Sink Mobility Strategies for Reliable Data Collection in Wireless Sensor Networks

Yuki Fujita*, Daichi Kominami† and Masayuki Murata*
*Graduate School of Information Science and Technology, Osaka University, Japan
{y-fujita, murata}@ist.osaka-u.ac.jp
†Graduate School of Economics, Osaka University, Japan
d-kominami@econ.osaka-u.ac.jp

***Abstract*—Internet of Things (IoT) and machine-to-machine (M2M) will take root throughout our life in the near future. Therefore more and more reliability is required in many wireless sensor network applications, such as the intruder detection and searching system for main rescuers with some sensor devices. However, without the assumption that all sensor nodes are reachable to one of sink nodes through multi-hop communication and the connectivity among the sensor nodes are stable, it is difficult to guarantee the reliability of data collection. In this paper, we focus on controlling the mobility of the mobile sink and propose two types of mobility strategies to collect the sensing data of all sensor nodes in the observed area certainly. The first strategy is learning the observed area and the other is collecting the sensing data using the learned information. Through computer simulations, we show that the mobile sink with the mobility strategies in our proposal can collect the sensing data of all sensor nodes.**

Keywords–Wireless sensor networks, Reliable data collection.

I. INTRODUCTION

Supporting assured data collection in wireless sensor networks (WSNs) is one of significant challenges in frequently changing environments. This is because that dynamic changes in the observed area or loss of connectivity occur in various situations, which cannot be dealt with conventional transport techniques. This promotes network-level reliable mechanisms for data collection. We focus on the mobility control of a data collecting node usually called *sink node* for realizing reliable data collection.

Wireless sensor networks, which facilitate to collect environmental information, are significantly expected to apply to various applications, e.g., monitoring of temperature and humidity in a farm, tracking of animals and etc. This feature greatly attracts attention of many researchers [1] [2]. In many cases, WSNs are composed of many sensor nodes and a few sink nodes which operate in a distributed manner and are connected to each other by wireless communications. Sensor nodes forward their sensing data to one of sink nodes through multi-hop communications, which makes it possible to collect environmental information in the location where one cannot get into.

This ideal scenario is satisfied under the assumption that all sensor nodes are reachable to one of sink nodes through multi-hop communication. However, this assumption is not always realistic due to the limitation of communication range of nodes. Since power saving of sensor nodes with a limited battery capacity is required in WSNs, it is inappropriate to expand communication range with much more transmission power. Also, it is difficult to deploy or add sensor nodes over the network paying excess attention with reachability and connectivity.

In the near future, a lot of machines will be connected mutually and will be quietly embedded in our life space. Thus, internet of things (IoT) and machine-to-machine (M2M) will make WSN systems brings to various applications. In the applications strongly tied to safety and security, it is one of the most important viewpoints with the reliability of information gathering. For example, an intruder detection system with surveillance cameras and some types of sensor devices, and a rescue system in disaster areas with autonomous robots and sensor devices demand more and more reliable data collection than conventional applications in WSNs do. In particularly, in recent years, synergy between sensor networks and autonomous robots is attracting a lot of attentions [3]. Our proposal is also what considers such synergy as discussed below.

We focus on a sink node with mobility called mobile sink. A mobile sink can achieve both reduction of power consumption and data reachability by approaching toward each sensor node and receiving data and carrying it to the static base station. Many studies have been conducted about mobile sinks as a solution for power saving, which is one of a challenging problems in WSNs. Most of those studies target at power-saving applications of the mobile sink, such as the optimal path planning with the optimal routing technique supposing knowledge of exact positions and residual powers of all sensor nodes [4]–[6], and use of the mobile sink in the viewpoint of reliable data gathering does not have many active researches. Here, we define *a network* as sets of sensor nodes reachable to each other by multi-hop communication and we take account of following two types of changes of a network that are caused by failures, energy depletion, or additions of sensor nodes.

