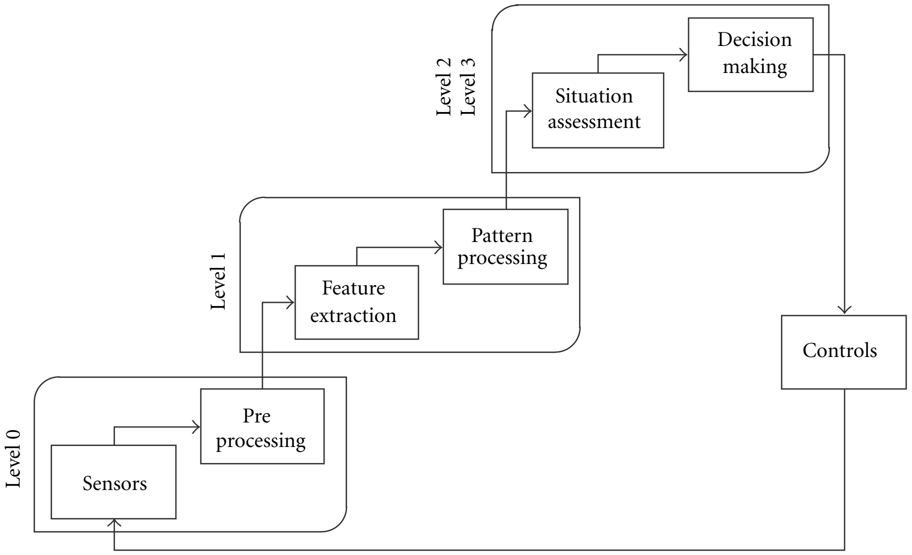


Figure 2: Waterfall information fusion model.

(iii) Situation assessment is similar to situation refinement in JDL model, level 2.

(iv) And finally, decision making corresponds to the third JDL level 3.

Although, waterfall model is more accurate in analysing the fusion process than other information fusion models, it presents some drawbacks, for example, the omission of any feedback data flow.

Taking into account pros and cons of both architectures, inContexto has been designed relying on JDL model. Its modularity gives us the advantages to divide some component on the smartphones and others on the cloud. Hence, it is able to operate in distributed systems.

2.2. Mobile Phone Activity Recognition Architectures. Normally, in the literature, there are two kinds of researches to obtain activity using mobile devices. The first one has been focused on ad hoc solution, and the second one and more recent is using smartphones solutions. Each activity recognition architectures are briefly described below with information on how inContexto builds on or differs from the ones described.

2.2.1. Ad Hoc Activity Recognition Architectures. Barralon et al. [13] work describes an MEMS architecture and shows the results of the time spent in three postural states (lying, sitting, standing) and the periods of walking in an eHealth scenario using a unique accelerometer, placed on the patients chest.

The study determines the global position of the patients of the sensor wearer, and they calculate the position of the patient considering the inclination of the sensor in every axis and then quantify this value. Finally, the study was made to evaluate the health of the patient, and they obtain about 76% of accuracy rate.

On the other hand, Bao and Intille describe an architecture [8] to acquired human motion using five biaxial accelerometers worn on different parts of the body from 20 subjects. Extracted features from each accelerometer were: signal mean, energy, frequency-domain entropy, and correlation of acceleration and subsequently classify using a decision tree, obtaining an overall accuracy rate of 84%. Although they reach a good accuracy, this architecture presents, a big problem, to wear five devices over the body.

2.2.2. Smartphone Activity Recognition Architectures. One of the most notable contributions presented up to now in mobile phone activity recognition is called CenceMe [16]. A mobile sensing architecture to obtainand share user physical activities on a social network.

Although CenceMe does not use social networks sites to collect information (they only use accelerometer), they introduce SNS into the activity recognition field sharing user activity on Facebook. The proposed architecture is split in three layers (sense, learn, and share):

(i) sense layer aims to collect raw sensor data from sensors embedded in the phone in the Apple iPhone in order to track body movements,

(ii) in learn layer, they propose to use a variety of data mining techniques to infer user rules. These techniques are used to interpret three-axis accelerometer raw data extracted in the sensor layer,

(iii) their approach aims to share activity information in a web portal where sensor data and inferences are easily displayed.

Chon and Cha [17] present LifeMap architecture, a Smartphone-based context provider for location-based services. Authors split their architecture in four components:

(i) all the sensors are placed on the low level; this level sends the obtained information to the component manager where information is processed and provide, high-level information, using high-level information from the component manager.

(ii) the context generator generates a point of interest (POI) which contains the user context. The context map is stored in a database to match and aggregate user contexts,

(iii) and finally, the database adapter is an interface to provide user context to other applications.

Our work differs from existing solutions in that it does not rely on external mobile devices nor the accelerometer position when the user wears it. In contrast, using a smartphone as a nonintrusive device permits to obtain user movements with embedded sensors. On the other hand, GPS only solutions work well for classification of activities with different speed; however, it is necessary for another sensor to distinguish between similar speed activities such as riding a bike or running. Accelerometer-based technique presents best results in that way. Finally, the most significant difference between our work and existing works is that we describe an architecture to handle information from different information sources (Accelerometer, Gyroscope, GPS, SNS, etc.) using information fusion techniques. Although one sensor was offline, it is possible to generate user information handling the other sensors.

Table 1 shows a summary of works that have taken place in this space along with the types of activity modes inferred, the test user base, and the classification accuracy.

Table 1

Research	Classes	Sensors	Mobile
CenceMe [10]	Still, walk, run	Accelerometer	Yes
lifeMap [17]	Still, walk, motor	Accelerometer, magnetometer, wifi, and GPS	Yes
Borriello [3]	Still, walk, stairs up and down, riding elevator, and brushing	GSM, Wifi	No

3. Describing Smartphone Context

First of all, in order to use context correctly, it is crucial to define what researchers think context is. In general, context aware is represented by applications which change their behaviour according to the conditions around them, in this case the smartphone conditions. Applications and services react specifically to their surroundings, location, and time. Summarizing, their behavior is able to change according to circumstances.

In 1994 was introduced the term context-aware computing by Schilit and Theimer [18]. *They defined context as a software which adapts according to its location of use, the collection of nearby people and objects, and changes to those objects over time.* Subsequently, some other researchers try to formally define context, for example, Schmidt et al. [19] define context as knowledge about the user's and IT device's state, including surroundings, situation, and location. it one of the most accurate definition given by Dey and Abowd [20]. they defined context as: *any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.*

Hence, everything in the world may be considered as an entity, for example, a bedroom has its own context, the people who is lying in, number of furniture, and so forth. Dey and Abowd defined the three kinds of entities:

(i) places: it represents a point or an area, for example, buildings, rooms, village, and so forth,

(ii) person: an individual or groups of people,

(iii) objects: electronic devices, physical objects, and so forth.

Each entity is characterized by four categories (Identity, location, status or activity, and time). According to Dey and Abowd's definition about context and entity, this work presents smartphone entity representation (see Figure 3).

(1) *Identity.* In order to identify one person is possibly to use different sources, hardware or software, hardware identification, as MAC address, presents several problems because you are identifying the smartphone instead of the person who carries it. Hence, if another user manipulates the same device, there will be identification problems. However, using software identification as Facebook platform (FP), this problems would be solved.

(2) *Location.* Location aware could be the main factor in the development of context applications. Nevertheless, location aware is only one aspect of context aware as a whole [21]. Location context may be described as an application dependent on the geographical location. Location answers the questions where is the action taking place? For example, it is possible to define a running action; however, it could be interesting to define where is he/she running? and where is he/she running to? To obtain location using mobile phones is really simple however, in outdoor environments; GPS provides a good solution to determine the location of mobile devices; however, in GPS-denied areas such as urban, indoor, and subterranean environments, unfortunately, an effective solution does not exist. Besides, every location system provides in its own way location data. Recently, W3C has reloaded a Geolocation API [22] to standardize an interface to get back the geographical location information for a client device.

(3) *Status or Activity.* Talking about status it is necessary to differentiate user status, and mobile phone status. smartphone status mainly refers to communication behaviour: calls and calls attempts, sent and received SMS, SMS content, battery level, wireless connections, and so forth. On the contrary, user status does not refer just to her/his calendar (working, sleeping, free-time, etc.), otherwise the relevant

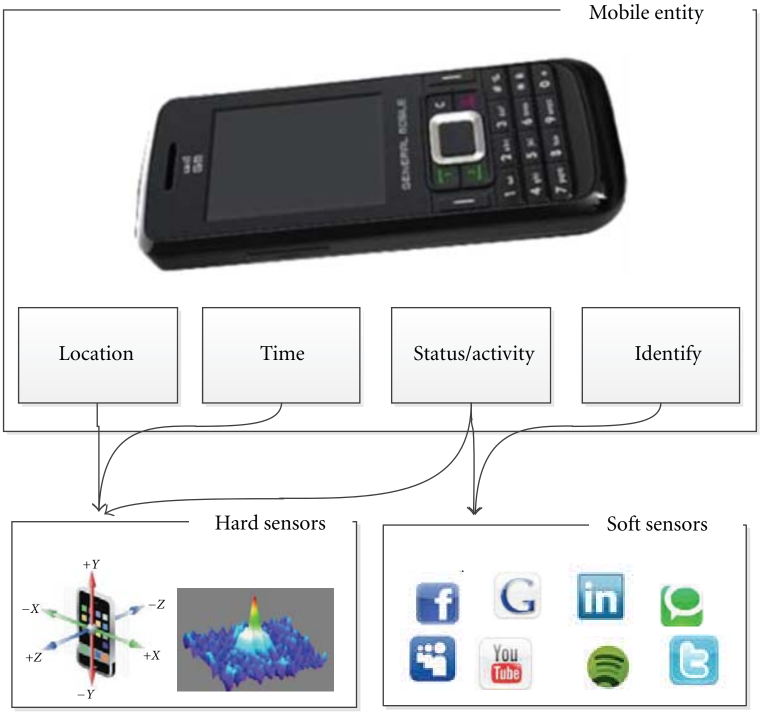


Figure 3: Entity representation using smartphones.

information about the user, normally, is included in the user profile as an instance (name, date of birth, where is she/he was born, etc.). As it was described previously, people movements are reflected in mobile devices sensors. The generated information can be used to identify different activities (e.g., running, walking, standing, cycling etc.) that the user is performing. These kinds of actions are obtained by low-level sensors provided by the mobile phone (accelerometer, Gyroscope, light sensor, microphone, etc.). For example, accelerometer is ably to describe the physical movements of the user carrying the phone.

(4) *Time.* activities taken by the user or the user's status do not have any meaning if it is impossible to set the action in a place and in time. For that reason time is an essential in context-aware applications.

3.1. Describing Sources of Mobile Context. According to entity representation (see Figure 3), this section aims to describe different sources of context in smartphones, and also it matches user actions with smartphones sensors.

3.1.1. Hard Sensor. The camera and microphone are probably the most used sensors in AmI systems. However, these sensors present several issues. In order to retrieve user information, it is necessary to process all the information and transform it from raw data to features.

Basically, using this kind of sensors, it is possible to obtain basic actions taken by the user such as running, walking, standing, talking, and listening music. These actions are obtained by low-level sensors provided by the mobile phone (accelerometer, gyroscope, light sensor, microphone, etc.).

(1) Accelerometer: A triaxial accelerometer is a sensor that returns a real-valued estimate of acceleration along the x-, y- and z-axes from its velocity. Accelerometers can be used as motion detectors as well as for body position and posture sensing [23]. Collected data from the accelerometer has the following attributes: time, acceleration along three axes (x, y, and z), not including gravity.

Accelerometer provides data from the origin of coordinates of the device which is placed in the lower-left corner with respect to the screen, with the X-axis horizontal and pointing right, the Y-axis vertical and pointing up, and the Z-axis pointing outside the front face of the screen. In this system, coordinates behind the screen have negative Z-values (Figure 4). Hence, if the mobile device is worn on a pocket, it is not clear which axis or axes represent the real world coordinates. In the next section, it is presented how to transform, using digital compass, coordinates from smartphone representation to real-world one which will be described.

Accelerometer sensor is well fit to be used to infer pedestrian movements due to acceleration data of walking or running displays distinct phases and periodicity of the signal; however, it is very difficult to differentiate transportation modes.

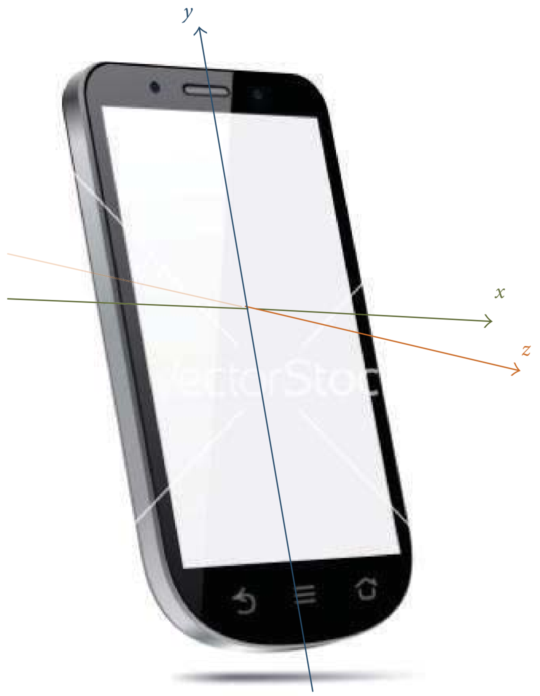


Figure 4: Smartphone Android OS coordinates origin.

(2) Digital compass provides two kinds of measures: the first one is the orientation whose values are in radians/second and measure the rate of rotation around the x (roll), y (pitch), and z (yaw or azimuth). Also, the coordinate system is the same as is used for the acceleration sensor.

Digital compass reports the angle between the magnetic north and the mobile phone's y-axis (orientation measurement). All values are in microTesla (uT), and it measures the ambient magnetic field in the x, y and z axes.

This sensors do not have a concrete value describing user actions, but it is usually used to determine user movements direction.

(3) Gyroscopes are the most commonly used sensors for measuring angular velocity and angular rotation in many navigation and homing applications. They measure how quickly an object rotates and, specifically, measure the rate of rotation around the X-, Y- and Z-axes. The coordinate system is exactly the same as is used for the acceleration sensor. Gyroscopes are the only inertial sensors that provide measurement of rotations without being affected by external forces, including magnetic or gravitational or fabrication imperfections.

(4) Location sensor: there are three ways to locate the smartphone, first of all using a GPS, in this case every smartphone provides an assisted GPS [24]. A-GPS improves the performance by adding information, through another data connection (Internet or other), more than unassisted GPS in order to receive and process signals as computationally costly, minimizing the amount of time and information is required from the satellites. The A-GPS receiver uses satellite to locate itself, but it can do more quickly and using weaker signals than an unassisted GPS. Normally, an A-GPS provides 2–4 meters error.

The second way to locate the smartphone is using GSM cell tower triangulation. This technique is reduced and more accurate than GPS; however, the energy consumption is reduced as well. According to the application goals, it is necessary to balance the accuracy and the energy consumption, and it could be enough a coarse location (GSM) instead of a precision location (GPS).

Finally, using Internet connection (Wifi) is possible to locate the smartphone thanks to W3C that has reloaded a Geolocation API to standardize an interface to get back the geographical location information for a client device (Geolocation API http://dev.w3.org/geo/api/spec-source.html).

3.1.2. Soft Sensor. Social networks sites (SNSs) are increasingly popular these days. In [25] is described social network site as: *Web-based services that allow individuals to (1) construct a public or semipublic profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site.*

Each SNS is implemented with specific features; however, all of them have a common point which consists of visible profiles. Daily, SNS users share their personal information, and SNS manage as uncountable gigabytes of useless user information. Why do not we use these data to obtain user context information?

Typically, user profiles include descriptors such as age, location, and interest schools attended. User profiles are becoming more precise: music preferences, movies, clothes, friendship relationships, personal agenda, and so forth.

3.2. Context Action Concept. Context action concept (see Figure 5) is the result of combining all the different contexts (identity, location, activity, and time context) according to Dey and Abowd's definition as well as each category which describes a concrete action. This paper distinguish two kind of activities: basic activities (e.g., walking, talking, running, cooking etc.) which cannot be decomposed into more simple actions and composite activities (context activities) which are composed by various simple actions (e.g., giving lectures, talking, standing, and relationships with other people).

