Computer Communications 000 (2015) 1–18

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Energy-efficient mobile targets detection in the presence of mobile sinks

Guaning Chen[a,*], Jiu-Shu Cheuh[b], Min-Te Sun[b], Tsun-Chieh Chiang[c], Andy An-Kai Jeng[c]

[a] *Department of Information Management, Kainan University, Taoyuan 338, Taiwan*
[b] *Department of Computer Science and Information Engineering, National Central University, Taoyuan 320, Taiwan*
[c] *Industrial Technology Research Institute, Hsinchu 310, Taiwan*

ARTICLE INFO

Article history:
Received 10 November 2014
Revised 17 July 2015
Accepted 30 August 2015
Available online xxx

Keywords:
Wireless sensor networks
Data dissemination
Geometric routing
Grid network
Query aggregation

ABSTRACT

Tracking moving targets has become an increasingly important application for sensor networks. Sensor nodes may sense moving targets far away from the Source, and hence a large amount of energy may be wasted by them to send sensory data to the Source. Designing efficient algorithms and protocols for data dissemination to mobile sinks is an interesting research and engineering issue, especially for large-scale wireless sensor networks (WSNs). Sink mobility brings new challenges to the design of data dissemination. The location updates for each mobile sink need to be continuously propagated through the field to all sensor nodes, so that future data reports can be correctly delivered to the sink. As energy and resources of a sensor node are limited, these algorithms and protocols should meet a high energy efficiency and a high delivery ratio. To deal with this issue, we propose a framework, called Tree Overlay Grid (TOG), for data collection and dissemination. To route queries and deliver data efficiently in our framework, a geometric routing *GFB* (Greedy Forwarding within Bound) is proposed to create a TTDD-like grid network, and a tree protocol is used to construct local trees around sinks. In addition, two mechanisms are introduced to prolong the network lifetime. The first mechanism tries to save energy by reducing the traffic load; the second one tries to slow down energy consumption by balancing the traffic load. The simulation results show that TOG outperforms the best known data collection solution and some current data collection solutions for WSNs with multiple mobile sinks.

1. Introduction

A wireless sensor network (WSN) consists of a large number of homogeneous battery-operated sensor nodes interconnected together to form a network autonomously. Those sensor nodes will assist in sensing ambient conditions in an interested area. They then will store and process sensory information and send and forward gathered data to sinks or base stations for post-analysis. Sinks are resource rich devices.

The following delay sensitive applications can be realized by WSNs: healthcare monitoring, environmental monitoring, habitat monitoring, traffic monitoring, manufacturing monitoring, disaster management, inventory tracking, forest fire detection, target tracking, intrusion detection (e.g., enemy detection or tracking), battlefield surveillance, and so on.

Among them, we are interested in applications, which are classified as an event-driven type. They include intrusion detection (e.g., enemy detection or tracking), target tracking, battle-field surveillance, habitat monitoring, and forest fire detection. Sensor nodes, in an event-driven application, monitor a sensitive area and are programmed to periodically sense some specific events, such as an intrusion. They gather event reports which are then forwarded towards sinks for post-analysis. For instance, in Fig. 1, we see that an enemy tank enters the field monitored by a WSN. Surrounding sensor nodes are programmed to detect enemy tanks in the monitored area. The detected event reports will be sent to the collector, the Source, which then sends collected sensory information to respond to the soldiers' requests. This procedure is called data dissemination.

The flooding method to send report events to static sinks seems to be a good mechanism for data dissemination protocols, because it does not involve costly network topology management and complex routing algorithms. However, it results in a hotspot problem which sensor nodes closer to static sinks drain their energy faster than other sensor nodes farther away from sinks.

Mobile-sink-based routing protocol is used to reduce or avoid the effects of the hotspot problem caused by static-sink strategies. In addition, it can prolong the network lifetime to some extent and optimize energy. Energy is one of the major concerns in designing protocols for WSNs. Various data dissemination schemes [1–4] have been proposed over the years to reduce energy consumption among sensor nodes. Virtual grid-based data dissemination schemes [5–7] are preferred for event-driven type applications, because they require less energy in grid-based routing algorithms.

---

[*] Corresponding author. Tel.: +886 3 341 2500; fax: +886 3 341 2173.
E-mail address: gchen@mail.knu.edu.tw, lu_hartz@hotmail.com (G. Chen).

http://dx.doi.org/10.1016/j.comcom.2015.08.015

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

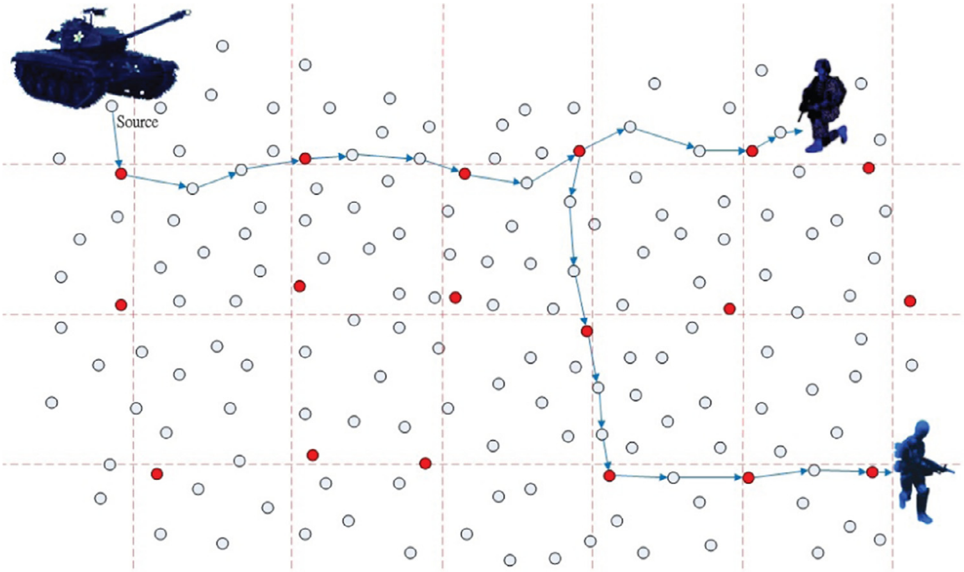


Fig. 1. The WSN is used in battlefield surveillance.