- Changes inside a network, which do not increase or decrease the number of networks, such as link failures and node failures not losing reachability, or changes in routes
- Changes outside a network, which increase or decrease the number of networks, such as disconnection of a network, joint of networks, or deployment of a new network

Changes inside a network have been well-studied, however, changes outside a network have not been considered in existing studies. Thus, our interests are on how can we collect all data in observed area when both types of changes occur, in other words, realizing *the reliable data collection*. We aim at reliable data collection with a mobile sink within a pre-defined observed area where both the number and the places of sensor nodes are unknown. Under this circumstances, it is a realistic method for a mobile sink to travel over the whole observed area since it is unknown how many networks are there in the observed area, but it takes much more time to travel every nook and cranny as the area gets larger. However, a few cycles of traveling throughout the area does not suffice for assured data collection under various situations.

Controlled mobility is a key idea for maintaining network connectivity and data reachability, where mobility of mobile sinks is controlled from both inside and outside of networks dynamically [7] [8]. We previously combined CM with the proactive routing mechanism in a wireless sensor network, where periodically exchanged route information messages lead a mobile sink toward a data collecting node [9]. In this paper, we propose two mobility strategies for mobile sink in order to manage those two changes using controlled mobility. The first strategy is to grasp the all positions of networks and impassable locations, and the other is to collect sensing data in each network. These two strategies are followed by either mobile sinks or other patrolling robots, and we assume the former hereafter. Therefore, a mobile sink in our proposal has two mobility phases.

Phase 1: **Mobility for learning the observed area**
A mobile sink sweeps across the observed area and learns the positions of networks and areas where mobile sinks can move in

Phase 2: **Mobility for collecting sensing data**
A mobile sink visits all networks using learned information in Phase 1. Moreover, each network leads a mobile sink to reach data possessing nodes with exploiting transmitted route-information messages when the mobile sink enters the network

In principle, it is difficult to catch an unexpected change by methods other than Phase 1 while it requires a lot of time. Therefore, Phase 1 is taken repeatedly for every fixed cycle. In Phase 2, we use a clustering technique and gather all data in a network to one or more cluster heads since visiting each node to collects data takes a considerable amount of time for a mobile sink. Moreover, cluster heads switch their role back to a non cluster-head state periodically for managing changes inside a network and for achieving load balancing. Combining of the mobility strategies with Phase 1 and Phase 2 can assure to collect all sensing data within the observed area.

The rest of this paper is organized as follows. In Section II, we present the mobility control strategy for memorizing locations of networks. In Section III, we show the mobility strategy for collecting sensing data in a network. Section IV presents simulation results and finally, we conclude our paper in Section V.

II. PHASE 1: LOCATION MEMORIZATION OF NETWORKS

A mobile sink has to periodically check the entire picture of the observed area, such as the positions of networks and a forbidding places, to determine the path for collecting data in Phase 1. In order to grasp these information, it moves in every corner of the observed area while identifying and memorizing all the different networks. In Phase 2 the information is also utilized by the mobile sink to visit all memorized networks in the order that it memorized.

In our proposal, a mobile sink moves so that it does not overlook even one sensor node placed in the observed area. To begin with, a mobile sink commences to move from a given initial position, which is one of vertices of the pre-defined square region including the whole observed area as illustrated in Fig. 1. Next, the mobile sink goes straight on toward one of nearby vertices until it reaches $\frac{d}{2}$ length short of the vertex, where d is the wireless communication range of the mobile sink and sensor nodes. Then, it takes a right-angled turn toward the other vertex, heads $\frac{d}{2}$, and again turns in the same direction. The mobile sink repeats to the same process until reaching all vertices, then it returns to the initial position.

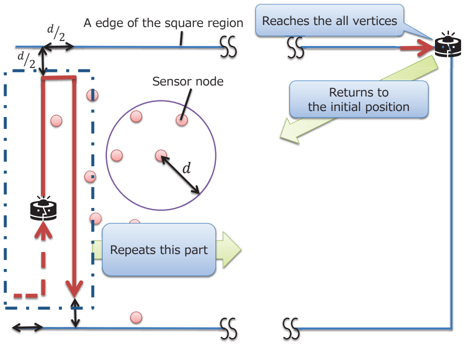

Figure 1. Mobility strategy for detecting all sensor node without any oversight

Algorithm 1 Memorizing the positions of networks associating with $netID$ by the mobile sink

```
// The mobile sink moves in every corner of the observed area.
repeat
    if intercepts PInforMsg($i, netID_i, pList_i, vList_i$) then
        if NetTable has no $entry$ with $netID_i$ then
            register NetTable($netID_i, pos$,PInforMsg.$rssi$)
        else
            if PInfoMsg.$rssi > entry.rssi$ then
                update NetTable($netID_i, pos$,PInforMsg.$rssi$)
            end if
        end if
        if $State_i$ is CLUSTER($i, 0$) then
            sends SensingDataRequest to $S_i$
        end if
    end if
until reaches the end of the observed area
```