For example, consider the following scenario, someone is sitting in her/his living-room watching TV. The accelerometer and the microphone may detect whether the user is sitting (Motion-Activity context) or the user is near a sound source (Sound-Activity context). If you use the both context and it is able to locate the action, Location context, (location is happening in the living room) it could figure out that the

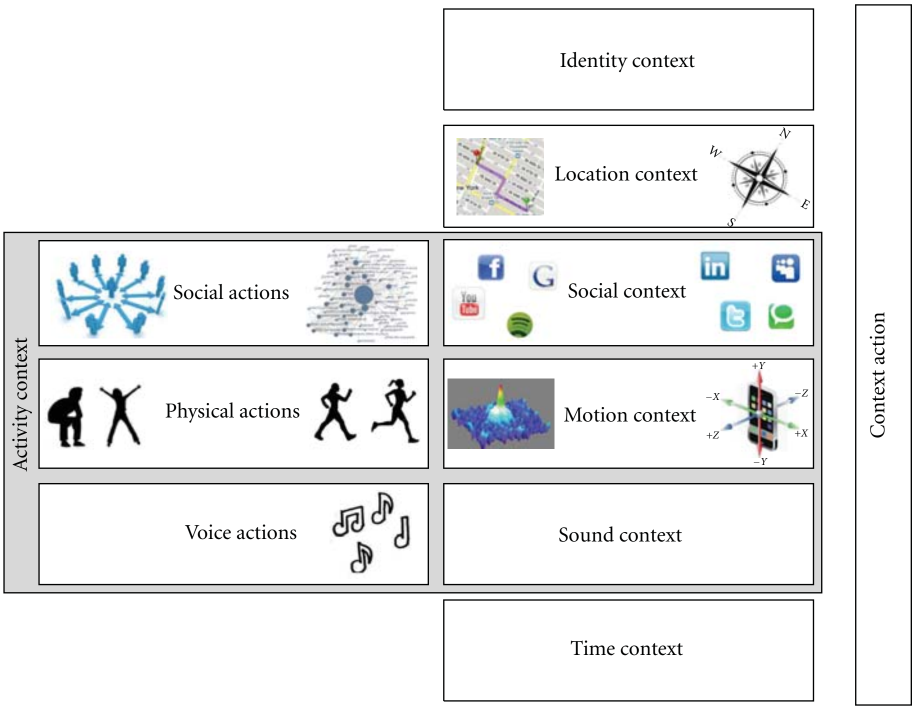


Figure 5: Context Action concept.

person is sitting in the living room watching TV (Context Action).

4. InContexto: Architecture Definition

In this section, inContexto is described (Figure 6) a multimodal architecture to obtain context from a user who carries out an smartphone.

It is based on the JDL model which proposes five different levels in order to transform input data into decision. These levels are called signal feature assessment (L0), entity assessment (L1), situation assessment (L2), impact assessment (L3), and process assessment (L4). Observational data may be combined from the raw data (or observation) level to a state vector level, or at the decision level.

Combining information fusion and activity recogonition techniques in a smartphone is not a vanal task due to energy restrictions and the computational cost of these techniques. Hence, it is important to highlight that, nowadays, it is not clear what architectural components should run on the device and what should run on the cloud. In this case, it is proposed that L0 and L1 are implemented in the smartphone; on the contrary, the other ones are executed in backend infrastructure.

InContexto is implemented following a distributed architecture where a communication component is designed to associate the smartphones with the backend server.

4.1. Data Collection Level 0. Data collection level aims to transform raw data (accelerometer, gyroscope, location, light sensor, and soft sensors) into processed data easy to manage by the features selection level.

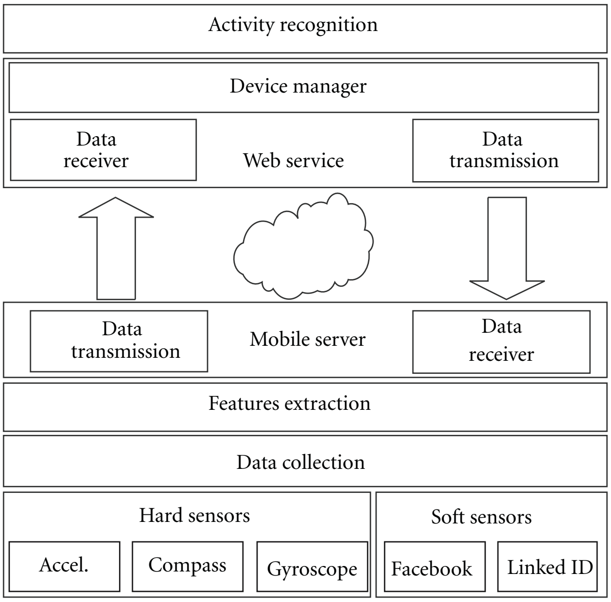


Figure 6: Overall overview of inContexto architecture.

It is largely to recall that the presented architecture is developed to obtain user context in a nonintrusive way. For that reason an smartphone instead of ad-hoc sensors has been chosen since smartphones can supersede these sensors, by reducing user's rejection since they are considered daily communication tools.

Hard sensor data is accessed through Android OS API, in concrete sensor manager class which provides methods to obtain all the mobile sensors. A low-level sensing module continuously gathers relevant information about the user activities using sensors. Thanks to Android OS that provides background processing, it is possible to run services without human control.

In order to provide an effective and efficient description of patterns, preprocessing is often required to improve performance, removing noise and redundancy in measurements. In this study, the accelerations and azimuths of the pedestrian were mainly collected with a smartphone with Android operating system. Andorid OS provides four different sampling frequencies. These frequencies are not fixed and depend on the operating system, and there is no control over it.

The sampling frequency can be adjusted according to the action studied. In this case, relying on the next study [26], the sampling frequency range requiring to obtain human actions is from 0.6 Hz to 2.5 Hz. Consequently, to prevent aliasing problem, the Nyquist-Shannon sampling theorem is as follows:

$$F_E \geq 2 * F_{\text{Max}}. \tag{1}$$

Sampling frequency is not clear in Android OS since it provides only four different sampling frequencies (fastest, game, normal and UI), and the value is not constant. The value depending on the computational workload of the smartphone but normally fastest sampling frequency is 50 Hz.

Besides, accelerometer and GPS raw data have been stored into a sliding window of 512 samples (approximately 5 seconds), 256 of which overlap with consecutive ones. Sliding windows with 50% overlap have been defined in previous works [8].

Besides, extracting features from a window is a fairly effective way to preserve class separability and can represent the characteristics of different activity signals in each window.

Social networks have plenty of information, and most of this information is unused. Thus, the selected features collected from different social networks are social network iD, social network name, born on, lives in, and relationships with others.

Acquiring context from soft sensors is not a banal work. Social network information is accessed thanks to provided APIs by the SNS. Hence, it is necessary that the user log into the site. In this first contact, inContexto was connected with facebook friends and smartphone agenda in order to create ties with people.

Facebook platform (FP) is a connect service which lets third-party application to retrieve SNS features [27]. Besides, FP is an open standard that describes how users can be authenticated in a decentralized manner (Figure 7), obviating the need for services to provide their own ad hoc systems and allowing users to consolidate their digital identities [28].

Facebook platform leverages OAuth 2.0 for authentication and authorization process. First of all, inContexto user authenticates using Facebook as an identity provider. Later, Facebook sends a message that permits inContexto access to the user basic profile (name, profile picture, gender, and friend list).

4.2. Smartphone-Placed Problem. Although there are multiple researches that show the best position to wear sensors [3], sensor placement is a real problem in activity recognition based on MEMS. Minimal changes in sensor placement or orientation create different data and a wrong activity classification. Although previous work suggests that the best place to wear this sensor is the hip [8]; however, in this work, the data collection process was made while the smartphone was worn in a trousers pocket. Hence, it is mandatory to create a system which enables our system to assume a random and possibly changing orientation for the mobile phone.

4.3. Features Extraction Level 1. JDL model depicts that in this level is made object detection process. Although normally object detection is not trivial task, in this case if is it due to tracking mobile phone user actions.

Features extraction level involves the extraction of symbolic features from sensor data obtained in L0. Features can be defined as the abstractions of raw data. The raw sensor data acquired by phones, independent of the amount or source (e.g., accelerometer, camera), are worthless without interpretation. The objective of feature extraction is to represent an activity with the main characteristics of a data segment.

This level aims to process and select which features are better to identify an action. The module processes several sensor observations (a sliding window) into a vector features that help discriminate between activities. The features extraction level is also implemented in the mobile phone.

In the literature, mainly there are two types of extract features from accelerometer raw data. The first ones are those techniques which use frequency properties analysis (DWT, CWT, and STFT), and secondly those that create a vector with statistical methods (SMA, signal mean, correlation, etc.). Barralon et al. [29] present a comparison study using wavelet and frequency features. They present that walking mode is characterized either by the foot impacts on the floor or by chest oscillations. Although the CWT and DWT methods present same performances, the CWT suffers time-consuming problem. Summarizing, frequency methods provide several advantages, one in particular is its resilience to signal level variations.

On the other hand, statistical features presented in [30] are other possible features to infer activities using accelerometer raw data. In this case, mean, standard deviation, and average energy are used which is well-fitted to distinguish sedentary activities from athletic activities; and also correlation between axes. Signal magnitude area (SMA) used in [31] provides good results. SMA is equal to the sum of acceleration magnitude summations over three axes of each window normalized by the window length, giving a total of thirteen attributes (Figure 8).

Furthermore, soft sensor L1 module aims to generate a meta-agenda collecting information for each available

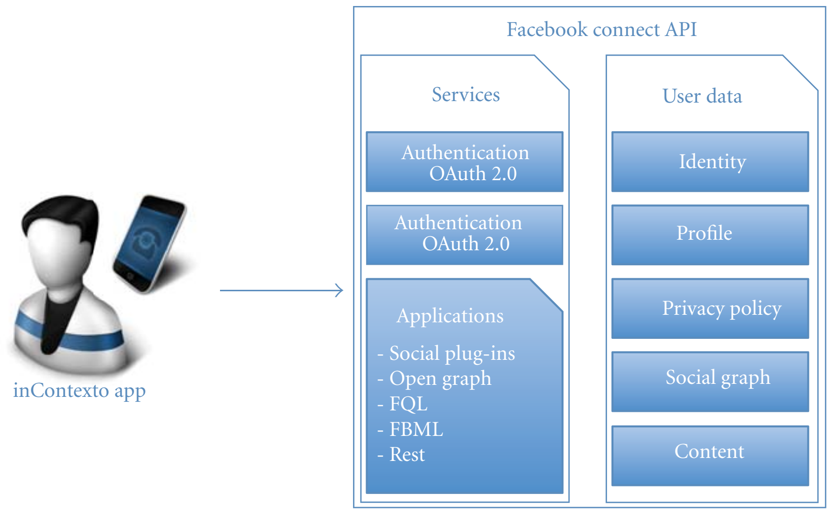


Figure 7: Facebook platform connect architecture.

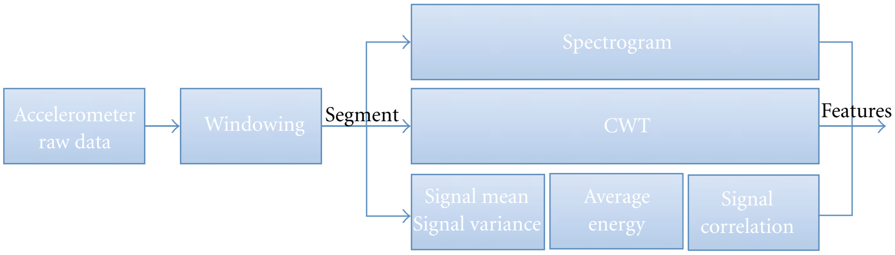


Figure 8: Hard sensor smartphone inContexto architecture.

SNS and the smartphone agenda (SA). The meta-agenda is composed by every person the user knows either on Facebook or SA. Probably, most of these contacts have an instance in both sides (SA and Facebook), therefore, they are joint in the same meta-agenda contact according to the email, name, or mobile phone number coincidence. Meta-agenda permits to create relationships between people and inContexto user (Figure 9).

Moreover, user meta-agenda contact profile is updated with all SNS information available. Summarizing, this new profile contains basically name, date of birth, Mobile phone number, email, and relationships. Besides, some other optional features are collected, for example, likes and dislikes, school degrees, employment, and so on.

4.4. Mobile Server, Web Service, and Device Manager. Both components aim to communicate the smartphone with the server. One of them (Mobile server) is implemented in the smartphones, and the other one (web service) is on the server. The Web service module is developed as web service which is designed to support interoperable machine-to-machine communication over a network. Web-services provide an interface which describe message format, specifically, Web services description language (WSDL) [32]. Device manager allows web services to view and control the devices attached to the service. When a device is not online, the web server keep the last device's IP address for a while, waiting for a new connection.

4.5. Activity Recognition Level 2. JDL level 2 uses the vector features provided by level 1 in order to infer what single activity is the individual engaged in. In this component, it would be implemented the pattern recognition techniques (supervised learning, probabilistic classification, and model-based or instance-based learning) to figure out the action.

Activity recognition level fetches the features selected by the last level and classifies them in order to return the current activities walking, running, sitting, standing, listening to music, talking, and so forth.

J48 decision tree has been chosen since they present several advantages over traditional supervised classification methods used in smartphone sensing. In particular, decision trees are fast in reasoning, so it is a crucial feature in a real-time system like this. In addition, they allow for missing values since it is defined as a classification procedure that recursively partitions a data set into smaller subdivisions. Finally, decision trees are easily interpretable to developers because of the structure.

Level 2 processing develops a description of current user contact actions in the context of their environment. Distributions of individual objects (defined by level 1 processing) are examined to aggregate them into operationally meaningful combat units and weapon systems.

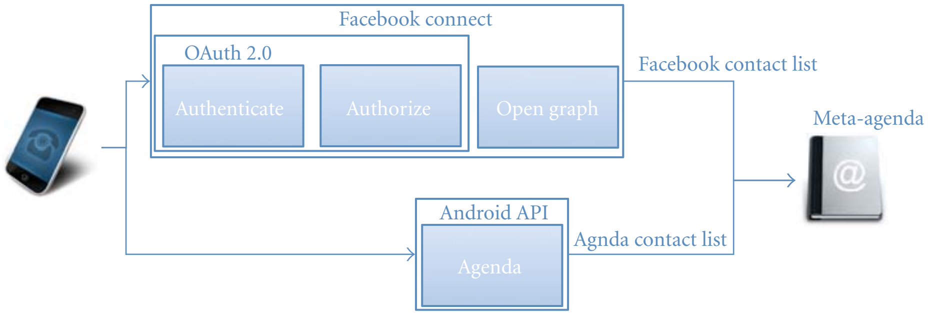


Figure 9: Soft sensor smartphone inContexto architecture.

If the motion context detects an activity, a corresponding message is emitted to the next level (L3), so that other sensors that may be interested in this activity will be triggered (e.g., social context).

4.6. High Level Action Reasoning Level 3. Finally, high level action reasoning level aims to compose all the received actions from the activity recognition level into a context action for each user. Beyond the standard reasoning model based on the subsumption ontology mechanism, it is possible to perform rule-based inferences using a description logic inference engine. At the beginning, these rules would be described by their own user in order to teach the system.

All the simple actions taken by the user would infer a global action with any relation between the other ones. For example, low level promotes running, listening to music, and free time context for a particular user. Maybe all these actions do not make sense in an individual way, but altogether, it could be possible to infer that the user is doing exercise.

For example, the accelerometer and the microphone may detect whether the user is sitting or the user is near to a sound source. If you use both the actions and it is able to locate the action (living room), it could figure out that the person is sitting in the living room watching TV (location action).

5. Experimentation

In order to generate enough trajectories examples to make the training process, the training data was made in a different way. This process has four steps: data collection, trajectories generation, features extraction, and Training process.

Eight male participants between the age groups of 20–37 years have been participated as subjects for the empirical data collection experiments. Users were encouraged to wear the device as much as possible in either of their pockets and perform three different activities (running, walking, and standing up).

Besides, the study relies on the power of the GPS to tag every action that the mobile phone takes. On one hand, every action which takes place outdoor (running, standing, and walking), the data acquisition layer records the speed and precision from the GPS (autotagging).

Finally, a dataset was created for the research community, and it is available online on this website (GIAA Web page http://www.giaa.inf.uc3m.es/).

Table 2: Dataset duration (min) and samples for each activity.

	Running	Standing	Walking
Instances	150,718	345,318	240,825
Minutes	32.36	77.42	40.5

5.1. Data Collection. In this study, the accelerations and azimuths of the pedestrian were collected with Android OS devices. The created dataset has the following attributes: 3-axis accelerometer values in the smartphone Cartesian reference system, 3-axis compass values, 3-axis accelerometer values in the real-world reference system, GPS precision, and GPS speed. Table 2 shows the number of instances for each activity. In this approximation the architecture just acquire data from digital compass, accelerometer, and Gyroscope.

Computing the inclination matrix I as well as the rotation matrix R transforms a vector from the device coordinate system to the world's coordinate system which is defined as a direct orthonormal basis. I matrix is a simple rotation around the X-axis and the rotation matrix R which is the identity matrix when the device is aligned with the world's coordinate system

$$a_{\text{realworld}} = a_{\text{smartphone}} * R * I, \qquad (2)$$

where I matrix is a simple rotation around the X-axis and the rotation matrix R is the identity matrix when the device is aligned with the world's coordinate system.