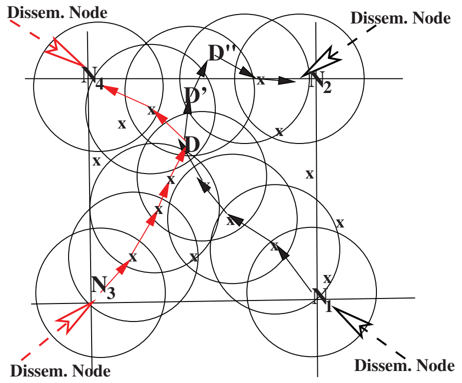


Fig. 2. The path between two dissemination nodes created by TTDD.

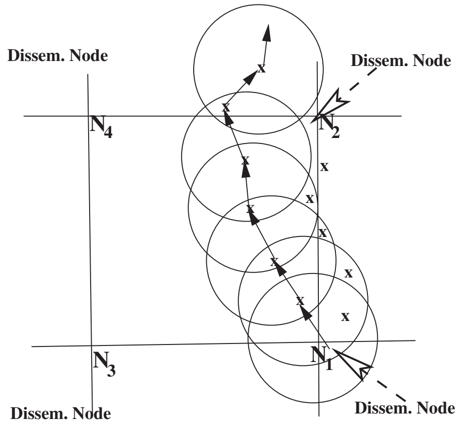


Fig. 3. The path between two dissemination nodes created by compass routing.

Authors of TTDD (Two-Tier Data Dissemination) [7], one of many studies on data dissemination in the presence of mobile sinks [8–12], employed a virtual grid structure to lessen energy consumption. However, there are a number of drawbacks due to the network itself and the routing algorithm used. In addition to the well-known dead-end issue [13], a routing path between two dissemination nodes may skew away from the line segment connecting the two dissemination nodes as shown in Fig. 2 when TTDD greedy geographical forwarding algorithm is employed. We also tried Compass Routing; Fig. 3 shows that Compass Routing fails to construct a path between two dissemination nodes. Namely, the resulting network may not only be disconnected but also non-grid-like. We wonder if there exists a sensor network and a routing algorithm such that routing paths constructed by the algorithm form a grid-like sensor network.

The major contributions of this study are as follows. First, unlike TTDD, the routing algorithm *GFB* in our framework guarantees that grid paths created by *GFB* do not intersect within a grid cell. Second, our framework introduces a new role of a sensor node, called a reporter, to handle multiple mobile stimuli/targets. Third, unlike TTDD, our framework uses only one network to handle multiple mobile targets. Fourth, we provide formal proofs for statements used in the study and algorithms.

The rest of the paper is organized as follows. In Section 2, we review related work. The network model and the detail of the design of TOG is described in Section 3. The simulation results and analysis are presented in Section 4. Section 5 concludes this paper. The facts used in Section 3 are presented as lemmas, theorems, propositions, or corollaries. With their proofs, they are collected in Appendices.

2. Related work

As mentioned in the previous section, many data dissemination protocols have been invented in the presence of mobile sinks. The primary objective of mobility schemes is to either deliver data in disconnected sensor networks or improve the network lifetime. Basagni et al. [14] classifies sink mobility schemes into three categories: random mobility, predictable mobility, and controlled mobility. In a random mobility scheme, no network information is required as the movement of the sink is random. The random mobility scheme is the most widely adopted scheme in WSN. TTDD and TOG protocols belong to such a scheme. In a predictable mobility scheme, a mobile sink injects its trajectory information into the network then sources use such information to determine the sink's future location. PMDD (Predictable Mobility-based Data Dissemination) [15] is an example of such a scheme. In a controlled mobility scheme, sink movement is controlled by certain network parameters such as residual energy, stimulus location, etc. A sink takes movement decisions to increase the network's lifetime. One such example is GMRE (Greedy Maximum Residual Energy) mechanism proposed by Basagni et al.

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

By considering the number of hops transmission towards the sink, Sharma et al. [11] broadly classifies data dissemination protocols in the presence of mobile sinks as: single-hop and multi-hop. In a single-hop data dissemination protocol, a mobile sink broadcasts the small beacon packets periodically while it moves to a new location. The sensor nodes in the mobile sink's communication range receive the beacon packets and transmit their collected data to the sink [16]. Authors of [17–19] claim these types of protocols are not suitable for the delay sensitive applications. In a multi-hop data dissemination protocol, most of the collected data are transmitted towards the sink by passing through more than one hop. Based on the routing schemes used, these types of protocols can be further divided into two broad categories: non-hierarchical (flat) routing and hierarchical routing.

In a non-hierarchical (flat) routing protocol, all sensor nodes with the same role and responsibilities work together to route the data in the network. Gossiping [20] and flooding are typical examples of non-hierarchical routing protocols. Such non-hierarchical routing protocols do not scale well due to frequent location updates from mobile sinks. Therefore, overlaying a virtual infrastructure over the physical network has often been investigated as an efficient strategy for data dissemination in presence of mobile sinks [8]. In a hierarchical routing protocol, sensor nodes in terms of their roles in a network are placed into different level of hierarchy. The role of a sensor node in the network may be assigned based on metrics such as energy, location, coverage, etc. Generally, roles are not static over the course of time. Sensor nodes may change their roles based on the underlying strategy. In the following, we will discuss three most commonly used hierarchical routing protocols in WSNs: Cluster based, Virtual Grid based, and Tree based.