In Phase 1, a mobile sink intercepts a message **PInfoMsg**, which all sensor nodes exchange with each other for updating routes (described in Section III-C in detail) and acquires a special identifier (ID) contained in the **PInfoMsg**. This ID is used to identify all networks and remember their positions. The mobile sink updates positions of networks according to Algorithm 1, where some terms are listed in Table I. A table **NetTable** for storing network positions is updated every after receiving a **PInfoMsg**. A **NetTable**'s entry is the tuple ($netID$, $postion$, RSSI) where $netID$ is an identifier of a network, $position$ is the position where mobile sink received **PInfoMsg** and updated the entry, and RSSI is the received signal strength indication of the **PInfoMsg**. An entry is always added to the table if there is no entry whose $netID$ equals to one in the entry, and an existing entry is updated iff the RSSI of a received **PInfoMsg** is greater than existing one with the same $netID$.

This is for ensuring that the mobile sink can obtain more accurate positions of networks. Note that the mobile sink demands sensing data to nodes by transmitting a message **SensingDataRequest** when it contacts with sensor nodes in Phase 1.

TABLE I. NOTATIONS IN OUR PROPOSED METHODS

Notation	Meaning
N	The number of sensor nodes which is placed initially
B	The number of sensor nodes which is planned to fail
A	The number of sensor nodes which is planned to be added
i	The identifier of sensor nodes.
S_i	The sensor node whose ID is i
$ND(S_i)$	The number of neighbor nodes of node S_i
$State_i$	The state of S_i which indicates whether S_i belongs any cluster or not. $State_i$ must be given either **UNCLUSTER** or **CLUSTER**(i, n).
t_{S_i}	The current time of node S_i
T_{limit}	The time limit for searching its neighbor nodes.
T_{S_i}	Delay time of node S_i.
T_{flood}	The period when cluster heads broadcast **PInfoMsg**.
$T_{break}(i)$	The time when S_i breaks.
$T_{add}(i)$	The time when S_i is added in the observed area.
T_h	The period when **NTable** can store a $entry$ non-updated.
$netID_i$	The ID of the network S_i belongs to.
pID	The ID of the potential field.
myP_i	The scalar value which S_i has.
$pList_i$	The set of pID
$vList_i$	The set of myP_i
LI	The interval of learning all the networks in the observed area
seq_c	A sequence number of the sensing data

Algorithm 2 Selection of cluster heads in a network

```
 1: // all nodes perform following;
 2: $State_i \Leftarrow$ UNCLUSTER
 3: $t_{S_i} \Leftarrow$ random value between 0 and $T_{limit}$
 4: broadcast UpdatePacket at $t_{S_i}$
 5: repeat
 6:     if receives a UpdatePacket then
 7:         $ND(S_i) \Leftarrow ND(S_i) + 1$
 8:     end if
 9: until $t_{S_i} \geq T_{limit}$
10: broadcast DegreePacket at $t_{S_i}$, $(T_{limit} \leq t_{S_i} < T_{limit} + T_{S_i})$
11: repeat
12:     if receive StatePacket with CLUSTER$(s, n)$ node then
13:         if $State_i$ is UNCLUSTER then
14:             $State_i \Leftarrow$ CLUSTER$(s, n + 1)$
15:         else if $State_i$ is CLUSTER$(i, m)$ and $m > n + 1$ then
16:             $State_i \Leftarrow$ CLUSTER$(s, n + 1)$
17:         end if
18:         if n+1 $\leq max\_n$ then
19:             broadcast StatePacket
20:         end if
21:     end if
22: until $t_{S_i} \geq T_{limit} + T_{S_i}$
23: if $ND(S_i)$ is larger than all neighbor nodes and $State_i$is UNCLUSTER then
24:     $State_i \Leftarrow$ CLUSTER$(i, 0)$
25:     broadcast StatePacket
26: end if
```

III. PHASE 2: DATA COLLECTION INSIDE A NETWORK

A. Overview of the mobility strategy inside a network

In our proposal, all networks within the observed area need one or more special nodes to gather and store sensing data of all sensor nodes in an individual network, and a mobile sink moves and sojourns at them to bring the sensing data to the base station. We elect this special node using a cluster head election algorithm. The mobile sink intercepts exchanged routing messages among nodes and interpret them to approach all cluster heads in a network.