Figure 10 represents the device accelerations and shows the changes of the three forces depending on the movement taken by the user (running, walking, standing). On the other hand, Figure 11 represents the transformation from the smartphone reference to the real-world reference.

This work uses GPS in order to obtain the speed of the person who is doing the action; thus, the classifier output value is the mean of the speed in the sliding window.

Three different vector features are compared in order to decide what is the best one. The first one is based on spectrogram function ((STFT), short-time fourier transform). A spectrogram is a time-varying spectral representation that shows how the spectral density of a signal varies with time. The second one is continuous wavelet transformation used to split a continuous-time signal into wavelets. Unlike Fourier transform, the CWT is able to construct a time-frequency representation of a signal which offers very good time and

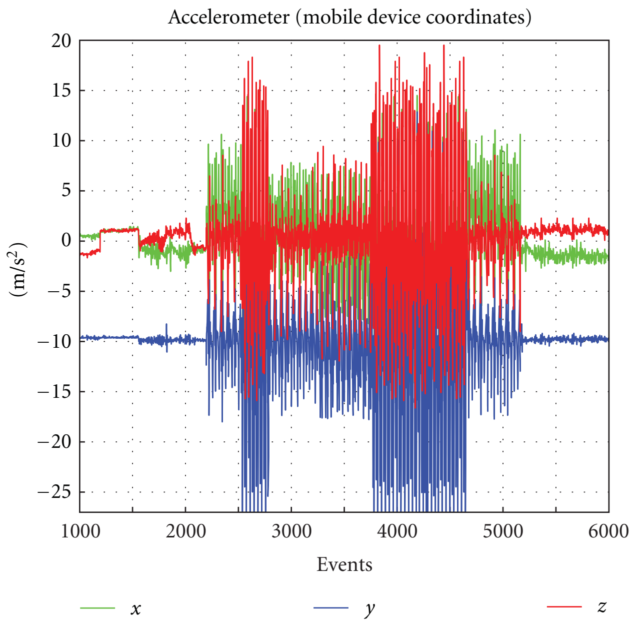


Figure 10: Sensing level: device 3 axes accelerations.

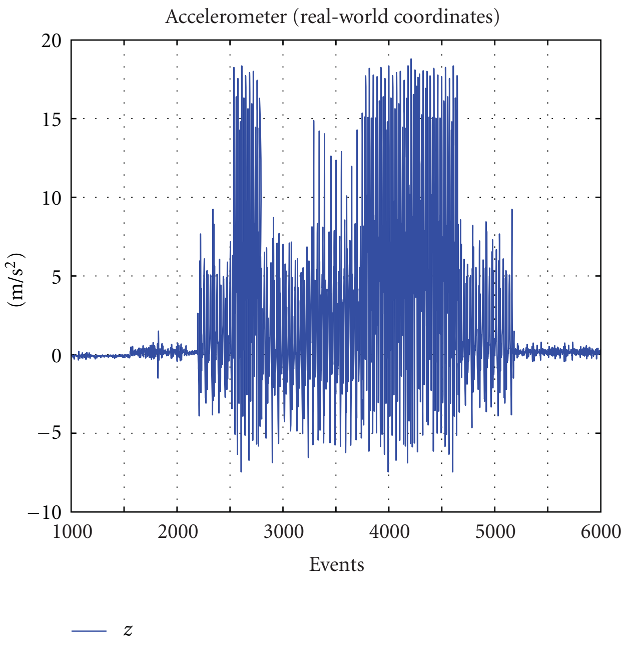


Figure 11: Real-world vertical acceleration.

frequency localization. Both of these techniques (STFT and CWT) present several vales (higher than 150), however, all of them are not necessary. For that reason, only the first 25 frequencies were selected such possible features. Besides, the signals need to be transformed from smartphone coordinate to real-world coordinates. Statistical method consists of eight features, consisting of signal mean, correlation between axes, energy, and variance, which are usually extracted from the triaxial acceleration data.

5.2. Samples Generation. It is necessary that a big amount of samples or trajectory (vector features) make correctly the training process. However, it is quite costly to generate enough samples to make this process.

In this case, the selected sample is made semiautomatically. First of all, we have 3 files corresponding each activity (running, walking, and standing up). Subsequently, a Java program has been created to mix all the activities generated a unique trajectory. Finally, all the generated trajectories have been stored to continue the pattern recognition process. However, there are some requirement to make this trajectories as real as possible:

(1) all the trajectories start with a standing up action;

(2) the next action could be the action besides (Figure 12) or the same action again;

(3) the minimum duration of each action is 2 seconds and the maximum is 7 seconds;

(4) finally, each trajectory consists in 10 actions.

When the trajectories generation process is over, it is necessary to discretize the speed value due to J48 tree users nominal values. Thus, all the samples are discretized in 5 classes.

(i) Stop class: it is when the GPS speed measurements are less than 1 km/h.

(ii) Walking class: speed value from the GPS is more than 1 km/h and less than 4.

(iii) Walking fast class: in this case, GPS speed values are among 4–6 km/h.

(iv) Running class: it is when the GPS speed measurements are more than 6 km/h and less than 10 km/h.

(v) Running fast class: finally, the last class takes the GPS speed values more than 10 km/h.

Finally, 1000 trajectories were created to infer activities. Every trajectory is different, in duration and actions, from the other. Weka (Weka web page http://www.cs.waikato.ac.nz/ml/weka/) was used as the machine learning tool in this paper, and it is necessary to transform data into arff format.

The selected machine learning algorithm is a J48 classifier which is the Weka version from the C4.5 decision tree algorithm. J48 was chosen to give results in tree model which can be easily transformed into real-time applications.

The selected parameters for the J48 decision tree are:

(i) confidence factor = 0.25,

(ii) minimum number of objects = 2,

(iii) unpruned = false,

(iv) test options = 10-fold cross-validation.

5.3. Results. After processing the training and testing sets with the J48 classifier in Weka, the results are highly accurate in vector and spectrogram features; however, results are poorly accurate if CWT features extraction is used.

Table 3 shows results from each selected technique to extract features. The best implemented technique is vector,

Table 3: Features of J4 tree generated by Weka.

	Features	Leaves	Tree size	Accuracy	Mean absolute error
CWT	25	8741	17481	62.85%	0.1631
Spectrogram	25	1007	2013	95.63%	0.0198
Vector	12	648	1295	97.20%	0.0131

Table 4

Research	Classes	Sensors	Time (h)	Accuracy
Cenceme [10]	Still, walk, run	Accelerometer	4	78%
LifeMap [17]	Still, walk, motor	Accelerometer, magnetometer, Wifi, and GPS	28	91%
Borriello [3]	Still, walk, stairs up and down, riding elevator and brushing	GSM, Wifi	7	84%
InContexto	Still, walk and run	Accelerometer	2	97%

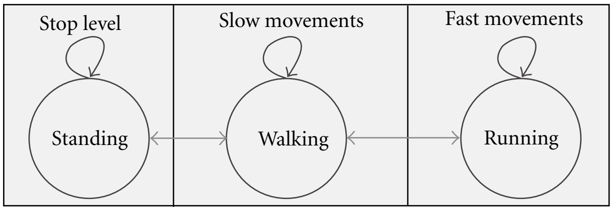


Figure 12: Generation trajectories model.

which is not only more accurate than the other ones; otherwise, it provides the smallest tree generated. The size of the tree is very important since this technique will be implemented in a real application in a mobile phone. A bigger size of the tree causes more energy consumption according to the increase of CPU cycles. Another way to study the quality of the feature extraction techniques is using the confusion matrix (Figure 13).

CWT technique is the worst of all the studied techniques, besides, it does not present any advantage over the other ones. Secondly, spectrogram achieves great results; besides, this technique uses only one signal (vertical movement in the real world) in order to obtain the spectrogram although confusion matrix shows that it is possible to classify an instance in a class not next to the real class. Thus, the best performance (high accurate and less tree size) is presented by vector technique. Besides, confusion matrix figure shows that vector features extraction just fail with the class near the one which is classified (e.g., running instead of running fast) (Table 4).

5.4. Energy Consumption Problem. Resource constraints power consumption is the main factor affecting smartphone activity recognition system. It is highly desirable that inContexto architecture is running as long as possible.

Normally, embedded sensors are placed in the same chipset. In this case, it is used as an HTC magic smartphone which is AK8976A marketed by Asahikasei Microsystems Co., Ltd (AKM). This chipset includes a 6-axis electronic compass that combines a 3-axis geomagnetic sensor with a 3-axis acceleration sensor in an ultrasmall package. Consequently, whether your applications query the accelerometer, compass or both, it consumes the same energy power.

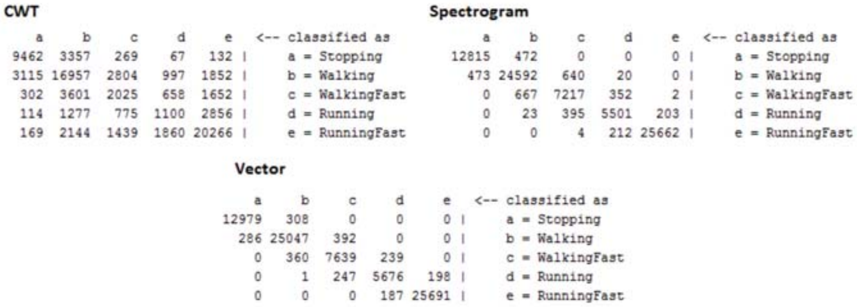
Besides, communication process between smartphone and the cloud consumes energy. This is very expensive and takes a toll on battery life. Reducing the number of upload, the system preserves energy. Considering computational power and energy consumption restrictions, it is necessary to select a good technique in order to balance the energy consumption and the global precision of the system. One way to do that is doing features extraction process on the mobile phone and creating a sliding window to reduce the amount of data.

Figure 14 shows inContexto energy consumption during 30 minutes. This value is about 21% of the 20% of the smartphone total energy. InContexto energy consumption is not high comparing with the screen value (66%) or Wi-Fi (11%).

6. Conclusions

In this paper, it inContexto was presented a distributed architecture to obtain mobile context from smartphones. The proposed architecture distributes the processing load between smartphone and a server placed on the cloud. With this approach, the energy consumption is reduced, increasing autonomy to offer a better service to the user. Also, a study comparing three different techniques in order to infer activity recognition using a J48 decision tree was presented. Besides, the study relies on inContexto architecture to collect accelerometer data. Overall, the presented work further demonstrates that using a mobile phone providing with accelerometers is enough to infer actions that user is doing.

Besides, a smartphone entity is defined according to Dey and Abowd's definition of context aware. An entity is defined as a smartphone which provides hard or/and soft sensors, provides internet connection everywhere, and is portable.



CWT

```
   a     b     c     d     e   <-- classified as
9462  3357   269    67   132 |     a = Stopping
3115 16957  2804   997  1852 |     b = Walking
 302  3601  2025   658  1652 |     c = WalkingFast
 114  1277   775  1100  2856 |     d = Running
 169  2144  1439  1860 20266 |     e = RunningFast
```

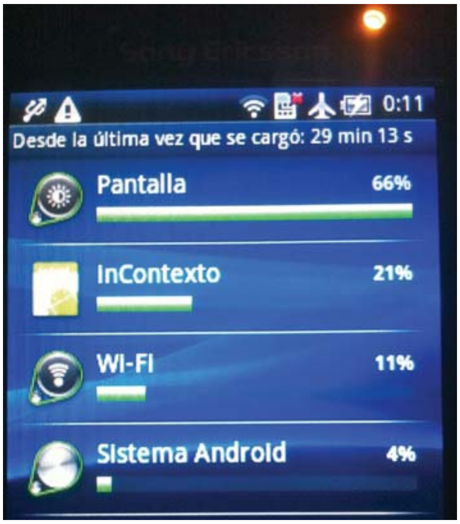
Spectrogram

```
    a     b     c     d     e   <-- classified as
12815   472     0     0     0 |     a = Stopping
  473 24592   640    20     0 |     b = Walking
    0   667  7217   352     2 |     c = WalkingFast
    0    23   395  5501   203 |     d = Running
    0     0     4   212 25662 |     e = RunningFast
```

Vector

```
    a     b     c     d     e   <-- classified as
12979   308     0     0     0 |     a = Stopping
  286 25047   392     0     0 |     b = Walking
    0   360  7639   239     0 |     c = WalkingFast
    0     1   247  5676   198 |     d = Running
    0     0     0   187 25691 |     e = RunningFast
```

FIGURE 13: Confusion matrix of each technique.



0:11

Desde la última vez que se cargó: 29 min 13 s

Pantalla 66%

InContexto 21%

Wi-Fi 11%

Sistema Android 4%

FIGURE 14: InContexto energy consumption.

Activity recognition systems identify and record in real-time selected features related on user activity using a smartphone. The paper describes how to face this problem using information fusion architecture in smartphones. Besides, it describes sensing module process, that is, one of the most important components in activity recognition systems.

The best given solution obtained an overall accuracy of 97.20% well to classify instances of 79250 different actions. This solution is a vector composed by energy, mean, standard deviation, and correlation of each axes.

The flexibility of the Android OS along with the phone's hardware capability allows this system to be extended, for example, creating an application which is able to send an sms or call to your relatives if you are doing strange movements.

Considering future works extends the development of the server module, and also it will extend activity classifier to more complex activities (group activities, interaction activities). Context information will be used to infer the user's emotional state, for example, according to the social network state, the music which is listened at the moment, the place where the user exists and using other hard and low sensors.

Acknowledgment

This work was supported in part by Projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485), and DPS2008-07029-C02-02.

References

[1] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[2] R. K. Ganti, S. Srinivasan, and A. Gacic, "Multisensor fusion in smartphones for lifestyle monitoring," in *Proceedings of the International Conference on Body Sensor Networks (BSN '10)*, pp. 36–43, June 2010.

[3] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proceedings of the 4th International Conference on Pervasive Computing (PERVASIVE '06)*, pp. 1–16, May 2006.

[4] L. Liao, *Location-based activity recognition*, Ph.D. thesis, 2006.

[5] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[6] J. Gómez-Romero, M. A. Serrano, M. A. Patricio, J. García, and J. M. Molina, "Context-based scene recognition from visual data in smart homes: an Information Fusion approach," *Personal and Ubiquitous Computing*. In press.

[7] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: a survey," in *Proceedings of the 23rd International Conference on Architecture of Computing Systems (ARCS '10)*, pp. 1–10, 2010.

[8] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of the 2nd International Conference on Pervasive Computing (PERVASIVE '04)*, pp. 1–17, Vienna, Austria, April 2004.

[9] M. Marin-Perianu, N. Meratnia, P. Havinga et al., "Decentralized enterprise systems: a multiplatform wireless sensor network approach," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 57–65, 2007.

[10] E. Miluzzo, N. Lane, K. Fodor et al., "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 337–350, 2008.

[11] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[12] J. Llinas, “Revisiting the jdl data fusion model ii,” Tech. Rep., DTIC Document, 2004.

[13] P. Barralon, N. Noury, and N. Vuillerme, “Classification of daily physical activities from a single kinematic sensor,” in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS ’05)*, pp. 2447–2450, September 2005.

[14] J. Garcia, J. Molina, A. Berlanga, and G. de Miguel, “Data fusion alternatives for the integration of millimetre radar in airport surveillance systems,” in *Proceedings of IEEE International Radar Conference*, pp. 796–801, 2005.

[15] M. Markin, C. Harris, M. Bernhardt et al., *Technology foresight on data fusion and data processing*, Publication of The Royal Aeronautical Society, 1997.

[16] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, “A survey of mobile phone sensing,” *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[17] J. Chon and H. Cha, “LifeMap: a smartphone-based context provider for location-based services,” *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, 2011.

[18] B. N. Schilit and M. M. Theimer, “Disseminating active map information to mobile hosts,” *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994.

[19] A. Schmidt, K. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde, “Advanced interaction in context,” in *HandHeld and Ubiquitous Computing*, pp. 89–101, Springer, 1999.

[20] A. Dey and G. Abowd, “Towards a better understanding of context and context-awareness,” in *Proceedings of the Workshop on the What, Who, Where, When, and How of Context-Awareness (CHI ’00)*, vol. 4, pp. 1–6, 2000.

[21] A. Schmidt, M. Beigl, and H. W. Gellersen, “There is more to context than location,” *Computers and Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[22] W. W. W. C. Editors, “Geolocation API specification @ONLINE,” February 2010.

[23] R. DeVaul and S. Dunn, “Real-time motion classification for wearable computing applications,” 2001.

[24] F. van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*, Artech House, 2009.

[25] D. M. Boyd and N. B. Ellison, “Social network sites: definition, history, and scholarship,” *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[26] M. Henriksen, H. Lund, R. Moe-Nilssen, H. Bliddal, and B. Danneskiod-Samsøe, “Test-retest reliability of trunk accelerometric gait analysis,” *Gait and Posture*, vol. 19, no. 3, pp. 288–297, 2004.

[27] M. N. Ko, G. P. Cheek, M. Shehab, and R. Sandhu, “Social-networks connect services,” *Computer*, vol. 43, no. 8, pp. 37–43, 2010.