1. Cluster-based: A cluster-based protocol aims to cluster the network into zones so that cluster heads (CHs) of zones can do some aggregation and fusion of data in order to save energy. It can be implemented in WSN for mobile sink data dissemination protocols. HCDD (Hierarchical Cluster-based Data Dissemination) [9] employs three procedures to reduce the overhead in the presence of a mobile sink. First, all sensor nodes are divided into clusters and each cluster designates a sensor node as the cluster head (CH). Second, a sink will register to one of the Routing Agents as it moves. Finally, CHs and Routing Agents cooperated to find the paths from sources to the mobile sink.
2. Tree-based: Tree-based protocols are parent-child hierarchical routing schemes. TEDD (Tree-based Efficient Data Dissemination) [11] consists of four phases: Neighbor discovery, Tree construction, Relay node selection, and Data dissemination. Through the neighbor discovery phase, the neighboring information of each sensor node is found. Using the neighboring information, a connected tree is established during the tree construction phase. During the relay node selection phase, relay nodes are picked and then data dissemination paths can be constructed. The sink collects the detected data stored in sources through its gateway node. If the sink moves out from the range of the current gateway node, then it elects another sensor node as the new gateway node.
3. Virtual grid-based: Some hierarchical routing protocols cluster the network into fixed zones to obtain a fixed rectilinear virtual topology. Inside each zone, a sensor node is optimally selected to act as CH. Data dissemination is performed on two levels: local and global. TTDD [7] employs a virtual grid structure to solve the issues of sink mobility and energy consumption. When a target is detected for the first time, one of the nearby sensor nodes will be elected as the Source and a grid network will be built. Intra-grid sink mobility is maintained by an immediate agent (IA) and primary agent (PA). Inter-grid sink mobility is handled by using flooding to locate a new immediate dissemination node. However, the flooding operation is expensive in WSN. In TTDD, if the target moves out of the sensing range of the Source, the grid needs to be rebuilt. Energy is wasted due to the reconstruction of the grid. Lee et al. [4] proposed a grid-based protocol which constructs an Independent Grid Structure (IGS) in the center of sensor network. IGS is a k-layer grid structure with four grid cells as the innermost layer. In Lee's Protocol, sources are required to send event reports to grid headers in the innermost layer of IGS and sinks can send queries to grid headers in IGS to request event reports. Sensor nodes in IGS provide aggregation of reports sent from sources and multi-casting of the aggregated reports to sinks. If a sink moves, it makes a routing path from previous location to a new location by the foot-print chaining or re-registers its location to IGS at the new location. The protocol uses the foot-print chaining for the local movement of a sink and the re-registration to IGS for the global movement of the sink. EEGBDD (Energy Efficient Grid-Based Data Dissemination) [1] exploits location information of sensor nodes to build a virtual grid structure over the entire sensor field once the first stimulus is appearing in the field. The proposed network model ensures queries and responses forwarding through a diagonal forwarding algorithm between sources and sinks. Queries and responses are forwarded by the dissemination nodes only. EEGBDD can be used to handle data dissemination even when both sinks and stimuli are mobile. Local cell sink mobility can be handled by local flooding of data within that cell; but inter-cell sink mobility requires an agent node, called sink manager, to keep track of the sink location in order to forward the data to the sink.

In this paper, we modify and extend our previous framework, DAG (Dynamic and Adaptive Grid) [21], to disseminate data in WSN. First, we construct a two-level network structure: the top level is a directed grid-based network created by the Source for the monitored area; the bottom level may consist of trees rooted at some dissemination nodes for Tree_Areas which is defined in Section 3.2.1. The grid-based network helps distribute queries, query responses, and event reports correctly to and from the Source. The trees help save energy by reducing the amount of messages forwarded in data dissemination and queries. Second, two mechanisms are introduced to prolong the network lifetime. The first mechanism tries to save energy by reducing the traffic load. TOG employs a query aggregation scheme to reduce the traffic load. In addition, a query aggregation scheme also improves the query response time. The second mechanism tries to slow down energy consumption by balancing the traffic load. TOG introduces four load balancing techniques to slow down energy consumption.

TOG can be said, in general, to be a cluster-base protocol in the presence of mobile sinks as HCDD. For data dissemination, TOG introduces a new routing algorithm, *GFB*, to create grid-like routing paths which are claimed to be paths aligned with grid lines. However, the routing algorithm provided by TTDD cannot guarantee such a behavior. Unlike TTDD and TOG, the grid-cell sizes of EEGBDD and Lee's Protocol are limited and inflexible depending on the radio communication range. To avoid TTDD's broadcast mechanism to find a dissemination node, TOG employs a tree-based routing structure as TEDD; but a tree is only created in a Tree_Area when a sink appears there.

3. Tree overlay grid (TOG)

This section describes the design of TOG data dissemination framework which extends our previous work, DAG. The network model and assumptions of TOG are listed in the following:

- The monitored field F is convex and fully covered by a wireless sensor network. We assume that the boundary of the monitored field is known. Given any two distinct boundary points, a boundary curve can be determined [22]. A boundary curve is assumed to be continuous and non differentiable at only countable points. We are interested in how such a field is monitored by a wireless sensor network. Sensor nodes considered here are located inside

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

the monitored field. Sensor nodes are homogenous in terms of fixed sensing range r_s and communication range r_c. We assume also that the communication range is twice as large as the sensing range. A point with its distance to a sensor node X less than r_s can be sensed by X. We say that the point is "covered" by X. However, the points on the circumference of a circle, $C_{r_s}(X)$, centered at X with a radius of r_s or outside the circle cannot be sensed by X. The sensing range is relatively small compared to the monitored field.

- Sensor nodes are deployed as a network to monitor, detect, and collaboratively collect information about specific events that occur in the field. Each sensor node has a unique ID and is aware of its geographical location through a global positioning system (GPS) [23] receiver or some localization techniques [24–28]. At the beginning, each sensor node knows the sensing mission and the grid size which is a multiple of the communication range r_c. For example, the missions of a sensor node can be motion detecting, temperature taking, or other target signature capturing.
- A one-hop neighbor (neighbor for short) of a sensor node, say X, is defined as a sensor node whose distance to X is less than the communication range r_c. A sensor node can send its information to and receive information from its neighbors. Each sensor node knows the geographical coordinates of its one-hop neighbors.
- A sensor node can be a reporter, a source, an Immediate Agent (IA), a Primary Agent (PA), a relay node, a tree node, or a dissemination node. In addition to those sensor entities, there are particular entities: targets and sinks. To make the model more realistic, both targets and sinks are mobile. A mobile target as an intruder, the network designed to monitor, will be detected as long as it enters into the monitored field. A target zone is an open disk centered at a target with a radius of r_s. We say that a target zone is formed when a target appears in the monitored field. The location of a target is the location of a reporter which senses a target and reports it to the Source. When a target moves away from the surveillance area of a reporter (initially, the area is observed by the Source), another sensor node will be elected as a new reporter to closely observe the target. The reporter generates event reports and sends reports to the Source which collects information of targets.
- An IA is a tree node in the neighborhood of a sink. It helps the sink forward query to and receive query's response from the Source via a PA which is a dissemination node. A relay node is an internal sensor node of a directed path between adjacent dissemination nodes. When receiving a query or a response, dissemination nodes and relay nodes need to record them. Sinks can move freely around the monitored field and may have information of an IA and its associated PA at hand. A sink can query the Source about targets via an IA through its associated PA.
- In addition to the role of a sensor node, such as a dissemination node, in a dissemination network, each sensor node has a data structure used to indicate its status in a dissemination tree rooted at a dissemination node. The data structure of a sensor node contains the tree ID (the ID of the tree root) used to identify the tree which is associated with the sensor node, the ID of its parent, the hop count of the sensor node (denoted as T-hop which is the height of the sensor node), and IDs of its children. The tree ID, parent ID, and T-hop value of a sensor node are initialized to ∞ to mean that no tree is associated with this sensor node. The IDs of its children are initialized to IDs of its one-hop neighbors. They are collected in Initialization 1 on page 24. Some sensor nodes may associate with more than one dissemination tree.