B. Cluster heads election

We select one or more cluster heads for each networks using a part of DEECIC algorithm [10] with minor modification. The cluster head election algorithm in DEECIC is described in Algorithm 2 with some terms which are tabulated in Table I. As shown in line 4, a sensor node S_i broadcasts an **UpdatePacket** to notify its neighbors of its presence at randomly chosen time t $(0 < t < T_{max})$. Then, S_i broadcasts a **DegreePacket** including $ND(S_i)$, which is the number of received **UpdatePackets** until T_{limit}, to inform its degree at t $(T_{limit} \leq t < T_{limit} + T_{S_i})$. T_{S_i} is given as $T_{S_i} = \alpha e^{1/ND(S_i)}$ where α is a constant to ensure $0 < T_{S_i} \ll T_{limit}$. In lines 11–22, sensor node S_i waits for a **StatePacket** including a state of a neighbor node, which means that whether the neighbor belongs to any cluster or not. Here, CLUSTER(i, n) presents that the node i can reach a certain cluster head by n hops and CLUSTER$(i, 0)$ means that node i is a cluster head. Upon receiving a **StatePacket** with **CLUSTER**(s, n), S_i sets $State_i$ to **CLUSTER**$(i, n{+}1)$ if $State_i$ is **UNCLUSTERD** or $State_i$ is **CLUSTER**(i, m) with $m > n + 1$. S_i broadcasts a **StatePacket** if $n + 1 \leq max\_n$ thereafter, which limits coverage of each cluster.

C. Construction and update of potential fields

We use potential-based routing [11], which is a proactive routing protocol, for data collection. The potential-based routing is known as a resilient routing protocol to environmental variations because it requires only local information. Every node updates its own potential, which is a scalar value calculated only with local information—own potential, neighbors' potential, and node degrees. It is worth noting that messages for the routing protocol is also utilized for the guidance of a mobile sink toward elected cluster heads.

In our methods, each cluster head constructs one potential field that is the shape of concave curve whose bottom corresponds to the cluster head. Therefore, multiple potential fields may be constructed in each network. Each potential field has a unique identifier pID that corresponds an identifier of the cluster head. The multiple potential-field construction process is given in Algorithm 3 with some terms tabulated in Table I.

First, cluster head i initializes its parameters such as $netID_i, myP_i$ and broadcasts **PInfoMsg** throughout the network (lines 1–10). **PInfoMsg** includes sender's ID, network ID, potential field IDs (pID), and potential values in correspondent potential fields, illustrated in Fig. 2. When receiving **PInfoMsg**, a sensor node updates **NTable**, myP_i, and $netID_i$ and broadcasts new **PInfoMsg** (lines 11-44). Here, **NTable** is a table to store information about a potential of neighbor nodes, and its entry is composed of five elements $(src, networkID, pID, pValue, time)$, a source node ID, a network ID, a potential field ID, a potential value in the correspondent potential field, and the time the entry was updated, respectively. **NTable** determines to either register or modify an entry when receiving **PInfoMsg** which contains information necessary for filling an entry of **NTable**. When a new entry has the same src and pID in **NTable**, the entry is registered and otherwise the existing entry is overwritten with it. S_i removes an old entry that is not updated for T_h, and it becomes a cluster head if **NTable** comes to have no entry.