[28] D. Recordon and D. Reed, “OpenID 2.0: a platform for user-centric identity management,” in *Proceedings of the 2nd ACM Workshop on Digital Identity Management (DIM ;06)*, pp. 11–16, November 2006.

[29] P. Barralon, N. Vuillerme, and N. Noury, “Walk detection with a kinematic sensor: Frequency and wavelet comparison,” in *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS ’06)*, pp. 1711–1714, August 2006.

[30] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Proceedings of the 20th National Conference on Artificial Intelligence*, vol. 3, pp. 1541–1546, July 2005.

[31] J. Y. Yang, J. S. Wang, and Y. P. Chen, “Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers,” *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213–2220, 2008.

[32] W. W. W. C. Editors, “Web services description language 1.1 @ONLINE,” Mar 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 515046, 8 pages
doi:10.1155/2012/515046

Research Article

Spatial and Temporal Correlations-Based Routing Algorithm in Intermittent Connectivity Human Social Network

Zhou Tao,[1] Xu Hongbin,[1] Liu Ming,[2] Liu Nianbo,[2] and Gong Haigang[2]

[1] *School of Automation, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Liu Ming, wing.lm@gmail.com

Received 13 December 2011; Accepted 14 January 2012

Academic Editor: Mo Li

The social network formed by people is one of the key applications of Delay-Tolerant Network (DTN). Owing to its intermittent connectivity and unique human mobility patterns, how to transmit data in an effective way is a challenging problem for the social network. In this paper, we propose the idea of Trip History Model (THM) which establishes a model on a single person's mobility, and then a Spatial and Temporal Correlations-Based Routing Algorithm (STC) is proposed. In STC, the node delivery probability is calculated according to both a node's current moving prediction and its history record to give guidance for message transmission. Our simulation results show that, compared with LABEL and PROPHET algorithms, STC effectively improves the routing performance of the network.

1. Introduction

The application of wireless network adapters in portable communication devices, such as mobile telephone, personal digital assistants, and laptops, can provide more chances to access global information resources. These portable devices can be connected to dynamic networks. By such networks, people can transfer information to each other. The movement of people and the limited communication range make the network unable to always guarantee its own connectivity. To make communication possible, DTNs (delay tolerant networks) [1] were proposed, which overcome the problems associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates by using store-and-forward message switching. PSN (pocket switched network) is a kind of DTNs, which is composed of mobile communication devices and has broad application in the future.

Because of the dynamic topology and the intermittent connectivity of PSN, people can only communicate by using store-and-forward message switching when they meet. Currently, many proposed PSN routing algorithms are based on historical records, and data is transmitted to the node which can constantly communicate with destination node. Since these algorithms fail to predict the trajectories of mobile nodes, they cannot guarantee the reasonableness of data forwarding. In addition, [2] shows that many people will move according to some certain rules. For instance, they like moving to specific sites periodically; in this case, regular space and time rules can help to guide routing in PSN.

Based on human movement discipline, this paper proposes Trip History Model (THM) to automatically predict the destination of people by studying their trip history recorded by portable communication devices. Meanwhile, according to the spatial and temporal characteristics of human movement, STC (Spatial and Temporal Correlations Routing Algorithm) is designed to forward the message to next node according to the prediction of the current node. Simulation results show that STC effectively improves the routing performance.

2. Related Work

In recent years, there has been extensive research on routing protocols in delay-tolerant networks. Vahdat and Becker [3] present epidemic routing protocol, which is the representative flooding routing protocol. It attempts to give all nodes a copy of every message through the communication with them to achieve high delivery ratios. Flooding routing protocol requires no knowledge about the sensor network, but it also has some defects. For instance, since there exists huge amounts of copies of messages, a large proportion of network bandwidth and node buffer resources will be occupied. Meanwhile, more energy of the sensor nodes will have to be consumed to send and receive these copies messages. Based on the research on human society, some scholars proposed the community-based intermittent connectivity routing algorithm.

Literature [4] proposes a Replication-based Efficient Data Delivery (RED) scheme, which consists of two key components for data transmission and message management, respectively. Data transmission scheme decides when and where to transmit data messages based on the delivery probability, and message management scheme makes decisions on the optimal erasure coding parameters according to its current delivery probability. However, the calculation of erasure coding parameters is not a refined one, and a lot of data blocks split by RED will consume lots of energy of the sensor nodes.

Literature [5] presents a LABEL routing scheme, which assumes every node belongs to a community and marks nodes in the same community with the same label. When a node needs to send a message, it will only select the nodes within the communication range which belong to the destination node community as the next hop node. However, when the source node is far from the destination node community, the message cannot be transferred. For this reason, Hui and Crowcroft [5] propose a BUBBLE routing scheme, which takes the node activity into account. Each node has a global level and a local level; the higher its global level is, the more active the node will be and thus having the stronger ability to transfer messages.

In addition, Daly and Haahr [6] present a delay-tolerant network routing technique (SimBet) based on the small-world social phenomenon, which comes from the observation that individuals are often linked by a short chain of acquaintances. The betweenness centrality and the social similarity value are jointly decide the Simbet value of each node. When two nodes meet in the network, the message will be transferred to the node with larger Simbet value in order to improve the delivery performance. Simulations using three real trace data sets demonstrate that the achieved delivery performance may be closed to that of Epidemic Routing but the overhead can be significantly reduced.

3. Analysis

In recent years, great progress has been made in the research of human complex social systems. Based on the tracking data of the wireless mobile network users, Hsu et al. [7] sum up two characteristics of people's movement, that is, preferring specific locations and going to the same location periodically. Based on the tracking data of thousands of anonymous mobile phone users, González et al. [8] propose people's activities have regular space and time rules. These research results prove the movement of people is nonrandom and it has regular space and time rules. In addition, a project named Mobidrive [9] recorded the movement of people in six weeks and found that human movement between 4 to 6 major locations may occupy more than 70% of all the movements. Therefore, the movement of people can be described as a spatial one between regular locations and thus the conclusion that the movement of many persons can be predicted.

The current human movement models fall into the following categories: firstly, the random movement model, including random waypoint, random walk, and so forth. In the random movement model, node randomly select the destination, direction, and thinking times, and so forth. The movements of all nodes in the models mentioned before follow the same rule. These models have simple structure and can be clearly defined, but they cannot reflect the movement features of people in real world for its lack of authenticity. Second, modeling and analyzing the movement features of people are done based on the collection of the real trajectory. Chaintreau [10] considers intercontact time of portable wireless devices nearly obeys the power-law distribution rather than the original exponential distribution. For such models, the main difficulty lies not in the model itself, but in how to organize data and obtain the implied movement mode through a series of mathematical methods. Third, the movement model is established based on community. Literature [11] divides people into different groups based on the social relations between them. These groups are randomly placed in the simulation area and the nodes select their destination and move according to certain rules. Literature [12] introduces agenda-based routing protocol (ARP), which considers the relationship between human social activities and movement locations based on the survey data of US Transportation Department, but it does not consider the two characteristics of human movement, that is, spatial and temporal. The models above have their defects, and therefore we propose Trip History Model that takes into account the two characteristics of human movement to predict the human activities, which can be easily saved and processed.

4. Network Model

4.1. Trip History Model. This paper assumes that the sensor nodes are distributed in a two-dimensional $M \times M$ square region A. All the sensor nodes have the same transmission radius R. Network can be described as an undirected graph $G = (V, L)$; V is the set of all nodes in the network, and L is the set of connections in the network. For any two nodes V_i and V_j, if the distance between them is less than R, there exists a connection between them and let *neighbor* (v_i) stand for the neighbor of V_i.

In addition, the movements of all the sensor nodes follow the Trip History Model. The power of portable devices can be replenished in time, so we assume that the energy of sensor nodes is infinite.

We consider the human activities have certain rules and can be described as a series of data including time and location. All the records of time and location of a person's activities constitute his travel history. Since a person's activities include some repetition every day, we save a person's trip in a day as a record cycle. Meanwhile, a day can be divided into two discrete sets, that is, Day (Mon., Tues., Wed., Thurs., Fri., Sat., Sun.) and Time (<8, 8-9, 9–11, 11-12, 12-13, 13–15, 15–17, 17–19, >19). For instance, the trip history of a person can be described as follows:

(1) Mon. 7:40 move from home to workplace along a certain way.

(2) Mon. 17:30 go home from the office and pass by the supermarket for some shopping.

(3) Sat. 10:20 move from home to the shopping mall.

(4) Sat. 18:30 go home from the shopping mall and pass by the restaurant to eat something.

The forth trip record presents a stopover, and such behavior is true in real life. The trip history above can be described as in Table 1.

In addition, for getting easy computing and saving, we adopt some methods to change continuous location into discrete value. Home or Office, or Supermarket, and so forth is not a real GPS location, but a neighborhood GPS location set. People probably play and move near home and office, and the value of GPS may be some measurement errors, so we can consider the positions within a certain range (100 meters) as the same position.

When a person starts moving, the portable device records the current time and position until he reaches the destination. After a trip is finished, the portable device compares the trip with the shortest way according to the map. If the trip is the shortest way, the portable device will save the time, source, and destination and delete other trip history data. Otherwise, it records the time, source, destination, and the midpoints which the person passed by.

4.2. Decision Tree. We use a decision tree to save Trip History Model and help make some decisions on transferring messages in order to improve the delivery performance. As Figure 1 shows, THM can be presented as the branches and leaves of the decision tree. P_{kq} is the probability that q is destination and can be calculated as

$$P_{kq} = \frac{f_{kq}}{N_k}, \tag{1}$$

where f_{kq} is the number of trips according with specific time, midway point, and destination choices at leaf node kq. Since category kq gives a final destination choice, P_{kq} represents the probability of moving toward the destination in history. Since the car knows current Source, Day, and Time values, it can find a Destination choice with a maximum P_{kq} value in the tree as an initiate prediction.

During the movement of people, the portable device will continue to check the current position and identify, if people are still on the way, the predicted midpoint, or destination. If the position disagrees with the predicted route, the new destination prediction will be calculated as

$$P_{kq} = \begin{cases} 0 & \text{if } q \text{ is infeasible,} \\ \dfrac{f_{kq}}{\sum f_k} & \text{other,} \end{cases} \tag{2}$$

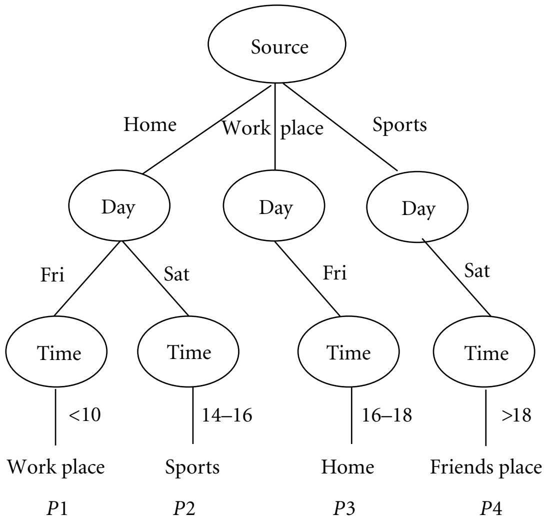


Figure 1: The structure of decision tree with trip history model data.

where $\sum f_k$ is the total number of the rest of the trips after removing the infeasible ones.

5. Spatial and Temporal Correlations Routing Algorithm (STC)

The trip history data is collected by the Trip History Model, including spatial and temporal information. To make use of the history data and to predict the movement of the people (nodes) in near future, we introduce spatial and temporal correlations routing algorithm (STC).

5.1. The Encounter Information Table. Assume each node n saves the encounter information including ID, encounter frequency, and predicted destination, as in Table 2.

Assume the node V_i's predicted destination is changed, so V_i will send a special message to all its neighbor nodes. This message includes the ID of the node V_i and the new predicted destination. And the message disseminates as epidemic-like approaches, until all nodes in the network receive this message. When node V_j receives this message, it will do the following.

(1) If V_i's information has be one included in V_i, V_j updates V_i's predicted destination.

(2) If no, V_j adds V_i's information in its encounter information table, and sets the Encounter Frequency is zero.

5.2. Transmission Probability. Let P_{io} stand for the transmission probability of node V_i transferring the message to node V_o:

$$P_{io} = \frac{\theta}{180^\circ}, \tag{3}$$

Table 1: Records of the trip history.

No.	Day	Time	Source	Midpoint	Destination
1	Mon.	<8	Home		Office
2	Mon.	17–19	Office	Supermarket	Home
3	Sat.	9–11	Home		Shopping Mall
4	Sat.	17–19	Shopping Mall	Restaurant	Home

Table 2: Encounter information table.

ID of encounter node	Encounter frequency	Predicted destination
5	2	D1
21	5	D2
3	3	D3
12	1	D1

where θ is the angle between two tangents of the predicted destination of node V_i to the transmission radius of predicted destination of node V_o as Figure 2. According to trigonometric formulas, we can get

$$\theta = 2\arcsin\left(\frac{R}{d}\right), \tag{4}$$

where R is the transmission radius of node V_o, and d is the distance between the predicted destination of node V_i and the predicted destination of node V_o.

5.3. Message Transmission. Suppose node V_i needs to send message to node V_o. Firstly, V_i would know V_o's predicted destination. If there is not the V_o's information in V_i's encounter information table, V_i sends inquiring message to its neighbor nodes. If its neighbor nodes also have not the V_o's information, it will inquire to more nodes, until it gets the V_o's information. Second, V_i gets its neighbor's predicted destination by shake hands; if its neighbor nodes' predicted destination is same as V_o, it will send the message to these neighbor nodes, if not, V_i will send the message to the neighbor node which has the higher transmission probability of V_o than V_i.

6. Simulation Results

6.1. THM Performance. First, the performance of trip history model is evaluated through the investigation of human trip. In order to get human real trip data, we conducted a 9-week investigation. Invite 5 campus staff members to participate in the survey, and these volunteers have smart handheld devices. In the survey, the drivers need to record the start time, start position, stop position, and midpoint position. The collected trip history data is shown in Table 3. We input these data into the electric map of Chengdu and establish trip history model for each volunteer. Finally, we divide the trip history in Table 3 into two parts. The trip history data in 1–8 weeks regarded as the training set is used to generate the decision tree, and the trip history data in the 9th week considered as the target set is used to evaluate the predicted result.

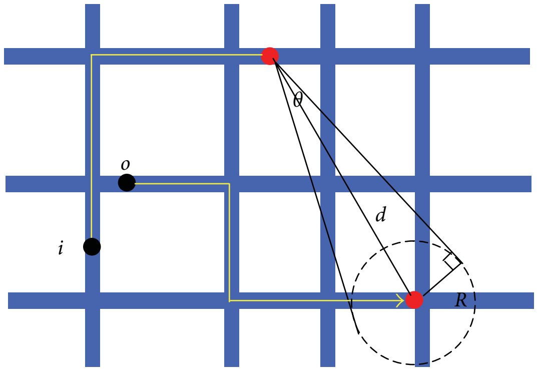


Figure 2: Transmission probability calculation (blue lines present the streets).

Table 3: Trip history collection of investigation.

People	1–8 week	9 week	Total	Average
A	155	17	172	2.73
B	214	29	243	3.86
C	169	18	187	2.97
D	181	25	206	3.27

When the vehicles began to travel, completed a quarter of the trip, half of the trip, three quarters of the trip, and all the trip, the hitting rate compared the predicted destinations of the decision tree with the real destinations, and the results are shown in Figure 3. The average starting accuracy rate is 65.83%, and with the movement of vehicles, the average prediction accuracy rate increases gradually, from 77.56% to 82.77%, to 91.42%, and finally to 97.04%. Although individuals' daily activities are highly regular, the trip in 1–8 weeks cannot fully cover the schedule in the 9th week. Unpredicted circumstances cannot be avoided, so the final average prediction accuracy cannot reach 100%.

6.2. STC Performance. To compare STC to other algorithms, we simulate STC, LABEL, and PROPHET algorithms. In PROPHET algorithm, each node calculates the transmission probability for different destination nodes and only transmits the message to the node which has larger transmission probability for the destination nodes.

6.2.1. Simulation Setting. Assume that the simulation area is 2000 × 2000, including 200 nodes and 60 sites which

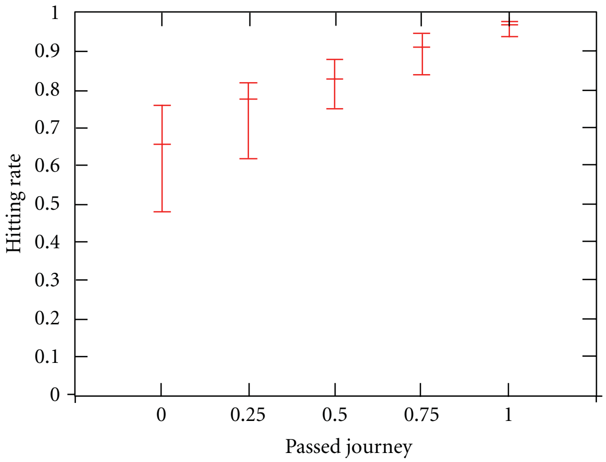


Figure 3: The prediction accuracy of trip history model.