Both the grid-based structure used in TTDD and the tree structure used in SEAD (Scalable Energy-efficient Asynchronous Dissemination) [2] have their pros and cons: the grid-based network helps distribute the data, but the routing path may not be the shortest one. On the other hand, the tree structure helps report events or send queries efficiently, but the sensor nodes near the root of a tree have the hotspot problem. Therefore, TOG employs a grid-based structure (called a dissemination network) on the top to distribute the traffic occurred in the whole network, and tree structures on the bottom to report events, send queries, and receive query responses occurred in grid cells.

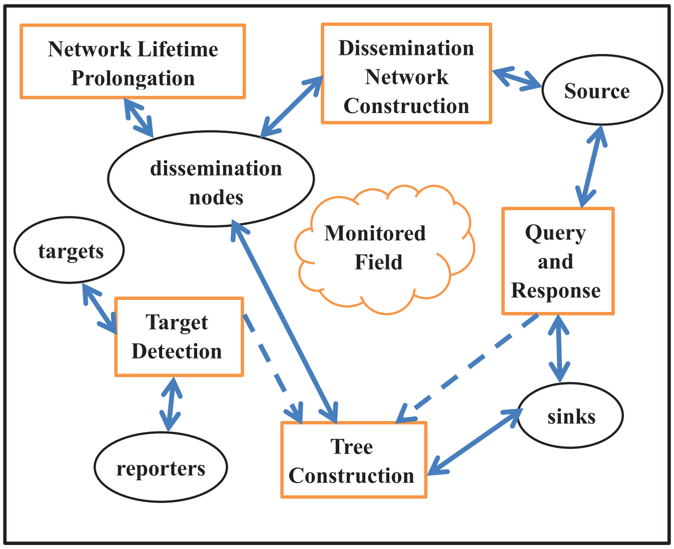


Fig. 4. Five modules and five main roles in TOG framework.

Fig. 4 shows the overall structure of TOG framework. There are five modules which are squared in Fig. 4. Each of them may involve more than one of the five main roles which are circled in Fig. 4. Forwarding queries and target information are first handled through tree nodes. Since forwarding queries occur under the query and response module and forwarding target information occurs under the target detection module, the tree construction module can be considered as their submodule.

- The first module is the dissemination network construction which involves the Source and dissemination nodes. Once the first target is detected, the Source, which is elected by sensor nodes around the target, will start to select a grid structure. Based on the selected grid structure, the Source employs *GFB* to determine dissemination nodes and construct directed paths to them. The determined dissemination nodes and path nodes continue to employ *GFB* to generate more dissemination nodes and construct directed paths between them until a dissemination network is completed.
- The second module is the tree construction which involves reporters, sinks, and dissemination nodes. A sink or a reporter will employ a tree structure to forward queries or target information to an immediate dissemination node towards the Source. If a tree structure does not exist in time, either a sink or a reporter will request an immediate dissemination node to create a tree in its Tree_Area.
- The third module is the target detection which involves targets and reporters. Whenever a new target or an existing target, which moves to a new location, is detected, a new reporter will be elected to monitor the target and send target information to the Source.
- The fourth module is the query and response which involves the Source and sinks. Whenever a sink needs target information, it sends a query to the Source and the Source will send back a response to the sink.
- The fifth module is the network lifetime prolongation which involves dissemination nodes. During the operation, dissemination

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

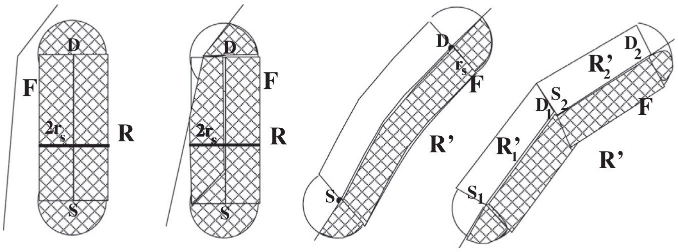


Fig. 5. The regular strips are shaded areas.

nodes are responsible for prolonging the network lifetime by reducing the traffic load to save energy and balancing the traffic load to slow down energy consumption.

In the following, we will describe each module in detail.

3.1. *Dissemination network construction*

Before describing the construction of a dissemination network, we introduce a novel routing algorithm which is a variation of Compass Routing [29] and called Greedy Forwarding within Bound (*GFB*). By bound here we mean that it is a narrow strip formed by four differentiable curves such that two (long) curves are parallel[1] and two other (short) curves are semi-circles having the same diameter. Let $\widetilde{SD}$ be a simple and differentiable curve inside the monitored field. For $X \in \widetilde{SD}$, we assume that the circle $C_{r_s}(X)$ intersects at the curve $\widetilde{SD}$ at most two points. As shown in Fig. 5, S and D are centers of two semi-circles of the strip and $\widetilde{SD}$ bisects the strip. $R(\widetilde{SD})$ (short for R) is used to denote the intersection area between the strip and the monitored field if $\widetilde{SD}$ is a line segment inside the monitored field. $R'(\widetilde{SD})$ (short for R') is used to denote the intersection area between the strip and the monitored field if $\widetilde{SD}$ is the part of the boundary of the monitored field. If $\widetilde{SD}$ is the part of a monitored boundary curve and not differentiable, it is broken into a series of curves $\widetilde{S_iD_i}$ at non differentiable points $S_i = D_{i-1}$ for $i = 1$ to $i = k$ such that each sub-curve is differentiable. Therefore, $R'(\widetilde{SD})$ may be composed of sub-strips $R'_1(\widetilde{S_1D_1}) \ldots R'_k(\widetilde{S_kD_k})$ such that R'_i is associated with $\widetilde{S_iD_i}$ as shaded area shown in Fig. 5(c). Those areas are called regular strips and denoted by R/R' or $R(\widetilde{SD})/R'(\widetilde{SD})$ interchangeably.