Through the use of **NTable**, S_i updates its network ID ($netID_i$) and potentials (myP_i). $netID_i$ is the lowest value of pID registered in **NTable** and myP_i is updated according

Algorithm 3 Potential fields construction and update

```
if $State_i$ is CLUSTER(i,0) then
    $netID_i \Leftarrow i$
    $myP_i[i] \Leftarrow initial\_potential$
    puts $i$ into $pList_i$
    puts $myP_i[i]$ into $vList_i$
    broadcast PInfoMsg($i, netID_i, pList_i, vList_i$) per $T_{flood}$
    clear $pList_i, vList_i$
else
    $netID_i \Leftarrow NULL$
end if
loop
    if receive PInfoMsg($s, netID_s, pLIst_s, vLIst_s$) then
        for $j = 1$ to $k$ do
            update the NTable($s, netID_s, pList_s[j], vList_s[j], t_{S_i}$)
        end for
        for all $entry$ in NTable do
            if $t_{S_i} - entry.time > T_h$ then
                remove the $entry$ from NTable
                if NTable has no $entry$ then
                    $State_i \Leftarrow$ CLUSTER($i, 0$)
                    broadcast StatePacket
                else if NTable has no $entry$ whose $entry.netID$ is $netID_i$ then
                    if $State_i$ is CLUSTER($i, 0$) then
                        $netID_i \Leftarrow i$
                    else
                        $netID_i \Leftarrow NULL$
                    end if
                end if
            end if
        end for
        for $j = 1$ to $k$ do
            update all $myP_i[pList_s[j]]$
        end for
        if $netID_i$ is $NULL$ or $netID_i > netID_s$ then
            $netID_i \Leftarrow netID_s$
        end if
        for $j = 1$ to $k$ do
            puts $pList_s[j]$ into $pList_i$
            puts $myP_i[pList_s[j]]$ into $vList_i$
        end for
        broadcast PInfoMsg($i, netID_i, pList_i, vList_i$) after $T_{forward}$
        clear $pList_i, vList_i$
    end if
end loop
```

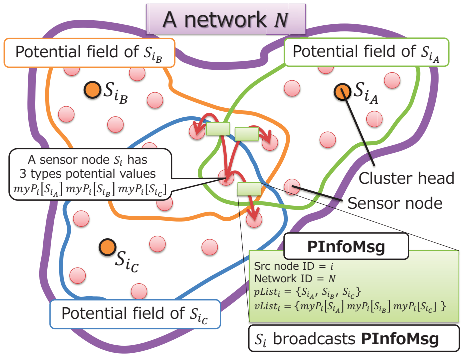

Figure 2. Example of a situation where S_i broadcasts a **PInfoMsg**

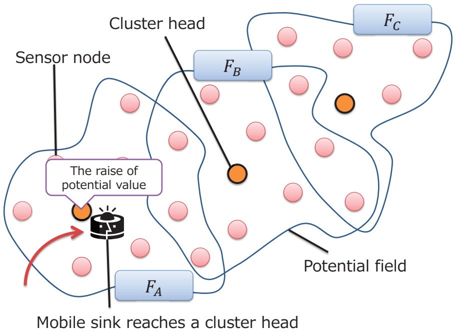

Figure 3. Increase of a potential value of a cluster head caused by arrival of the mobile sink

to the function in [11] where S_i's potential is calculated as an average potential of its neighbors. When disconnection of a network occurs, **NTable** may become to have no entry with $netID_i$ which equals to that of the table's owner, and therefore, S_i resets own $netID_i$ in that case. Thus, our construction method for multiple potential fields can respond to environmental changes inside a network.

Finally, S_i updates all myP_i and puts it to **PInfoMsg** and $T_{forward}$ after, it broadcasts **PInfoMsg**.

D. Traveling to cluster heads according to potential fields

Each cluster head has a peculiar potential field and holds the minimum potential value in its potential field. In these multiple potential fields, all sensor nodes have to do is forwarding their sensing data to one of their neighbor nodes which have a smaller potential value in **NTable**, and then, all sensing data reach to one of cluster heads. The mobile sink also utilizes these multiple potential fields, that is, utilizes $vList$ in **PInfoMsg** transmitted by a sensor node to reach a cluster head by repeatedly approaching sensor nodes which have a smaller value in $vList$ [9]. After arrival of a mobile sink to a cluster head, the cluster head makes its potential value raise greatly, which decreases the priority of the potential field for a mobile sink against other potential fields because the mobile sink approaches a sensor node with a 'smaller' potential value as (Fig. 3). After that, the mobile sink can find a new potential field (Fig. 4).