Table 4: The performance comparison under default parameters.

	STC	LABEL	PROPHET
Delivery ratio (%)	85.6	48	83
Average copies for each message	7.3	1.5	10.5
Average delay(s)	108.7	131.4	110.2

consist of 20 "homes", 15 schools, 15 offices, and 10 shops or restaurants. These sites are randomly distributed in the simulation area. In addition, we define three types of nodes, namely, students, workers, and other personnel. For students, the first everyday activity is going to school as it is going to office for workers. Other personnel's first everyday activity is going to other sites except schools and offices (the first activity is randomly selected between 6:00 AM to 8:00 AM). All the people will go home and rest before 12:00 PM Assume that the network bandwidth is 8 Kbps. The message queue length of the nodes is 500, and the survival time of the message is 300 seconds. The simulation duration is 3 days, and the simulation results are as follows.

6.2.2. The Position of Nodes at Different Time. Figures 4 and 5 show the nodes' positions in the simulation area at 8:00 AM and 12:00 AM. The positions of nodes in Figure 4 are different from those in Figure 5, and this shows nodes have already moved. The positions of the nodes are randomly distributed at different times because the sites are randomly distributed in the simulation. In addition, we also find that the density of nodes in the central region is bigger than that in the edge region, and the nodes tend to move to the central region. Considering the actual situation, there will be an uneven distribution of nodes, for example, more nodes should be at "home" in the evening.

6.2.3. Performance Comparison. The following is the performance comparison of three algorithms under default parameters in Table 4. STC's delivery ratio is bigger than PROPHET's delivery ratio, but the average number of the copies of each message in STC is lower than that in PROPHET, and this shows the former's average transmission energy consumption is smaller. In addition, we do not consider the energy consumption of exchanging routing tables between two nodes in PROPHET. If the network has more nodes, the energy consumption will be enormous. In STC, when two nodes meet, they do not need to exchange the routing information and much energy will be saved. Table 4 shows the average delivery ratio of LABEL is the lowest and the average delay of LABEL is larger than that of STC, because nodes only transmit collected data to the node with same predicted destination with them in LABEL, and so message transmitted slowly and transmission delay will be longer.

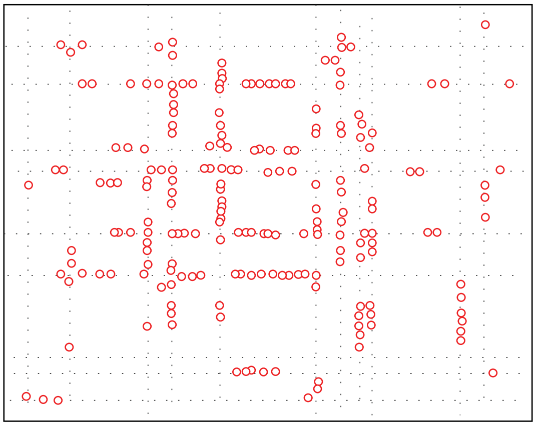


Figure 4: Node distributions at 8 AM.

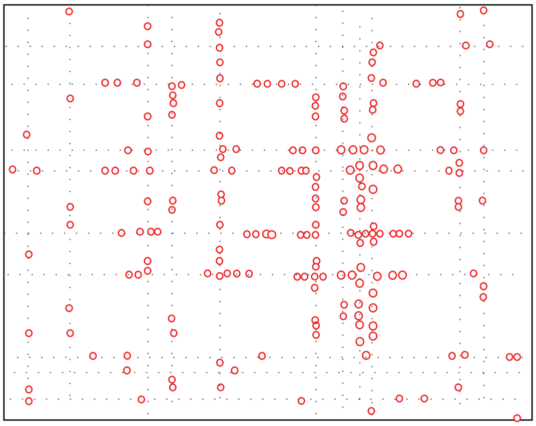
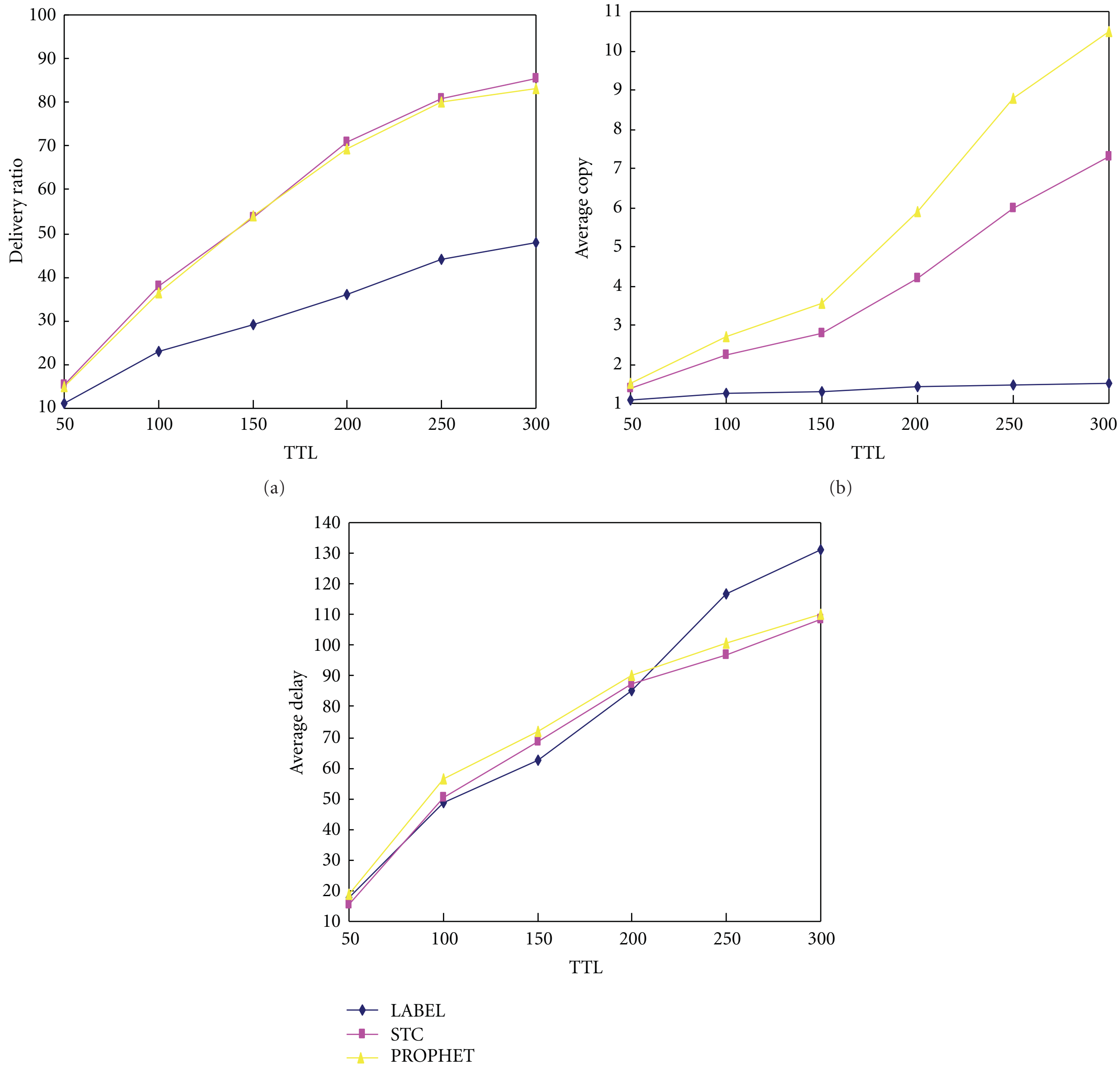


Figure 5: Node distributions at 12 AM.

6.2.4. TTL's Impact on Performance. We change TTL (Time To Live) of messages from 50 s to 300 s, and the simulation results are shown in Figures 6(a)–6(c). In the figures, with the increase of TTL, the average delivery ratio of these three algorithms will increase. There is little difference between STC and PROPHET in terms of the delivery ratio, but compared with LABEL's delivery ratio, theirs are much higher. It shows that the longer survival time the messages have, the more opportunities they get to be transferred to the destination node. Figure 6(b) shows the average number of the copies of each message will increase with the growth



(a)

(b)

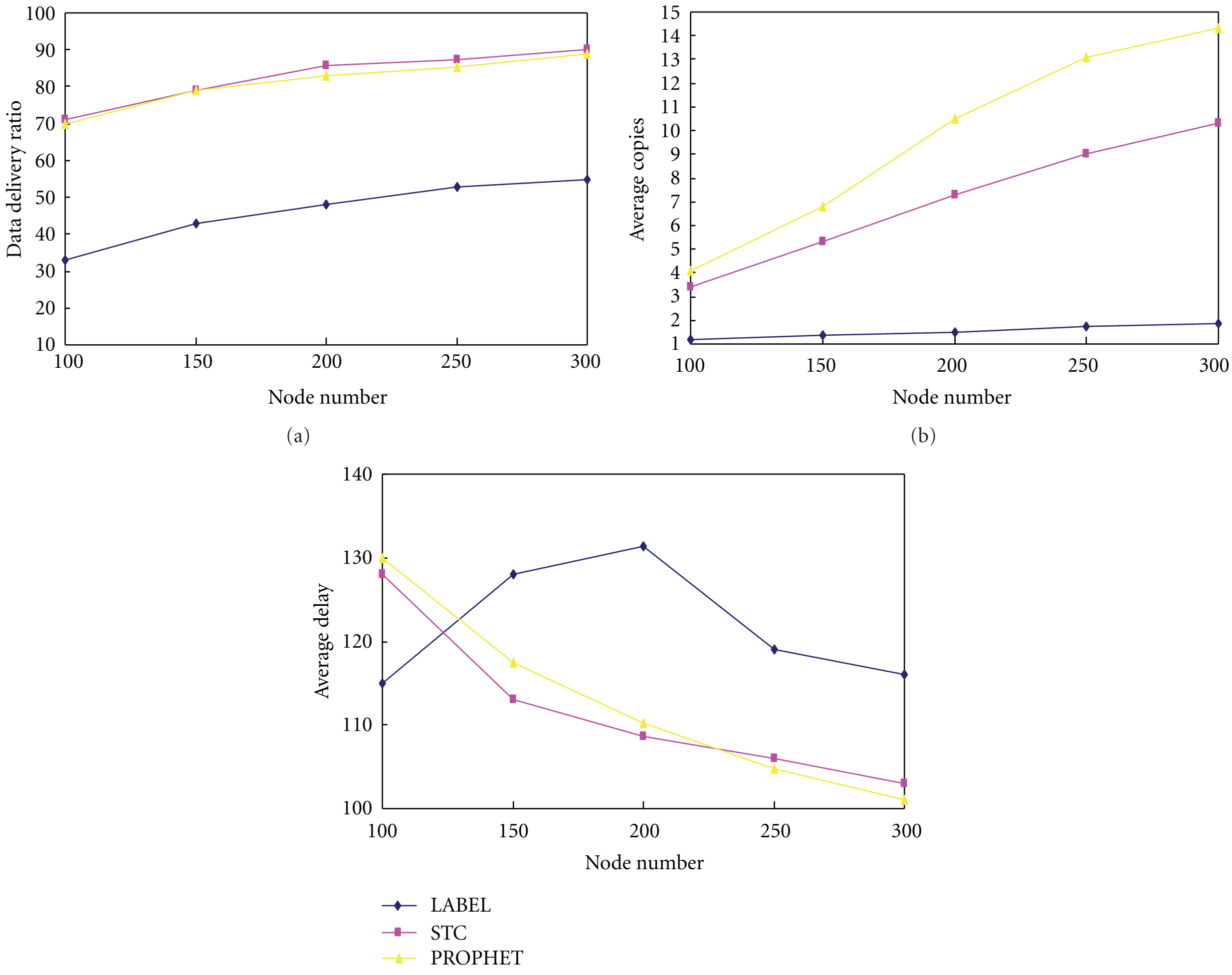
(c)

Figure 6: TTL's impact on performance. (a) Average delivery ratio. (b) Average copies number. (c) Average delay.

of TTL. Although there is little difference between STC and PROPHET in terms of the delivery ratio, the average number of the copies of each message in STC is much lower than that in PROPHET, which implies that STC has less energy consumption. Figure 6(c) shows that average delay increases with the growth of TTL, and the average delay in STC is slightly lower than that in PROPHET.

6.2.5. Node Density's Impact on Performance. The total number of nodes is increased from 100 to 300 to change the node density, and the simulation result is shown in Figures 7(a)–7(c).

Figure 7(a) shows that in all the three algorithms the delivery ratio will increase with the growth of node density. The increase in the number of nodes will cause a change in the transmission range of each node; as a result, there will be more next hop nodes satisfying the transmission conditions, and thus increasing the delivery ratio. Figure 7(b) shows the average number of the copies of each message will increase with the growth of node density. Although there is little difference between the delivery ratio in STC and that in PROPHET, the average number of the copies of each message in STC is much smaller than that in PROPHET. Figure 7(c) shows the average delay in STC is much lower than LABEL with the number of nodes increasing. With the increase in the number of nodes, there will be more message copies in the network, and it will reduce the number of hops between the source node and the destination node.



(a)

(b)

(c)

Figure 7: Node density's impact on performance. (a) Average delivery ratio. (b) Average copies number. (c) Average delay.

7. Conclusions

This paper explores the trip history model (THM) of people's activities and proposes spatial and temporal correlations routing algorithm. In view of the observation that many people have some regularities in their daily movements, we create a model (THM) to describe the mobility characteristics of people who may be treated as mobile nodes and further make motion prediction. Investigation and experiment results have shown that THM can more accurately predict the moving destination of individuals.

In order to make better use of THM, we propose spatial and temporal correlations routing algorithm (STC). Messages route based on the prediction of node movements in THM, and data will always be transferred to the node which has higher transmission probability of meeting the destination node. Combined with judgments of human's trajectory, STC effectively avoid the blindness in data forwarding caused by the random movement model. According to simulation results, STC can obtain a higher data delivery ratio with lower energy consumption and transmission delay.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants nos. 60903158, 61170256, 61173172, 61103226, 61103227.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'03)*, pp. 27–34, Karlsruhe, Germany, August 2003.

[2] J. Crowcroft and P. Hui, "Pocket switched networks: real-world mobility and its consequences for opportunistic forwarding," Tech. Rep., Association for Computing Machinery, New York, NY, USA, 2005.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep., Duke University, Durham, NC, USA, 2000.

[4] Y. Wang and H. Y. Wu, "Replication-based efficient data delivery scheme (RED) for delay/fault-tolerant mobile sensor

network (DFT-MSN)," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 485–489, IEEE Computer Society Press, Washington, DC, USA, 2006.

[5] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2007*, pp. 65–70, White Plains, NY, USA, March 2007.

[6] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, Montreal, Canada, September 2007.

[7] W. J. Hsu, T. Spyropoulost, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM'07)*, pp. 758–766, Anchorage, Alaska, USA, May 2007.

[8] M. C. González, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[9] S. Schonfelder and U. Samaga, "Where do you want to go today?—More observations on daily mobility," in *Proceedings of the 3rd Swiss Transport Research Conference*, Monte Verita, Ascona, 2003.

[10] A. Chaintreau, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM '06)*, 2006.

[11] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: From Theory to Reality (REALMAN '06)*, pp. 31–38, Florence, Italy, May 2006.

[12] Q. Zheng, X. Hong, and J. Liu, "An Agenda-based Mobility Model," in *Proceedings of the 39th IEEE Annual Simulation Symposium (ANSS-39-2006)*, p. 8, Huntsville, Ala, USA, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 365421, 12 pages
doi:10.1155/2012/365421

Research Article

Statistically Bounding Detection Latency in Low-Duty-Cycled Sensor Networks

Yanmin Zhu[1, 2]

[1] *State Key Laboratory of Software Development Environment, Beijing, China*
[2] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Yanmin Zhu, yzhu@cs.sjtu.edu.cn

Received 18 August 2011; Accepted 11 October 2011

Academic Editor: Mo Li

Detecting abnormal events represents an important family of applications for wireless sensor networks. To achieve high performance of event detection, a sensor network should stay active most of the time, which is energy inefficient for battery driven sensor networks. This paper studies the fundamental problem of bounding detection delays when the sensor network is low duty cycled. We propose a novel approach for statistically bounding detection latency for event detection in sensor networks. The key issue is the wakeup scheduling of sensor nodes and minimization of wakeup activity. We propose a lightweight distributed algorithm for coordinating the wakeup scheduling of the sensor nodes. A distinctive feature of this algorithm is that it ensures that the detection delay of any event occurring anywhere in the sensing field is statistically bounded. In addition, the algorithm exposes a convenient interface for users to define the requirement on detection latency, thereby tuning the intrinsic tradeoff between energy efficiency and event detection performance. Extensive simulations have been conducted and results demonstrate that this algorithm can successfully meet delay bound and significantly reduce energy consumption.