GFB is a distributed routing algorithm and can also be used to connect the current path-node to the next such that the constructed path is located inside the strip bound $R(\widetilde{SD})$ or $R'(\widetilde{SD})$. When we say a path is located inside R or R', we mean that all its path nodes are inside R or R'. *GFB* is employed and a path construction is started at a sensor node which covers the point S of $\widetilde{SD}$. The routing is stopped at a particular sensor node or at any sensor node which covers D; otherwise, a current path-node employs *GFB* to select its next path-node, say N, inside the strip R or R' such that $C_{r_s}(N)$ intersects $\widetilde{SD}$ at a point, which has the least curve length to D.

The Source is a unique sensor node dedicated to collecting event reports sent by reporters. It will send out event reports to queriers, i.e., sinks. It may be elected by other sensor nodes or assigned by the previous source. Initially, the newly elected source creates a referenced grid structure for a dissemination network by setting itself as the origin, d as the grid size, and cardinals E/W and N/S as the x-axis and y-axis of the referenced grid structure, respectively. Based on the given grid structure, a dissemination network for the sensor network will be constructed. A dissemination network is a directed graph consisting of vertices and edges. A vertex is called a dissemination node which is a sensor node closest to a referenced grid point and the one with the least ID wins the tie-break. An edge may exist if two dissemination nodes are adjacent. We say that two dissemination nodes are adjacent if their associated referenced grid points are adjacent. An edge is a directed communication path connecting two adjacent dissemination nodes. Now, we will describe the steps for constructing a dissemination network: the calculation of next referenced grid points, the determination of whether there are directed paths towards those referenced grid points, and then the construction of directed paths.

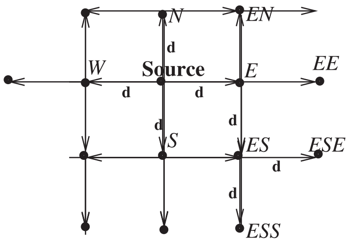
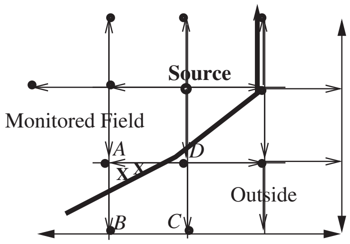


Fig. 6. The next referenced grid point.



Grids marked by ⊙ are considered as dissemination nodes

Fig. 7. The next referenced grid point under the monitored field.

3.1.1. *Calculation of next referenced grid points*

The construction of a dissemination network starts from the very first dissemination node which is the Source. The determination of next downstream (away from the Source) dissemination nodes and directed paths to these new dissemination nodes is based on the location of the current referenced grid point. We use N_{cd} to denote the current dissemination node and P_{cr} to denote the current referenced grid point. If the current referenced grid point is the origin, the current dissemination node (i.e., the Source) calculates next four referenced grid points in four cardinal directions. The next four referenced grid points are one grid size, d, away from the current referenced grid point (i.e., the origin). The next referenced grid point is denoted as P_{nr} and its associated new dissemination node will be denoted by N_{nd}. If the current referenced grid point is a cardinal point, the current dissemination node calculates next three referenced grid points which are away from the Source and one grid size away from the current referenced grid point. If the current referenced grid point is a point in a quadrant, the current dissemination node calculates next two referenced grid points which are away from the Source and one grid size away from the current referenced grid point. The main idea is to select next dissemination nodes away from the Source. For example in Fig. 6, if the current referenced grid point *ES* is in the fourth quadrant, next two referenced grid points are *ESE* and *ESS* due East and South, respectively. Although we are not interested in the area outside of the monitored field, some referenced grid points outside the monitored field may be used to calculate dissemination nodes. For example in Fig. 7, the grid cell, *ABCD*, contains sensor nodes; but only the grid point *A* is located inside the monitored field. Although the other three grid points are outside the field, their corresponding dissemination nodes are still needed to be calculated for directed paths.

---

[1] A parallel of a curve is the envelope of a family of congruent circles centered on the curve [30].

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

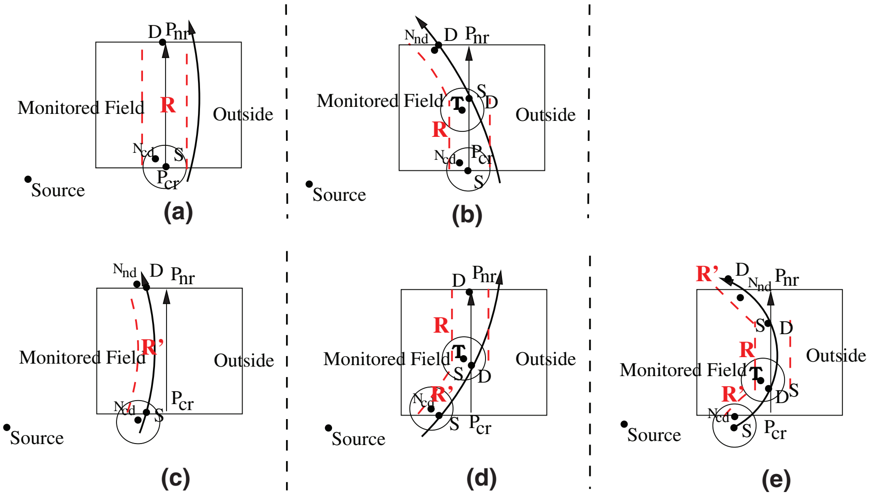


Fig. 8. The possible directed paths.

3.1.2. Determination of whether there are directed paths

Once the current dissemination node N_{cd} of point P_{cr} finishes calculating the next referenced grid point P_{nr}, the path construction procedure is stopped at N_{cd} if both the left-hand side and the right-hand side grid cells of $\overrightarrow{P_{cr}P_{nr}}$ contain no sensor nodes; otherwise, the directed path construction procedure, *GFB*, will be invoked to construct a path.

3.1.3. Construction of directed paths

How a directed path is constructed depends on whether the whole or a part of $\overline{P_{cr}P_{nr}}$ is located inside or outside the monitored field. First, N_{cd}, the dissemination node of P_{cr}, checks whether P_{cr} is inside the monitored field and whether the line segment $\overline{P_{cr}P_{nr}}$ is broken by the boundary curve. The construction of a directed path can be divided into five scenarios, and the first two (in Fig. 8(a and b)) occur when P_{cr}'s are inside the monitored field. WLOG, they are shown in Fig. 8.