After visiting all cluster heads, a mobile sink goes away to unvisited networks. As explained in Section. II, a mobile sink visits all networks in the order it visited and memorized them in Phase 1. Therefore, the mobile sink moves on to the next network position.

IV. SIMULATION EVALUATION

In this section, we show that our proposal realizes the reliable data collection even though the changes of networks are caused by additions or breakdowns of sensor nodes.

A. Scenario of simulation

We assume a 1000m × 1000m square region including the whole observed area and deploy N sensor nodes $S_i(i = 0 \cdots N - 1)$ at uniformly random positions in the observed area, where the communication range of the sensor nodes is

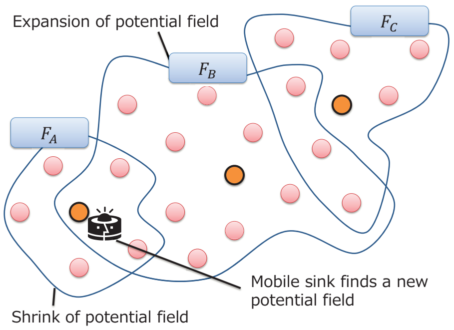

Figure 4. New potential field F_B is detected by the mobile sink after potential field F_A decrease its potential

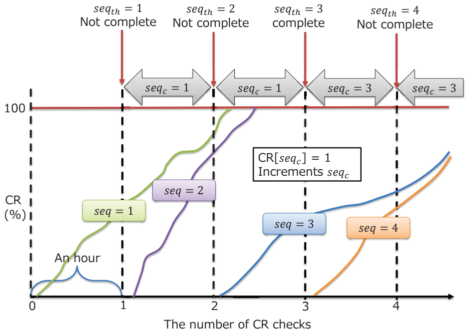

Figure 5. Example of CR over some seq

represented by a circle of radius 100m. The sensor nodes observe the surrounding environmental phenomena every hour and forward the packet including the sensing data, i of sender, and a sequence number seq, to one of cluster heads with potential-based routing. Furthermore, a number of sensor nodes $S_i(i = 0 \cdots B-1)$ will not be able to communicate with after T_{break} by failure and a number of sensor nodes $S_i(i = N \cdots N+A-1)$ newly will be deployed at random places at T_{add} in the observed area.

The mobile sink, whose speed is 5m/s, starts to move at 100s, and then, executes Phase1 and Phase 2 mobility strategies alternately. In this paper, we assume that the mobile sink can go all the region of the observed area. The mobile sink returns to the initial position to charge its battery and waits until the next hour begins. To put it in the concrete, the mobile sink, to begin with, sweeps the observed area to learn the networks deployed there and then returns to the initial position (Phase 1). After that, the mobile sink visits and enters all the networks using learned information on the networks to collect the sensing data and then returns to the initial position (Phase 2). The mobility of Phase 2 is executed LI times repeatedly. After that, the mobile sink performs Phase 1 mobility strategy again.

As the evaluation of reliable data collection, we calculate the collection rate every hour using (1) and some terms which are tabulated in Table I. Here, CDN means the number of already collected data, and EDN means the number of uncollected and existing data in the observed area.

$$CR[seq] = 100 \times \frac{CDN[seq]}{EDN[seq]} \qquad (1)$$

The mobile sink increments seq_c as long as $CR[seq_c]$ is equivalent to 100, which means that the all sensing data with seq_c are guaranteed to be collected. Then, we define the achievement of the reliable data collection as the situation that seq_c is the same as seq_{th}. Here, seq_{th} is a threshold and is equivalent to seq of the sensing data observed within this hour. Fig. 5 illustrates a example of the evaluation of CR over some seq. In Fig. 5, all sensor nodes observe the sensing data with $seq = 1$ within the first hour and the mobile sink has to collect them between the first CR check and the second CR check. The mobile sink, however, executes the Phase 1 mobility strategy which takes considerable amount of time at first. Therefore, the sensing data of the sensor nodes in the area where the mobile sink has already gone past are still uncollected. At the time of second CR check, the mobile sink has not finished collecting the sensing data with $seq_c = 1$ in spite of the fact that seq_{th} is 2. However, at the time of third CR check, $CR[sec_c = 1]$ reaches 1 and $CR[2]$ is 1 too. Then, we increment $CR[sec_c]$ twice and sec_c reaches $seq_{th} = 3$ at that time.