1. Introduction

Recent years have witnessed the rapid development of wireless sensor networks. The surge of interest in sensor networks is driven by the promising advantage of sensor network as a low-cost solution to a wide range of real-world challenges [1–6]. Event detection is an important class of applications for sensor networks. The key issues of designing a sensor network for distributed event detection are twofold. First, the system needs to provide quality event detection. That is, the detection of any event that occurs in the physical environment should be as timely as possible. Second, energy efficiency is critical since the battery-powered system is supposed to be continuously functional for months or even years.

Existing work [7–11] for event detection has extensively focused on providing full sensing coverage such that any potential event can be immediately detected after it arises. For energy efficiency, only a fraction of sensors are selected to be active, and the rest are put into sleep mode. The advantage of these algorithms is that no detection latency is incurred. The obvious drawback, however, is poor energy efficiency due to the fact that all active sensors need to be powered up constantly. Moreover, if a sensor fails, the sensing coverage supported by this sensor becomes a blind spot, and consequent critical events occurring at this spot will be lost, which is so-called the blind spot problem.

Most physical events are persistent, rather than ephemeral, which exist for a certain time, such as tens of seconds or even minutes, after its occurrence [12]. Examples for such events include fire, radiation, and pollution. This essential property allows sensors to capture the events while being in a low-duty cycle. A straightforward approach for energy-efficient event detection works as follows, like in [12, 13]. Each sensor sleeps most of the time and wakes up every τ_{cycle} time units, as shown in Figure 1, while in active mode, a sensor detects any potential event that occurs in its vicinity. Let τ_{on} denote the active time in every cycle of τ_{cycle}. The duty cycle of the sensor in Figure 1 is clearly

$$\delta = \frac{\tau_{\text{on}}}{\tau_{\text{cycle}}}. \tag{1}$$

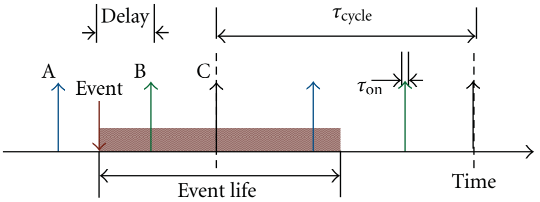


Figure 1: Example timing of three equally cycled sensors.

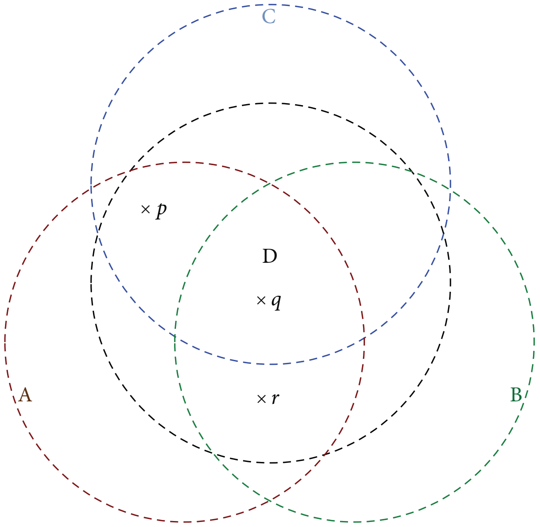


Figure 2: Illustration of the overdetection problem.

This suggests that the lifetime of the sensor node can be roughly extended by a factor of $1/\delta$. Nevertheless, it should be noted that the lifetime extension comes at the cost of additional detection latency. However, we argue that most practical applications tolerate detection latency. Note that it also takes time for the network to report the event back to the base stations through many hop-by-hop transmissions.

Recent study [6, 14] has shown that to guarantee full sensing coverage of the field, the density of sensors needs to be high. This implies that an event can possibly be detected by several ambient sensors. A critical challenge emerges in this situation. The dense deployment can cause a serious *overdetection problem* with Figure 2 when every sensor blindly wakes up in each cycle. As illustrated in Figure 2, an event that occurs at point q will be detected, within a single cycle, four times, and an event at point p will be detected three times since its emergence. Not only does the overdetection problem waste precious energy but also incurs additional energy overhead for event transmissions. Extra efforts are further needed to make a decision on whether or not event reports from different sensors are actually caused by the same physical event. The overdetection problem strongly motivates the idea that the duty cycle of each sensor should be reduced for better energy efficiency.

In this paper, we propose an innovative wakeup scheduling algorithm called PAD for energy-efficient event detection. The central idea is to reduce the duty cycle of every sensor via probabilistic wakeup, exploiting the dense deployment of sensor networks. The wakeup of a sensor is not deterministic, but instead probabilistic and adaptive to its sensing neighbors, therein significantly alleviating the overdetection problem. A distinctive feature of the algorithms is that it allows users to specify the requirement on detection latency and meanwhile ensures that the detection of any event is better than this requirement. The algorithm is truly scalable and power efficient, prolonging the system lifetime significantly. We have made the following contributions in this paper.

(i) By recognizing the essential latency-tolerant property of event detection applications, we investigate the energy-efficient approach for event detection, which addresses the serious overdetection problem.

(ii) We propose a soft bound model for detection delay specification and devise a simple yet effective metric to realize such a statistical soft bound.

(iii) We present insightful analysis on the nonadaptive scheme, in which the sensors wake blindly, and reveal the necessity to make adaptive control on wakeup frequency.

(iv) We propose a lightweight algorithm in which each sensor works adaptively and reduces its power dissipation substantially, hence remarkable prolonging system lifetime.

The rest of the paper is structured as follows. In Section 2, we present the system model, statistical bound modelfor detection latency, and the problem description. In Section 3, we analyze `DoC` and detection delay with a simplified wakeup scheduling and motivate the scheduling algorithm design. Section 4 proposes the wakeup scheduling algorithm and an extension. The performance evaluation is presented in Section 5. Related work is reviewed in Section 6. The paper is concluded in Section 7.

2. System Model and Problem Description

It is intuitive that there is an intrinsic tradeoff between system lifetime and detection latency. Thus, it is unrealistic to minimize detection latency and meanwhile to maximize system lifetime. For real-world surveillance applications, the system should deliver twofold performance. On the one hand, the detection delay of an event should not be arbitrarily large. Instead, it should be constrained to a certain range. On the other hand, the system should operate in a very power-efficient fashion. A longer system lifetime certainly requires the wakeup scheduling to be energy efficient. To extend the network lifetime, it is crucial to reduce the duty cycle of each individual sensor.

In this section, we present the system model, propose a soft bound model for event detection, and give the problem description. In the rest of this paper, we adopt the notations in Table 1.

2.1. Network Model. We consider a square field F with side length L, and the sensors are deployed in the field according

Table 1: Notations and descriptions.

Notation	Description
r	The detection range
δ	The duty cycle of the sensing device
φ	The duty cycle of the transceiver
$F_D(\cdot)$	The cumulative distribution function of random variable D
D_L	The longest delay specified by the user
D_0	The corresponding delay for χ_0
DoC	Detectability in one cycle
χ_P	The DoC of an event at point p
χ_0	The lowest DoC all over the field
γ_Q	The wakeup probability of sensor Q
$\gamma_Q(p)$	The necessary wakeup probability of sensor Q at point p
$S(p)$	The set of sensors covering point p
$U(Q)$	The set of grid points contained by sensor Q

to a 2-dimensional Poisson process with rate n/L^2. We focus on persistent events which exist for a certain time before they disappear. The event life is much longer than the wakeup cycle, and we can safely assume that an event is always detected. Each sensor has the knowledge of its location. A good number of power-efficient algorithms [15, 16] have been proposed for practical localization. Finally, each sensor has a detection range defining a detection disk centered at the sensor. An event is reliably detected by an active sensor if it resides in the detection range of this sensor.

The power consumption of a sensor node is mainly attributed to three units: processor, sensing device, and radio transceiver. Ideally, each unit has separate power control [17]. The duty cycle of the transceiver is subject to the control of communication protocols. We focus on the study of the duty cycling of the sensing devices. The transceiver may have a different duty cycle from the sensing devices. This indeed increases the flexibility for the algorithm to work with different communication protocols.

A sensor node can be attached with multiple sensing devices of different types. In the algorithm design, we assume, for simplification, that a sensor node is equipped with a single sensing device. However, such design can be easily extended to accommodate the situation of multiple sensing devices. In the rest of the paper, we call a sensor node simply a sensor if not confused with the sensing device.

2.2. Soft Bound for Detection Delay. The detection delay of an event is a random variable dependent on the arrival time of the event, the number of sensors covering the event, and the wakeup schedules of these sensors. It is ideal that the system provides a hard bound for detection delay, that is, any detection delay is less than a given value. However, this compels sensors to wake up at least once in every cycle, which will cause the serious aforementioned overdetection problem.

Providing *soft bound* for event detection is also very valuable for users. More specifically, the user specifies a longest delay (D_L) that is characterized by a cumulative distribution function (CDF). For example, it may be desirable for the user that 30% of events are detected within 1 s, 50% are within 2 s and 80% are within 3 s. Note that this longest delay specified by the user is actually a random variable. The system should then ensure that the detection delay (D) at any point is less than D_L.

Definition 1. Random variable D_1 is less than D_2, denoted by $D_1 \ \le D_2$, if the following condition holds:

$$F_{D_1}(d) \ge F_{D_2}(d), \quad d > 0, \tag{2}$$

where D_1 and D_2 share the same domain.

To specify the requirement on detection latency, the users can simply set the CDF of D_L. The objective of the system then becomes to ensure that the detection delay of any event is less than D_L. However, we have to address a new critical issue, that is, how to realize such a soft bound. To address this, we devise a simple yet effective metric, *detectability in one cycle* (DoC).

Definition 2. The DoC of point p (denoted by χ_{p}) is the probability that any event at p is detected, by at least one sensor, within a single wakeup cycle since its occurrence.

In fact, the DoC of point p characterizes the detection delay of any event at p, denoted by D_p. We derive the CDF of D_p which reveals the essential relationship between χ_p and D_p.

Theorem 3. *The CDF of D_p is given by*

$$F_{D_p}(d) = 1 - \left(1 - \chi_p\right)^c \left(1 - \frac{d - c\tau_{cycle}}{\tau_{cycle}}\chi_p\right), \quad \text{where } c = \left\lfloor \frac{d}{\tau_{cycle}} \right\rfloor. \tag{3}$$

Proof. By definition, the CDF of D_p is

$$F_{D_p}(d) = \Pr\left(D_p \le d\right) = 1 - \Pr\left(D_p > d\right). \tag{4}$$

This implies that there is no sensor wakeup in the duration of d since the emergence of the event. There are c full cycles and an additional length of $d - c\tau_{\text{cycle}}$. The probability that no wakeup happens within one single cycle is $1 - \chi_p$, and that within duration of $d - c\tau_{\text{cycle}}$ is $1 - \chi_p(d/(\tau_{\text{cycle}} - c))$. □

2.3. Problem Description. With the introduction of DoC, it becomes possible to realize the soft bound on detection latency. First, we determine such a DoC χ_0 that the corresponding D_0 is less than D_L, that is,

$$D_0 \ \preccurlyeq D_L. \tag{5}$$

Second, we let the DoC of any point within the sensing field meet the following constraint:

$$\chi_p \ge \chi_0, \quad \forall p \in F. \tag{6}$$

It is apparent that a higher DoC at a point implies a shorter latency of event detection at this point. Thus, the derived D_p is less than D_0, that is,

$$D_p \preccurlyeq D_0, \quad \forall p \in F. \tag{7}$$

By combining (5) and (7), we can conclude that

$$D_p \preccurlyeq D_L, \quad \forall p \in F. \tag{8}$$

Thus, by guaranteeing that the DoC of any point is larger than χ_0, we are able to ensure that the detection delay of any event is less than the user's requirement D_L. Note that a more rigid requirement on real-time detection needs a higher χ_0. In the following, we derive the expected value of D_0, which follows a theorem.

Theorem 4. *The expected value of D_0 is*

$$E(D_0) = \frac{(2-\chi_0)\tau_{cycle}}{2\chi_0}. \tag{9}$$

Proof. The expected delay is $\tau_{\text{cycle}}/2$ if the event is detected within the first cycle. If it is detected in the jth cycle, $j > 1$, then additional $(j-1)\tau_{\text{cycle}}$ latency is introduced. Let M denote the number of full cycles that an event undergoes before it is detected. The probability mass function (*PDF*) of M is given by

$$\Pr(M = k) = (1-\chi_0)^{k-1}\chi_0, \quad k \geq 0. \tag{10}$$

We derive the expected delay by conditioning on M,

$$\begin{aligned} E(D_0) &= \sum_{i=0}^{\infty}\left(\frac{\tau_{\text{cycle}}}{2 \times \Pr(M = i)}\right) \\ &= \frac{(2-\chi_0)\tau_{\text{cycle}}}{2\chi_0}. \end{aligned} \tag{11}$$

The expected delay is a function of χ_0 and is inversely proportional to χ_0. □

The goal of the network is to make sure that any event e that occurs in the sensing field is detected by the sensor network with a detection delay, D_e, that is statistically bounded by D_0:

$$D_e \preccurlyeq D_0. \tag{12}$$

At the same time, the sensor network should be as energy efficient as possible.

3. Analysis of DoC and Detection Delay

In this subsection, we are interested in the nonadaptive scheme (NAS) in which the wakeup probability of each sensor is identical, that is, not adaptive to its neighborhood. To guarantee that the DoC of any point is greater than χ_0, NAS simply sets the wakeup probability in each cycle of every sensor to χ_0. The problem is that when the density is high, the actual DoC of a point can be much higher than χ_0, resulting in unnecessary waste of energy.

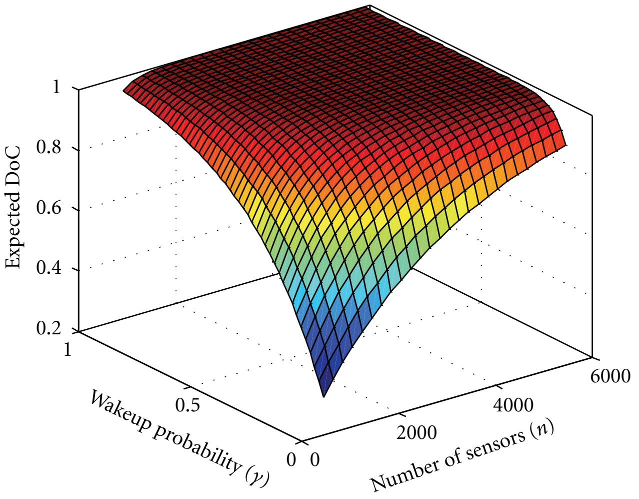


Figure 3: Expected χ_p as a function of n and γ, $L = 300$ m, and $r = 10$ m.

3.1. *DoC Analysis.*

First, we analyze the DoC of any point in the field of NAS and present it in a theorem.

Theorem 5. *With NAS, the expected DoC of any point in the sensing field is*

$$E(\chi_p) = 1 - e^{-\lambda\gamma}. \tag{13}$$

Proof. Let point p be an arbitrary point in the field. Note that we do not consider the special points on the edge. The DoC of p is actually a random variable because it relies on the number of covering sensors and the wakeup probabilities of the sensors covering p. The number of sensors covering point p is a random number, denoted by X_p, which is Poison distributed with the PDF as

$$\Pr(X_p = k) = \frac{1}{k!}\lambda^k e^{-\lambda}, \quad \text{where } \lambda = \frac{n\pi r^2}{L^2}. \tag{14}$$

Let γ denote the identical wakeup probability. By conditioning on X_p, we have

$$E(\chi_p) = \sum_{k=1}^{n}\left(\left(1-(1-\gamma)^k\right) \times \Pr(X_p = k)\right) = 1 - e^{-\lambda\gamma}. \tag{15}$$

□

Figure 3 plots the expected value of χ_p as a function of n and γ. We can find that both increasing density and increasing the wakeup probability can increase the DoC of p. When the density is high, even a relatively low wakeup probability can produce a high DoC close to one. This strongly suggests that the wakeup probability can be reduced so that the DoC is close to χ_0, therefore conserving more energy.

3.2. *Delay Analysis.*

Next, we consider the detection delay achieved by NAS.

Theorem 6. *With NAS, the expected detection delay of an event that happens at any point is given by*

$$E(D) = \left(\frac{1-(1+\lambda\gamma)e^{-\lambda\gamma}}{\lambda\gamma} + \frac{e^{-\lambda\gamma}}{1-e^{-\lambda\gamma}}\right)\tau_{cycle}. \tag{16}$$

Proof. Let Y_p denote the number of sensors being active in a cycle. We derive the PDF of Y_p by conditioning on X_p,

$$\begin{aligned}\Pr\left(Y_p = i\right) &= \sum_{j=0}^{n}\Pr\left(Y_p = i \mid X_p = j\right) \times \Pr\left(X_p = j\right) \\ &= \frac{1}{i!}(\lambda\gamma)^i e^{-\lambda\gamma}.\end{aligned} \tag{17}$$

Interestingly, Y_p turns out to be Poisson distributed with rate $\lambda\gamma$.