Scenario (a): Let S be P_{cr}, D be P_{nr}, $\widetilde{SD}$ be $\overline{SD}$, and $R(\widetilde{SD})$ be its corresponding regular strip as shown in Fig. 8(a). Now, N_{cd}, the dissemination node which covers S, employs *GFB* to construct a directed path inside $R(\widetilde{SD})$. By Theorem 5 and Corollary 6, a directed path from N_{cd} to N_{nd}, which is the dissemination node of P_{nr}, can be constructed such that the path is located inside R.

Scenario (b): Let S be P_{cr}, D be the intersection point of $\overline{P_{cr}P_{nr}}$ and the boundary curve, and $R(\widetilde{SD})$ be its corresponding regular strip as shown in the lower part of Fig. 8(b). Now, N_{cd}, the dissemination node which covers S, employs *GFB* to construct a directed path inside $R(\widetilde{SD})$. By Theorem 5, a directed path from N_{cd} to a sensor node, say T, which covers D can be constructed such that the path is located inside R. Now, let S be the above point D and the new point D be the intersection point of the boundary curve and the grid side along path direction. As can be seen from the upper part of Fig. 8(b), P_{nr} is a point outside the monitored field and the angle between $\overrightarrow{SD}$ and $\overrightarrow{SP_{nr}}$ is less than 90°. By Corollary 9, the dissemination node N_{nd} of P_{nr} can be found. Let R' be a regular strip containing T and N_{nd}. To construct the remaining path, *GFB* is employed by the new N_{cd} denoted by T in Fig. 8(b), which covers a new S. Again, Corollary 9 tells us that a directed path from T to N_{nd} can be constructed and located inside R'.

Scenario (c): Let S be a boundary point covered by N_{cd} and D be a boundary point such that $\overrightarrow{SD}$ is aligned with the path direction. As shown in Fig. 8(c), P_{nr} is a point outside the monitored field and the angle between $\widetilde{SD}$ and $\overrightarrow{SP_{nr}}$ is less than 90°. By Corollary 9, the dissemination node N_{nd} of P_{nr} can be found. Let R' be a regular strip containing N_{cd} and N_{nd}. Now, N_{cd}, the dissemination node which covers S, employs *GFB* to construct a directed path inside R'. Again, Corollary 9 tells us that a directed path from N_{cd} to N_{nd} can be constructed and located inside R'.

Scenario (d): Let S be a boundary point within the sensing range of N_{cd}, D be the intersection point of $\overline{P_{cr}P_{nr}}$ and the boundary curve $\widetilde{SD}$ be part of the boundary curve connecting S and D, and $R'(\widetilde{SD})$ be its corresponding regular strip as shown in the lower part of Fig. 8(d). Now, N_{cd}, the dissemination node which covers S, employs *GFB* to construct a directed path inside $R'(\widetilde{SD})$. By Theorem 5, a directed path from N_{cd} to a sensor node T covering D can be constructed such that the path is located inside R'. At this moment, let S be the previous point D, the new point D be P_{nr}, $\widetilde{SD}$ be $\overline{SD}$, and $R(\widetilde{SD})$ be its corresponding regular strip. To construct the remaining path, *GFB* is employed by the new N_{cd}, denoted by T in Fig. 8(d), which covers a new S. Again, by Theorem 5 and Corollary 6, a directed path from T to N_{nd}, the dissemination node of P_{nr}, can be constructed so that the path is located inside R.

Scenario (e): It is not hard to see that the lower part of this scenario is the same as the lower part of scenario (d), the middle part is similar to the lower part of scenario (b), and the upper part is the same as the upper part of scenario (b).

From the above description, it is easy to see the following proposition.

Proposition 1. *Let P_{cd}^{nd} be a directed path from N_{cd} to N_{nd} which are two adjacent dissemination nodes. If the whole path P_{cd}^{nd} is neither located inside strip R nor R', then portions of P_{cd}^{nd} are located inside R and other portions are located inside R'.*

3.2. Tree construction

In TTDD, a sink will flood the message in a grid cell to find the nearest dissemination node. The flooding operation causes much overhead in wireless sensor networks. The tree structure makes data dissemination more efficient. We also believe that sinks and targets move around the field most likely following the locality principle. Taking advantage of it, trees in TOG will only be constructed in grid cells where targets or sinks move in.

3.2.1. Tree_Area definition

Before describing the actions taken by the algorithm *TREE* Procedure 3 to construct a tree, we first introduce a concept of a Tree_Area. For each referenced grid point P_r, a Referenced Tree_Area, shown in Fig. 9 as a dashed square, is defined to be the same shape and size of a referenced grid cell but centered at P_r. The Tree_Area of

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

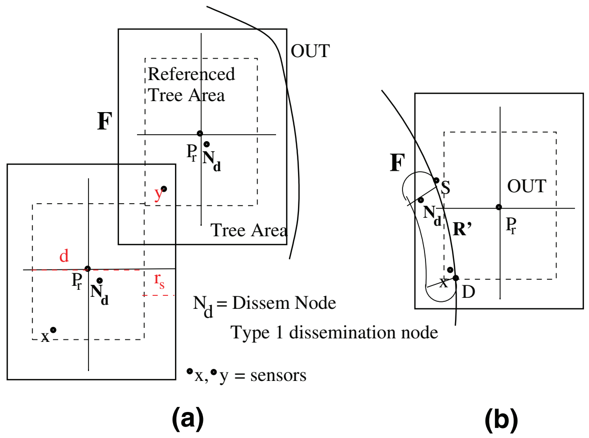


Fig. 9. The Tree_Areas and dissemination nodes are type 1 sensor nodes.

a dissemination node (or its associated referenced grid point) is defined as an expanded area of its Referenced Tree_Area shown in Fig. 9 as a solid square. It is a square with a side length of $d + 2r_s$ and centered at P_r. If a dissemination tree, rooted at a dissemination node, needs to be created, it will be constrained in its Tree_Area. It is easy to see that some sensor nodes, e.g., sensor node y in Fig. 9(a), may belong to up to four different dissemination trees. If the Referenced Tree_Area of a referenced grid point contains only non-dissemination nodes, a regular strip $R'(\widetilde{SD})$, as shown in Fig. 9(b), will be created. The strip $R'(\widehat{SD})$ is constructed so that it contains the dissemination node[2] N_d as shown in Fig. 9(b) and at least one sensor node in the Referenced Tree_Area. Also, the boundary point S is covered by the dissemination node N_d. Since some sensor nodes may belong to four possible dissemination trees, each sensor node has a data structure to relate its connected trees. Items of the data structure and their initializations are listed in Initialization 1 and exist at the beginning of the deployment.