B. Parameter settings

In this paper, we simulate five scenarios in every cases which LI is from 1 to 5. All scenarios are simulated for 604,800 seconds and we performed 30 simulations for different arrangement of 40 sensor nodes for each scenario. Moreover, in those scenarios, 5 sensor nodes fail at different T_{break} and 10 sensor nodes will be placed in the observed area at different T_{add}.

C. Simulation results

To begin with, we show two types of transition of CR by the passage of simulation time when LI is 1 and LI is 5 in Fig. 6 to demonstrate that the reliable data collection can be attained by the combining of the two mobility strategies in our proposal even when the change of networks in the observed area occur. Here, in Fig. 6, the X axis is for the number of CR checks which are conducted every hour and the Y is for the average CR of 30 simulations at each CR check. Also, T_{break} and T_{add} in these scenarios are shown in Fig. 6 with arrows.

The line graph of $LI = 5$ in Fig. 6 shows that the average of CR starts to fall after the first T_{break} and T_{add}. However, after a while, it recovers to 100% again. This is because the mobile sink executes the learning of the networks within the observed area for every LI times CR checks and catches the changes of networks. Similarly, although the decline of the average of CR occur several times during the simulation, it always recovers to 100% after a while.

On the other hand, the average of CR remains 100% throughout the simulation when $LI = 1$. This is because the mobile sink frequently moves with the mobility of Phase 1 to learn the networks and catch the changes of networks quickly in the observed area.

Next, in order to evaluate the effects of the different value of LI give the reliability and efficiency of data collection, we show the relation between the achievement degree of reliable data collection and the total delay of the mobile sink's movement in Fig. 7. Here, in Fig. 7, the X axis is the total time of which the mobile sink moves for both learning the networks and collecting the sensing data and the vertical is

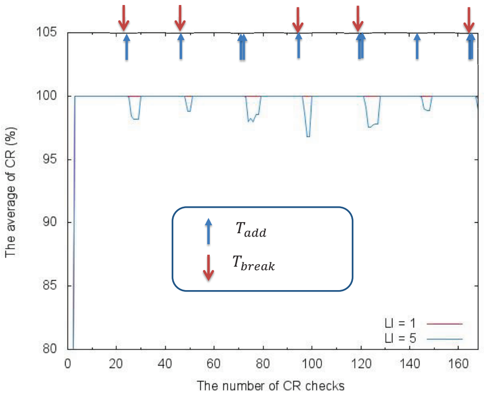

Figure 6. The transition of CR where $LI = 1$ and $LI = 5$

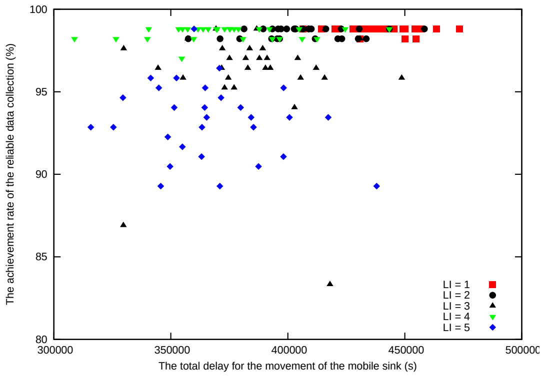

Figure 7. The relation between the reliability of data collection and the delay for mobility where $LI = 1 \cdots 5$

for the achievement rate of the reliable data collection, which is expressed as a percentage which CR is 100% in all CR checks.