According to the analysis in [6], the expected delay of an event is given by $D_c = \tau_{\text{cycle}}/Y_p$, if it is detected in the first cycle. The expectation of D_c is given by

$$E(D_c) = \sum_{k=1}^{n}\left(\frac{\tau_{\text{cycle}}}{k+1} \cdot \Pr\left(Y_p = k\right)\right). \tag{18}$$

Let N denote the number of full cycles that elapsed before an event is detected. The PDF of N is

$$\Pr(N = i) = (1-\theta)^{i-1}\theta, \tag{19}$$

where θ is the probability that an event is detected within one cycle. It is apparent that

$$\theta = 1 - \Pr(Y = 0) = 1 - e^{-\lambda\gamma}. \tag{20}$$

If an event is detected in the ith cycle, an additional latency of $(i-1)\tau_{\text{cycle}}$ is introduced. Thus, we can compute the expectation by conditioning on N,

$$\begin{aligned}E(D) &= \sum_{i=1}^{\infty}\left(\left(E(D_c) + (i-1)\tau_{\text{cycle}}\right) \times \Pr(N = i)\right) \\ &= \left(\frac{1-(1+\lambda\gamma)e^{-\lambda\gamma}}{\lambda\gamma} + \frac{e^{-\lambda\gamma}}{1-e^{-\lambda\gamma}}\right)\tau_{\text{cycle}}.\end{aligned} \tag{21}$$

□

Figure 4 plots the expected delay as a function of wakeup probability. We have two observations. First, increasing wakeup probability produces a decreasing delay expectation, as is obvious in the sense that an event is more likely to be detected in earlier cycles. Second, expected delays of NAS are all much smaller than D_0, especially when the density of sensors is high. This highly suggests that the system should introduce adaptive control over the wakeup probability such that each sensor operates more energy efficiently.

4. Probabilistic Wakeup

The design goals of the sensor system are (1) to extend system lifetime by reducing the duty cycle of every sensor; (2) to ensure that the detection latency of any event is statistically bounded by the requirement posed by the users. As discussed previously, this is to be achieved by ensuring that the DoC of any point is larger than χ_0. PAD adopts a probabilistic approach and solves the overdetection problem by adaptively tuning wakeup frequency, exploiting the natural dense deployment.

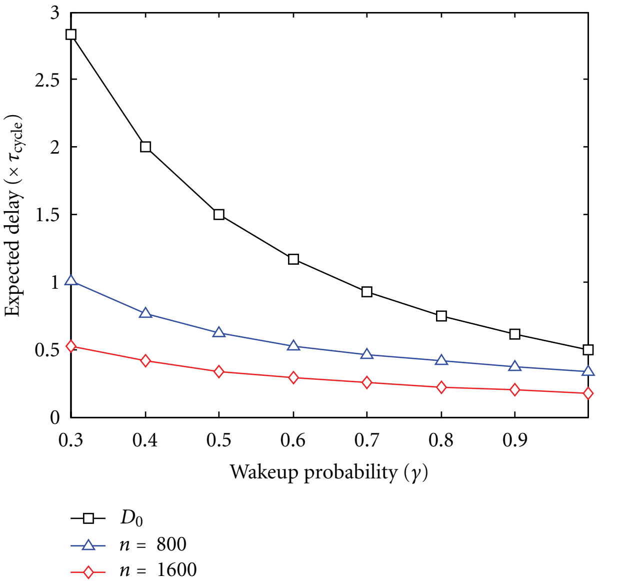


Figure 4: Expected delay as a function of γ, L = 300 m, and r = 10 m.

Following the probabilistic approach, a sensor Q wakes up in each cycle with probability γ_Q and remains in sleep mode with probability $1 - \gamma_Q$. The key issue is clearly the determination of the wakeup probability. The wakeup probability should be as small as possible for the power efficiency purpose. At the same time, however, it ought to be sufficiently large to guarantee the DoC of location points within its ambient neighborhood.

An event can arise anywhere in the sensing field, and it is impossible to predict the arising location of the event. Thus, we need to consider any location point in the field. As there are infinite number of points, we divide the whole sensing field into virtual grids and consider the finite set of grid points. It is obvious that with division of smaller grids, we can have more fine-grained guarantee on detection latency. At the same time, however, a smaller grid size will introduce more grid points, and thus a higher space and time complexity will be incurred on each sensor node.

The state transition diagram of the algorithm is depicted in Figure 5.

4.1. Design of Wakeup Scheduling Algorithm. The algorithm executes in two phases. In the initialization phase, each sensor discovers its neighbors. Based on the neighborhood information, a sensor determines a conservative wakeup probability. This probability is sufficiently large to guarantee the DoC of any point, and hence it results in power inefficiency. In the next phase, in response to energy inefficiency,

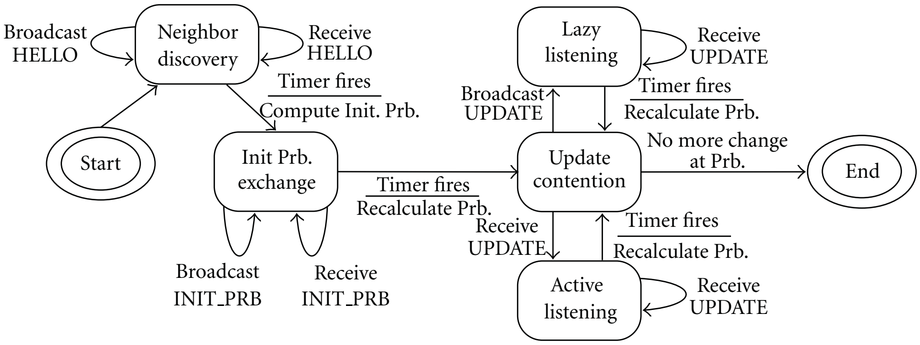


Figure 5: State transition diagram of the PAD algorithm.

an iterative optimization procedure is carried out among the sensors, to further reduce wakeup frequency.

4.1.1. Conservative Initialization. At the beginning, each sensor tries to find its neighbors within $2r$ distance from itself by exchanging HELLO messages with each other. For a given sensor, a neighbor is a *sensing neighbor* (distinguished from a communication neighbor) if its distance to the neighbor is less than $2r$. Every sensor maintains a table for its sensing neighbors. Upon receiving a HELLO, the sensor records the sender in the table if the sender is a sensing neighbor; otherwise, the HELLO is silently dropped.

After the neighbor discovery, the sensors start to compute its initial wakeup probability by executing the *Conservative Wakeup Determination* (CWD) algorithm. The initial wakeup probability guarantees that the DoC of any point is greater than χ_0. Let $S(p)$ denote the set of sensors covering point p. The DoC of point p is

$$\chi_p = 1 - \prod_{B \in S(p)} (1 - \gamma_B). \tag{22}$$

To meet constraint (6), each sensor firstly computes the necessary probability for every grid point (*point level*) within its detection range and then computes the wakeup probability of the sensor (*node level*). In CWD, the sensors covering the same point are supposed to play an equally important role in detecting events at this point. Take sensor Q, for example, and its necessary probability for a point p within its detection range is,

$$\gamma_Q(p) = 1 - \sqrt[k]{1 - \chi_0}, \quad \text{where } k = |S(p)|. \tag{23}$$

To compute its wakeup probability at the node level, Q takes the maximum out of the wakeup probabilities of all grid points within its detection range. Let $U(Q)$ denote the set of all the grid points within the detection range of Q. Then, the node-level wakeup probability of Q is

$$\gamma_Q = \max\{\gamma_Q(p), \forall p \in U(Q)\}. \tag{24}$$

CWD is conservative since each sensor takes the maximum as its wakeup probability. The wakeup probability is sufficiently large for every grid point in its detection range to have a larger DoC than required. The consequence is that the DoC of a point may actually be much larger than the required χ_0. Such conservativeness incurs unnecessary energy consumption.

4.1.2. Optimization. It is imperative to further improve the energy efficiency after the initial selection. Therefore, we propose a *cooperative refinement procedure* (CRP) to refine the wakeup frequency of each sensor node. Following this procedure, each sensor derives a new wakeup probability based on the wakeup probabilities of its sensing neighbors. If the newly computed wakeup probability is smaller, it tries to adjust its wakeup probability, attempting to reduce its wakeup frequency. CRP executes round by round. In each round, a sensor can update its probability at most once. It is guaranteed that CRP terminates in a finite number of rounds.

After determining the initial wakeup probability, sensors exchange their wakeup probabilities by local broadcast. Each sensor recalculates a feasible wakeup probability based on the wakeup probabilities of its sensing neighbors. Similar to CWD, a sensor firstly computes a new wakeup probability for each grid point. The new feasible wakeup probability for point p is given by

$$\gamma_Q{}^{(k+1)}(p) = \begin{cases} 0, & \text{if } \prod\limits_{B \in S(p) - \{Q\}} (1 - \gamma_B) > 1 - \chi_0 \\ 1 - \dfrac{1 - \chi_0}{\prod\limits_{B \in S(p) - \{Q\}} \left(1 - \gamma_B{}^{(k)}\right)}, & \text{otherwise}, \end{cases} \tag{25}$$

where (k) denotes the number of generation to which the corresponding wakeup probability belongs.

To compute the new wakeup probability at the node level, Q also takes the maximal probability among those of all the grid points within its detection range,

$$\gamma_Q{}^{(k+1)} = \max\left\{\gamma_Q{}^{(k+1)}(p),\ p \in U(Q)\right\}. \tag{26}$$

If the new probability is smaller than the original one, the sensor will update the probability to the new one for the energy efficiency purpose. Thus, any sensor that obtains a smaller new probability makes an update attempt, trying to reduce its probability.

Due to the computation dependence, it is critical to avoid parallel updates. CRP requires that before a sensor can actually update its wakeup probability, it must broadcast the new probability to its sensing neighbors and suppress them from updating simultaneously. An UPDATE message is used to enclose the ID and the new probability of a sensor. Before an UPDATE is broadcast, the sensor undergoes a random backoff to minimize UPDATE transmission collisions. If a sensor receives an UPDATE from its sensing neighbor before its own UPDATE is broadcast, it suppresses its planned UPDATE broadcast and cancels its own update attempt (if any). However, if a sensor successfully broadcasts its UPDATE, it commits the update attempt, updating its wakeup probability.

After successfully broadcasting an UPDATE, in theory, a sensor would not receive any UPDATE from its sensing neighbors. However, unreliable wireless transmissions make it still possible that the sensor receives some. In CRP, a sensor that has successfully committed its update stays in the *lazy* state, where it ignores any UPDATE following its broadcast. For those sensors that cancelled its update attempt, they actively listen, receive all the UPDATE from its sensing neighbors, and update the corresponding wakeup probability in the locally maintained table.

4.2. Extension for Differentiation. It is sometimes necessary for some areas to be more carefully monitored, necessitating detection differentiation. The differentiation can be in either *detection latency* or *detection degree*. When the differentiation is in detection degree, we modify the definition of DoC to accommodate the need of higher degrees. Recall that the DoC of a point is the probability that an event is detected, by at least one sensor, within one cycle. If a point requires a degree of two, we define the *quadratic* DoC of a point as the probability that an event at this point is detected, by at least *two sensors*, within one cycle. A higher degree provides more robust event detection against sensor failure.

(i) To have a shorter latency for a specific point q, we can easily set a larger DoC for point q, for example, $\chi_0(q)$. Then all sensors covering q should replace χ_0 with $\chi_0(q)$ in all previous computations.

(ii) To have a higher degree, more sophisticate modifications over PAD are necessary, which are elaborated as follows.

Suppose the degree for grid point q is two, rather than one. In the process of computing the initial probability, likewise, the sensors covering q should play an equal role, therefore having an identical probability, denoted by γ. For differentiation, we term the DoC at q as quadratic DoC, denoted by $\widehat{\chi}_q$

$$\widehat{\chi}_q = 1 - (1-\gamma)^k - k(1-\gamma)^{k-1}\gamma, \quad \text{where } k = |S(q)|. \tag{27}$$

Each sensor covering q needs to compute the necessary initial probability for q. It is difficult to compute the exact root of (27) since it needs to solve a high-dimensional equation. Nevertheless, the quadratic DoC monotonously increases with increasing γ. Thus, we can develop a numerical procedure to find a desirable γ that is close to the real root.

In the refinement procedure, a sensor adjusts its wakeup probability based on the wakeup probabilities of its sensing neighbors. When the detection degree is two, the formula (25) should be reformulated as follows:

$$\gamma_Q{}^{(k+1)}(q) = \begin{cases} 1 - \dfrac{1-\chi_0-v}{v\omega}, & \text{if } 1-\chi_0-v < v\omega, \\ 0, & \text{otherwise,} \end{cases}$$
$$\text{where } v = \prod_{B\in S(q)-\{Q\}} \left(1-\gamma_B{}^{(k)}\right), \tag{28}$$
$$\omega = \sum_{B\in S(q)-\{Q\}} \frac{\gamma_B{}^{(k)}}{1-\gamma_B{}^{(k)}}.$$

5. Evaluation

In this section, we first introduce the experiment methodology and simulation setting. Next, we discuss the evaluation results.

5.1. Methodology and Simulation Setup. To evaluate the performance of PAD, we conduct extensive simulation experiments with a simulator developed for simulating a low-duty-cycled sensor network. We adopt the data of the eXtreme Scale Mote [12]. The setting of the key simulation parameters is shown in Table 2, if not specified elsewhere. A sensor is usually powered by two AA batteries, which can typically provide about 2×10^4 J energy. In our simulations, however, the initial energy for every sensor is set to 50 J to reduce lengthy simulations. The results presented in this section are averaged over 20 independent experiments with different sensor deployments.

We compare the performance of PAD with NAS and the upper theoretical bound in terms of system lifetime extension. We define the *hard lifetime* as the time from the starting time to the time instant when the DoC of any point within the field drops below χ_0. We define *α-lifetime* as the time until less than α% area of the field can meet the DoC requirement. The hard lifetime is highly subject to the influence of irregular deployment. To address this issue and study the energy conservation ability of PAD, soft lifetime is more suitable, which is less sensitive to sensor deployment.

Table 2: Simulation setting.

Parameter	Value	Parameter	Value
R	30 m	L	300 m
r	10 m	ξ	50 J
ρ_S	19.4 mW	τ_{cycle}	10 s
ρ_P	20 mW	τ_{on}	0.5 s
ρ_R	24 mW	n	2700
φ	0.01	χ_0	0.8

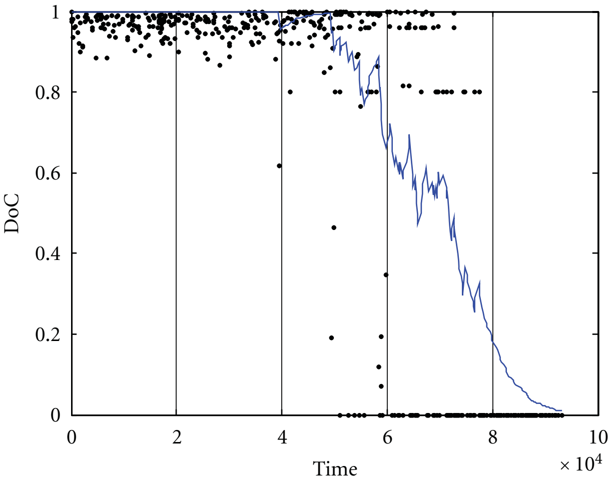


Figure 6: DoC over time.

It is difficult to derive the tight bound of the hard system lifetime. We give an optimistic upper bound of the lifetime. Let ρ_P, ρ_S, and ρ_R denote the power rates of the processor, the sensing device, and the transceiver, respectively. A point in the field is covered by λ sensors. Ideally, these sensors share the same wakeup probability, which is $1-(1-\chi_0)1/\lambda$. Thus, the actual power consumption rate of the sensor unit is $\rho_s\tau_{\text{on}}(1-\sqrt[\lambda]{1-\chi_0})/\tau_{\text{cycle}}$. The upper bound of the hard lifetime can be computed accordingly,

$$\mathcal{T}_{ub}=\frac{\xi}{\rho_P\varphi+\rho_S\delta'+\rho_R(\varphi+\delta')},$$
$$\text{where } \delta'=\frac{\tau_{\text{on}}}{\tau_{\text{cycle}}}\left(1-\sqrt[\lambda]{1-\chi_0}\right). \tag{29}$$

5.2. Typical Run. The first set of experiments investigates the performance of PAD in a typical run. Figure 6 reports DoCs of events over time. Each point in the figure represents the DoC of a random point. The exponentially weighted moving average of DoC is also shown using a solid curve. We can see that PAD successfully guarantees that the DoC of any point is larger than 0.8 before 4 × 10^4 s. After that time, some region becomes uncovered because of sensor depletion. Beyond the time 4 × 10^4 s, the DoC in some area falls below 0.8. This results in the factor that different areas in the sensing field have different numbers of covering sensors.