Initialization 1 INIT(SensorNode).

```
SensorNode Sender, Receiver;
  SensorNode.ID;
for i=1 to 4 do
    SensorNode.T_i-ID ← ∞;
    SensorNode.T_i-parent ← ∞;
    SensorNode.T_i-hop ← ∞;/* record the hop count of the i-th tree to the root */
    SensorNode.T_i-child ← Sensor node's 1-hop neighbors;
end for
```

If a referenced Tree_Area contains sensor nodes, the dissemination node of this referenced grid point is classified as a type 1 dissemination node; otherwise, the dissemination node is classified as a type 2 dissemination node. Of course, no tree will be constructed for any type 2 dissemination node.

3.2.2. Tree creation algorithm

Whenever a dissemination node initiates or is requested to create a tree, it calls *TREE* Procedure 3 to complete its work. *TREE* Procedure 3 first broadcasts a message TREE_CNST (Line 5) to ask receivers, which are inside the Tree_Area (Line 11 to Line 14), to check and execute the following (Lines after 14): If the receiver is already a tree node with the shortest hop-count (Line 15 to Line 20), it stops re-broadcasting. If the receiver is already a tree node with a longer hop-count (Line 21 to Line 28) or is not a tree node (Line 29 to Line 35) of a being constructed tree, *TREE* Procedure 3 will call TREENODE Subroutine 2 (Line 24 or Line 31) to insert the receiver into the tree. Once a new tree node is inserted, it will re-broadcast the message TREE_CNST (Lines 25 and 32) to its neighbors and starts over again (Line 24 to Line 26 and Line 31 to Line 33) until all sensor nodes inside the Tree_Area have gone through the process. As mentioned above, a sensor node may belong to up to four different dissemination trees. Therefore, for-loops (Lines 15, 21, and 29) related to the tree node may need to repeat up to four times. As claimed by Corollary 12, the set of sensor nodes formed by this way is actually a tree rooted at the dissemination node.

Subroutine 2 TREENODE(tree, SensorNode, broadcast_msg).

```
OUTPUT: /* A sensor node is inserted into a tree rooted at a Dissem-node. */
SensorNode.tree-ID ← broadcast_msg.tree-ID;
SensorNode.tree-parent ← Sender.ID;
SensorNode.tree-hop ← broadcast_msg.hop-count;
Sensor node sends an ACK to Sender; /* parent-child relationship. */
if broadcast_msg.Sender receives an ACK then
   broadcast_msg.Sender.tree-child.ID ← SensorNode.ID;
end if
```

3.3. Targets detection

In a target zone, all sensor nodes except a reporter will activate and reset their timers at a predefined time point once they detect a target. The reset values of timers for tree nodes are set to TIME1, which is slightly longer than the time of one-hop message transmission, t_c.[3] The reset values of timers for non-tree-node dissemination nodes are set to TIME2, which is TIME1+t_e, where t_e[4] is the maximal time taken to elect a new reporter in a target zone. The reset values of timers for the remaining sensor nodes in the target zone are set to TIME3, which is TIME2+t_e. There are four different actions to take depending on the roles of sensor nodes in the target zone. They are elaborated on as follows.

3.3.1. Target zone contains the current reporter

If a reporter still detects a target, it continues as scheduled to report the event to the Source. The reporter sends the event report through its tree parent node then it informs sensor nodes in the target zone that it is still the reporter. At the same time, all sensor nodes will deactivate their timers and continue to sense their surroundings periodically.

3.3.2. Target zone contains some tree nodes but not the current reporter

After some wait time, say TIME1, if no INFORM message is received, all tree nodes in the target zone collaboratively elect a new reporter among themselves. Immediately, the elected reporter informs sensor nodes in the target zone that it is a new reporter and will periodically report the event via the tree root to the Source. At the same time, all sensor nodes will deactivate their timers and continue to sense their surroundings periodically.

3.3.3. Target zone contains dissemination nodes but neither tree nodes nor the current reporter

At the end of the wait period, TIME2, dissemination nodes in the target zone receive no INFORM messages from a reporter. It is easy to

---

[2] By the definition of a dissemination node, this dissemination node exists in the Tree_Area but outside its Referenced Tree_Area.

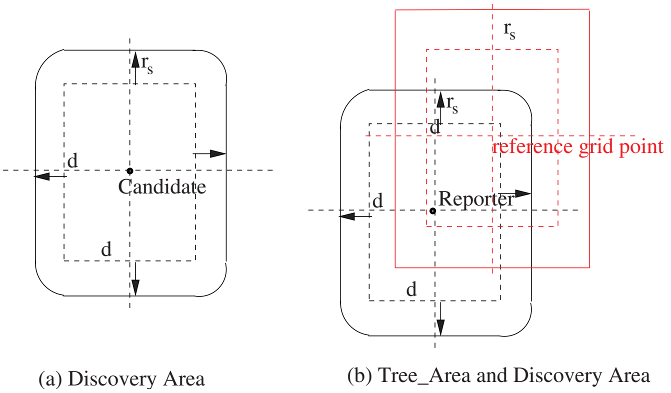
[3] $t_c = r_c/v$, where r_c is a communication range and v is a radio speed.

[4] t_e depends on an election mechanism employed. It may take a few rounds of broadcasts to elect a new reporter. Namely $t_e \le n \cdot r_c$, where n is a small number.