According to the simulation results, the average delays for movement of the mobile sink are 438428s, 406360s, 385653s, 373907s, 364648s and the average achievement rates of the reliable data collection are 0.988, 0.986, 0.960, 0.986, 0.934 for each $LI = 1 \cdots 5$. With respect to the result for the delay, the more frequent the mobile sink executes the mobility for learning the networks in the observed area, the more delays it takes for the mobile sink to move, and vice versa. In other words, too frequent learning of the networks leads to redundant movements of the mobile sink in the case where there is no changes in the observed area. Then, the efficiency of data collection is not high. On the other hand, the achievement rates of the reliable data collection is higher value as the value of LI is higher expect for the case that LI is 4. This is because, in our scenarios, we suppose that the failure of the sensor nodes occurs only in the end of days and the mobile sink always executes the mobility for learning the networks in the end of days when LI is 4. The synchronization of occurrence of the failure of sensor nodes and the beginning of the learning networks allow the mobile sink to grasp the changes of the networks in the observed area very quickly. Therefore, the average achievement rates of the reliable data collection when LI is 4 is higher than when LI is 3.

From the results of simulations, it is obvious that the mobile sink with the combining of two mobility strategies in our proposal can collect the sensing data in the observed area in spite of the changes of networks. Moreover, the reliability of data collection and the efficiency of the data collection have a relation of a trade-off.

V. CONCLUSION

In this paper, we present two mobility strategies of the mobile sink to realize reliable data collection despite of the changes of networks in the observed area. Then, we demonstrate that reliable data collection is achieved by our proposal and show the relation of trade-off between the reliability and the efficiency of data collection. This trade-off can be adjusted by changing the frequency of the learning phase of the mobile sink. Our current interests are on the path planning among networks, and on implementing our proposal in actual mobile nodes.

REFERENCES

[1] H. Karl and A. Willing, Protocols and architectures for wireless sensor networks. Wiley, Oct. 2007, ISBN: 978-0-470-09510-2.

[2] S. V. Deshpande and P. Devale, “Recent trends in using wireless sensor networks in industrial environment,” International Journal of Computer Networking Wireless and Mobile Communication (IJCNWMC), vol. 3, 2013, pp. 11–20, ISSN: 2250-1568.

[3] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” The Journal of Supercomputing, vol. 68, 2014, pp. 1–48, ISSN: 0920-8542.

[4] J. Luo and J. Hubaux, “Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: the case of constrained mobility,” IEEE/ACM Transactions on Networking (TON), vol. 18, 2010, pp. 871–884, ISSN: 1063-6692.

[5] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, “Sink mobility protocols for data collection in wireless sensor networks,” in Proceedings of the 4^{th} ACM International Workshop on Mobility Management and Wireless Access (MOBIWAC) Octorber 2–6, 2006, Torremolinos, Malaga, Spain, Oct. 2006, pp. 52–59, ISBN: 1-59593-488-X.

[6] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, “Exploiting sink mobility for maximizing sensor networks lifetime,” in Proceedings of the 38^{th} Annual Hawaii International Conference on System Sciences (HICSS) January 3–6, 2005, Hawaii, America, Jan. 2005, pp. 287–295, ISBN: 0-7695-2268-8, ISSN: 1530-1605.

[7] F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo, and C. Richard, “Controlled mobility sensor networks for target tracking using ant colony optimization,” IEEE Transactions on Mobile Computing (TMC), vol. 11, 2012, pp. 1261–1273, ISSN: 1536-1233.

[8] R. Falcon, H. Liu, A. Nayak, and I. Stojmenovic, “Controlled straight mobility and energy-aware routing in robotic wireless sensor networks,” in Proceedings of the 2012 IEEE 8^{th} International Conference on Distributed Computing in Sensor Systems (DCOSS) May 16–18, 2012, Hangzhou, China, May 2012, pp. 150–157, ISBN: 978-1-4673-1693-4.

[9] S. Toyonaga, Y. Fujita, D. Kominami, and M. Murata, “Implementation of controlled sink mobility strategies with a gradient field in wireless sensor networks,” in Proceedings of the 7^{th} International Conference on Sensor Technologies and Applications August 25–31, 2013, Barcelona, Spain, Aug. 2013, pp. 27–32, ISBN: 978-1-61208-296-7, ISSN: 2308-4405.

[10] Z. Liu, Q. Zheng, L. Xue, and X. Guan, “A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks,” Future Generation Computer Systems, vol. 28, 2012, pp. 780–790.

[11] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled and self-organized routing for large-scale wireless sensor networks,” ACM Transactions on Sensor Networks (TOSN), vol. 10, 2013, p. 13.