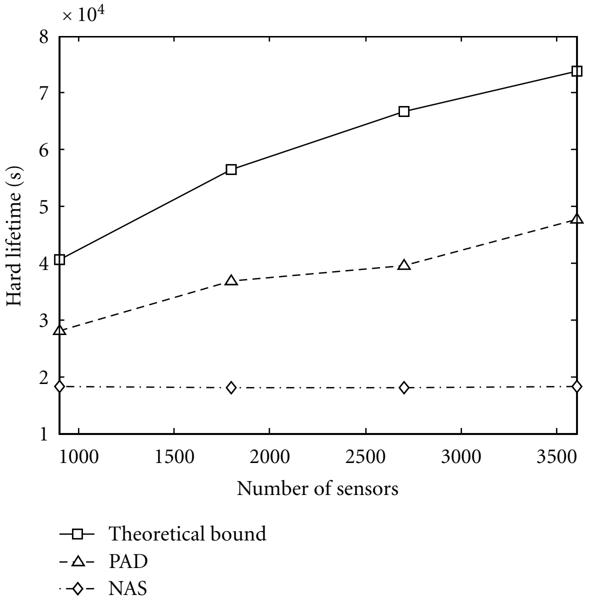


Figure 7: Hard lifetime versus number of sensors.

5.3. Lifetime Extension. The second set of experiments investigates lifetime extension of the algorithm, in comparison to NAS and the theoretical bound. We vary the number of sensors to study lifetime extension under different density configurations. As we can see in Figure 7, the hard lifetime of the upper bound increases proportionally with the increasing number of sensors. We can see that PAD extends the hard lifetime remarkably, compared with NAS. With the increasing number of sensors, the lifetime extension becomes more significant. This demonstrates that PAD can effectively exploit high sensor density. NAS fails to extend system lifetime even if the sensor density becomes higher. It is because with NAS every sensor wakes up blindly with probability χ_0 in each cycle. In Figure 8, we show the soft lifetime of PAD and NAS. As we can see, the soft lifetime significantly increases with the increasing number of sensors. For NAS, however, the soft lifetimes for different α remain the same. In addition, the increasing density does not lead to a longer system lifetime.

5.4. Detection Delay. In the third set of experiments, we explore detection delay for different schemes. Figure 9 reports the average detection delay under different sensor densities. We can see that the average delay achieved by PAD is below the bound but larger than that of NAS. PAD effectively reduces wakeup probability of the sensors and consequently increases detection delay. In contrast, in NAS, each sensor has the same wakeup probability. A point is covered by more sensors when the density increases. This suggests that detection delay decreases. It is worth noting that with increasing density, PAD's detection delay decreases, but the decrease is much slower than NAS.

5.5. Algorithm Convergence. In this set of experiments, we explore the convergence by investigating the number of

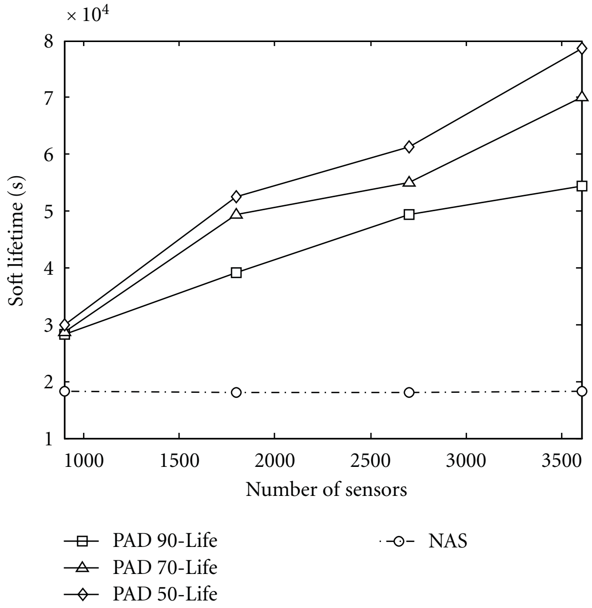


Figure 8: Soft lifetime versus number of sensors.

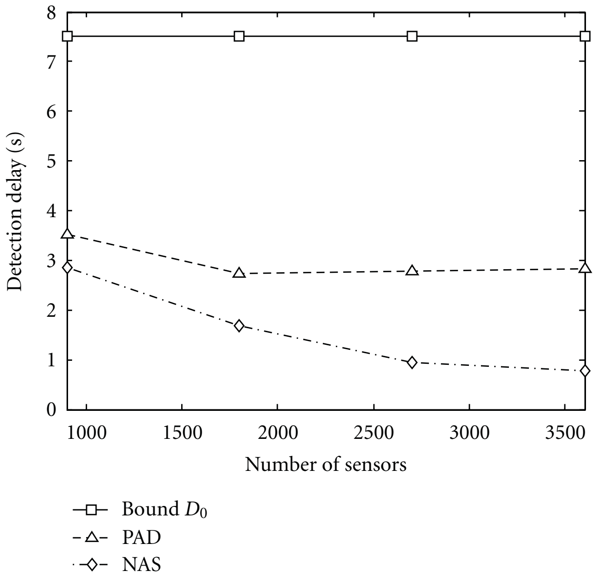


Figure 9: Detection delay versus number of sensors.

requesting sensors (the sensors that try to update wakeup frequency) in each round over time. The PAD algorithm converges when there are no more requesting sensors. Figure 10 illustrates the number of requesting sensors over time. We can see that the number of requesting sensors in a round decreases as time elapses. There is no more requesting sensor, and the refinement procedure terminates when the time reaches 28 s. This shows that the refinement procedure of PAD is able to quickly converge and starts functioning soon after the deployment of the sensor network.

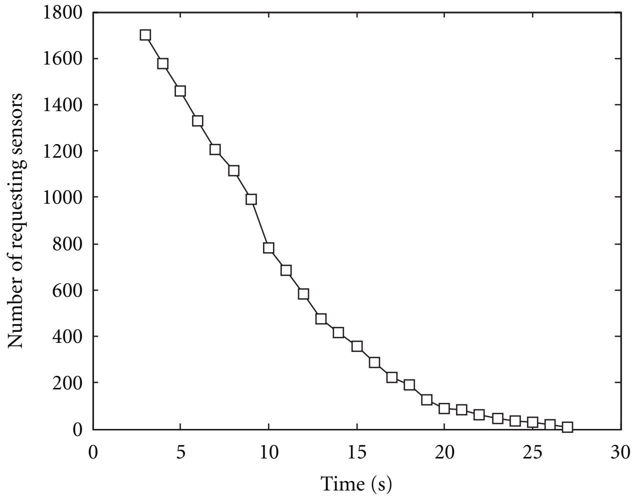


Figure 10: Number of requesting sensors over time.

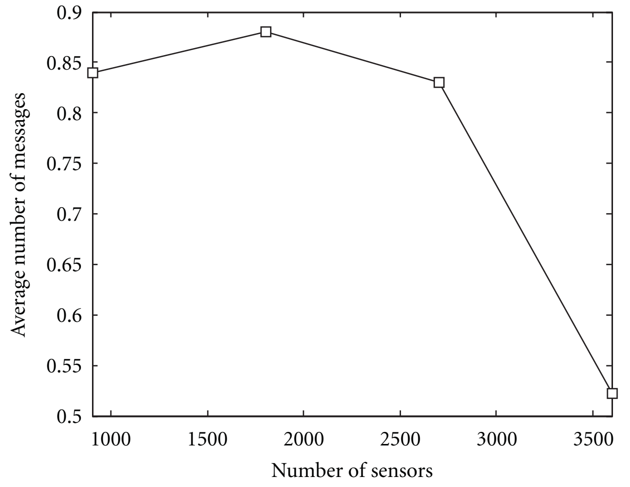


Figure 11: Number of messages per sensor versus number of sensors.

5.6. Overhead. We also study communication overhead introduced by the PAD algorithm. Figure 11 shows the number of algorithm messages per sensor under different configurations with varying number of deployed sensor nodes. We can see that the average number is below 0.9 messages per sensor. More interestingly, the number of algorithm messages per sensor is decreasing when the number of sensors is increasing. This shows that increasing density does not incur higher overhead per sensor. Such communication complexity is affordable for sensors, and hence PAD is scalable with respect to sensor density.

6. Related Work

It has been an effective approach for conserving power of sensor nodes with duty cycling [6, 12, 13, 18–22]. Many methods have been proposed in the literature. Armstrong made a good survey on energy-conserving methodologies

[23]. A lot of power-scheduling algorithms [18] are aimed to extend network lifetime via scheduling sleep/active states of sensors. Asynchronous wakeup scheduling [10] is superior since it is not dependent on time synchronization. However, it introduces additional packet delivery delay. On the MAC layer, the low-duty cycling of the radio transceiver can be employed to reduce energy consumption of the senor node [24–26].

For object tracking applications, when the sensor nodes are duty cycled, energy-quality tradeoffs are intrinsic [2, 4]. Probabilistic coverage in sensor networks has been studied in the context of object tracking [27]. In [28], the authors deployed a test of 70 sensors to track positions of mobile vehicles. In the testbed, only 5% of sensor nodes were kept active, and the rest of the senor nodes operated at a very low-duty cycle (4%). Under this configuration, the network was still capable for tracking vehicles.

Maintenance of full sensing coverage has been of significant importance for many sensor network applications. Several algorithms have been proposed to select a small subset of the sensor nodes to stay active to maintain full coverage and turn off the rest sensor nodes for energy conservation. PEAS [10] makes use of a heuristic that when a sensor is active its neighborhood sensors can go to sleep. Each sensor periodically sends probe signals based on which neighbor sensors can decide to sleep or not. Yan et al. [9] identifies a redundant sensor node whose sensing coverage is jointly covered by its active neighbors. In [29], random and coordinated algorithms have been studied for maintaining the network coverage of a sensor network in which sensor nodes are low duty cycled.

Gupta et al. [30] proposed a randomized algorithm to determine an active schedule of the sensors. At any time, the set of currently active sensors guarantee to provide full sensing coverage. Some other effort [8] considers both sensing coverage and network connectivity. These algorithms provide full sensing coverage and meanwhile maintain network connectivity.

Shakkottai et al. [31] studied the coverage of a sensor network where the sensor nodes are not reliable. In [30], the sensors were divided into several groups and algorithms were developed to maximize the sum of sensing coverage. Event detection using low-duty-cycled sensors has also been discussed [12, 13]. In [13], the authors propose a two-stage optimization algorithm to minimize detection latency. A node platform eXtreme Scale Mote [12] was designed for long-lived operations detecting ephemeral events.

Some existing works focus on detecting complex events [3, 32, 33]. In these works, an event may span a certain region. The occurrence of an event must meet a certain requirement on its boundary. Thus, contour mapping becomes an important operation for event detection in sensor networks. A few effective distributed algorithms [3, 32, 33] have been proposed for event detection based on contour mapping.

In this work we focus on providing soft detection bound for event detection in sensor network. The unique contribution of this work is twofolds. First, we propose a novel soft bound model for delay tolerance specification by users or applications. Second, we propose a distributed wakeup scheduling algorithm that ensures the detection delay of any event in the sensing field to be statistically bounded. Thus, this work is complementary to existing works for event detection. Some preliminary results of this work have been published in [22].

7. Conclusion

In this paper, we have investigated the probabilistic approach to distributed event detection in sensor networks. We empower the users to define the requirement on desirable detection latency of event detection. The system guarantees that the detection latency of any event is statistically bounded by the latency requirement by the users. The probabilistic paradigm allows each sensor to tune its wakeup frequency and hence minimize its power dissipation. It also finely solves the overdetection problem. The developed algorithm is completely distributed, being scalable up with increasing network scale and sensor deployment density. In addition, it supports fine-grained differentiation of event detection throughout the sensing field. Comprehensive simulation experiments demonstrate that the algorithm remarkably prolongs the functional network lifetime and introduces minimal communication overhead.

Acknowledgments

This research is supported by Shanghai Pu Jiang Talents Program (10PJ1405800), NSFC (no. 61170238, 60903190, 61027009, 60970106, 60673166, and 60803124), Shanghai Chen Guang Program (10CG11), 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01 and 2009ZX03006-004), Doctoral Fund of Ministry of Education of China (20100073120021), and 863 Program (2009AA012201 and 2011AA010500). In addition, it is partially supported by the Open Fund of the State Key Laboratory of Software De-velopment Environment (Grant no. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

References

[1] A. Arora, P. Dutta, S. Bapat et al., "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[2] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 129–143, October 2004.

[3] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, article 10, 2009.

[4] S. Pattem, S. Poduri, and B. Krishnamachari, "Energy-quality tradeoffs for target tracking in wireless sensor networks," in *Proceedings of the 2nd International Workshop Information Processing in Sensor Networks (IPSN '03)*, vol. 2634 of *Lecture*

Notes in Computer Science, pp. 32–46, Palo Alto, Calif, USA, 2003.

[5] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and yield in a volcano monitoring sensor network,” in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI ’06)*, pp. 381–396, 2006.

[6] Y. Zhu, Y. Liu, L. M. Ni, and Z. Zhang, “Low-power distributed event detection in wireless sensor networks,” in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM ’07)*, pp. 2401–2405, May 2007.

[7] D. Tian and N. D. Georganas, “A node scheduling scheme for energy conservation in large wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2003.

[8] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys ’03)*, pp. 28–39, November 2003.

[9] T. Yan, T. He, and J. A. Stankovic, “Differentiated surveillance for sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys ’03)*, pp. 51–62, November 2003.

[10] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “PEAS: a robust energy conserving protocol for long-lived sensor networks,” in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems (ICDCS ’03)*, pp. 28–37, May 2003.

[11] Y. Zou and K. Chakrabarty, “A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 978–991, 2005.

[12] P. Dutta, M. Grimmer, A. Arora, S. Bibykt, and D. Culler, “Design of a wireless sensor network platform for detecting rare, random, and ephemeral events,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 497–502, April 2005.

[13] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, “Towards optimal sleep scheduling in sensor networks for rare-event detection,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 20–27, April 2005.

[14] S. Kumar, T. H. Lai, and J. Balogh, “On k-coverage in a mostly sleeping sensor network,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom ’04)*, pp. 144–158, October 2004.

[15] M. Li and Y. Liu, “Rendered path: range-free localization in anisotropic sensor networks with holes,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[16] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O’Dea, “Relative location estimation in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[17] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04)*, pp. 188–200, November 2004.

[18] C. F. Chiasserini and R. R. Rao, “A distributed power management policy for wireless ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1209–1213, September 2000.

[19] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, “Estimating clock uncertainty for efficient duty-cycling in sensor networks,” in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys ’05)*, pp. 130–141, San Diego, Calif, USA, 2005.

[20] Y. Gu, T. He, M. Lin, and J. Xu, “Spatiotemporal delay control for low-duty-cycle sensor networks,” in *Proceedings of the Real-Time Systems Symposium (RTSS ’09)*, pp. 127–137, December 2009.

[21] Y. Zhu and L. M. Ni, “Probabilistic approach to provisioning guaranteed QoS for distributed event detection,” in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08)*, pp. 1265–1273, April 2008.

[22] Y. Zhu and L. M. Ni, “Probabilistic wakeup: adaptive duty cycling for energy-efficient event detection,” in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM ’07)*, pp. 360–367, October 2007.

[23] T. Armstrong, “Wake-up based power management in multi-hop wireless networks,” Tech. Rep., University of Toronto, 2005.

[24] I. Rhee, A. Warrier, M. Aia, and J. Min, “Z-MAC: a hybrid MAC for wireless sensor networks,” in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys ’05)*, pp. 90–101, San Diego, Calif, USA, 2005.

[25] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’02)*, pp. 1567–1576, June 2002.

[26] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle MAC with scheduled channel polling,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys ’06)*, pp. 321–334, November 2006.

[27] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, “Probabilistic coverage for object tracking in sensor networks,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM ’04)*, Philadelphia, Pa, USA, October 2004.

[28] T. He, S. Krishnamurthy, J. A. Stankovic et al., “Energy-efficient surveillance system using wireless sensor networks,” in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys ’04)*, pp. 270–283, 2004.

[29] C. F. Hsin and M. Liu, “Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN ’04)*, pp. 433–442, April 2004.

[30] H. Gupta, S. R. Das, and Q. Gu, “Connected sensor cover: self-organization of sensor networks for efficient query execution,” in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC ’03)*, pp. 189–200, June 2003.

[31] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: coverage, connectivity and diameter,” in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1073–1083, April 2003.

[32] M. Li and Y. Liu, “Iso-Map: energy-efficient contour mapping in wireless sensor networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 699–710, 2010.

[33] M. Li, Y. Liu, and L. Chen, “Nonthreshold-based event detection for 3D environment monitoring in sensor networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 12, pp. 1699–1711, 2008.