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

Procedure 3 TREE(dissem-node, ref-grid-pt, Tree_Area).

```
INPUT: /* In addition to the referenced grid point and Tree_Area, each sensor node has a data structure to relate its four possible dissem. trees as mentioned before. */
label child_node, out;
message TREE_CNST, ACK; /* the information TREE_CNST message contains */
TREE_CNST.Tree_Area;TREE_CNST.tree-ID ← Dissem-node.ID;
TREE_CNST.hop-count ← 1; TREE_CNST.sender ← Dissem-node;
OUTPUT:/* A tree rooted at this Dissem-node and bounded within its Tree_Area. */
Dissem-node.T-hop ← 0; /* dissemination node sets itself as a tree root*/
Dissem-node.T-parent ← Dissem-node.ID;Dissem-node.T-ID ← Dissem-node.ID;
Dissem-node broadcasts TREE_CNST;
Receivers ← Dissem-node's Neighbors;
child_node:
if {Receivers == ∅} then
  goto out;
end if
for all {Receiver ∈ Receivers} do
  if {Receiver is outside Tree_Area} then
    Receiver drops TREE_CNST; Receivers ← Receivers \ Receiver;
    goto child_node;
  else
    for i=1 to 4 do
      if {{Receiver.$T_i$-ID == TREE_CNST.tree-ID} and {TREE_CNST.hop-count ≥ Receiver.$T_i$-hop}}
      /* Receiver has received the same broadcast message with larger hop-count */ then
        Receiver drops TREE_CNST; Receivers ← Receivers \ Receiver;
        goto child_node;
      end if
    end for
    for i=1 to 4 do
      if {{Receiver.$T_i$-ID == TREE_CNST.tree-ID} and {TREE_CNST.hop-count < Receiver.$T_i$-hop}}
      /* Receiver has received the same broadcast message with smaller hop-count */ then
        Receiver asks Receiver.$T_i$-parent to remove parent-link;
        call TREENODE($T_i$, Receiver, TREE_CNST);
        TREE_CNST.hop-count ← TREE_CNST.hop-count +1;
        TREE_CNST.sender ← Receiver;
        Receiver re-broadcasts TREE_CNST;
        Receivers ← Receivers \ Receiver; Receivers ← Receivers ∪ Receiver's Neighbors;
        goto child_node;
      end if
    end for
    for i=1 to 4 do
      if {Receiver.$T_i$-hop = ∞}/* Receiver receives the broadcast message the first time */ then
        call TREENODE($T_i$, Receiver, TREE_CNST);
        TREE_CNST.hop-count ← TREE_CNST.hop-count +1;
        TREE_CNST.sender ← Receiver;
        Receiver re-broadcasts TREE_CNST;
        Receivers ← Receivers \ Receiver; Receivers ← Receivers ∪ Receiver's Neighbors;
        goto child_node;
      end if
    end for
  end if
  out: return 0;
end for
```



(a) Discovery Area (b) Tree_Area and Discovery Area

Fig. 10. The dissemination network discovery area and Tree_Area.

see that they are not tree nodes. Among those dissemination nodes, type 1 dissemination nodes elect one among themselves as a reporter; otherwise a type 2 dissemination node will be elected as a reporter. The elected reporter immediately reports to the Source the event and then it informs sensor nodes in the target zone that it is a new reporter. At the same time, all sensor nodes will deactivate their timers and continue to sense their surroundings periodically. If the elected reporter is a type 1 dissemination node, it calls *TREE* Procedure 3 to construct a tree, rooted at itself, within its Tree_Area. Of course, no tree will be constructed if the elected reporter is a type 2 dissemination node.

3.3.4. *Target zone contains sensor nodes which are neither tree nodes nor dissemination nodes*

Receiving no INFORM messages at the end of the wait period, TIME3, sensor nodes in the target zone will elect among themselves a sensor node as a candidate of the Source or a reporter. The elected candidate broadcasts a message asking all neighboring sensor nodes to deactivate their timers. It then activates its timer and sets it to TIME4[5] which is the maximal time taken to determine if there exists a dissemination network. At the same time, the candidate invokes the dissemination network discovery process to see if there exists a dissemination network.

Dissemination network discovery process: A candidate or a sink, acted as an invoker, starts the dissemination network discovery process by broadcasting a DISSEM DISCOVERY message to sensor nodes inside its Discovery Area. A Discovery Area of an invoker is formed by (1) expanding a $d \times d$ inner square centered at the invoker outwards on each side by r_s and then (2) connecting four expanded sides by four quarter arcs of a circle of radius r_s with centers at the four corners of the inner square, as shown in Fig. 10(a). Only sensor nodes inside the Discovery Area do rebroadcasting. However, a sensor node stops rebroadcasting and sends a FOUND message back to the invoker if it is a tree node, a path node, or a dissemination node. As claimed by Proposition 16, this process can determine whether a dissemination network exists or not.

After completing the dissemination network discovery process, the role of the candidate can be determined:

- Once the candidate receives a FOUND message during the dissemination network discovery process, which means that a dissemination network exists, the candidate sets itself as a new reporter. Since the dissemination network exists, the reporter must belong to one and only one Referenced Tree_Area as shown in Fig. 10(b). By definition, this implies that there exists a type 1 dissemination

---

[5] The value of TIME4 depends on the size of the Discovery Area, message processing and delay at each sensor node.

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

node such that the reporter and the dissemination node are located in the same Tree_Area. Since the Reporter is neither a tree node nor a dissemination node, by Lemma 13, this type 1 dissemination node cannot be a tree root. The reporter will search through the Tree_Area its dissemination node. The found dissemination node will call *TREE* Procedure 3 to create a tree rooted at itself. Once the tree is created, the reporter then reports periodically an event through its parent node of the newly created tree to the Source. The reporter informs all sensor nodes in target zone that it is a new reporter. At the same time, it tells all sensor nodes to deactivate their timers and continue to sense their surroundings periodically.

- If a candidate has not been claimed to be a reporter after the broadcast stops (ACK or TIME4), it will be the Source. In this case, the Source then starts a dissemination network construction process as described in Section 3.1 and then inform all sensor nodes in target zone that it is a new reporter.

3.4. Query and response

When sending a QUERY message to the Source, a sink has different actions to take, depending on the scenarios it encounters, which are elaborated on as follows:

3.4.1. Sink's neighborhood contains the current IA

If the sink is still in the neighborhood of its Immediate Agent (IA), it periodically sends a QUERY message to the Source via the IA and waits for an ACK from the IA. The sink then waits for a Query response from the Source if it receives an ACK from its IA.

3.4.2. Sink's neighborhood contains tree nodes but not the current IA

If the sink has left the neighborhood of its IA or has not received an ACK from the IA after a wait time, TIMEa, which is slightly higher than $2 \times r_c$. Now, the sink resets its timer to TIMEa then broadcasts a request to see if there is any tree node in its one-hop neighborhood. The sink selects one from the tree-node responders as a new IA and its tree root as a new PA. At the same time, the sink informs its neighbors that a new IA has been selected. It then sends a QUERY message via the new IA towards the Source every query period. The new IA forwards the QUERY message through the new PA to the Source and replies with an ACK to the sink. If the previous IA and PA exist, the sink will ask the new PA to tell the previous PA that it is the new PA, and the old QUERY response will be forwarded to the new PA by the previous PA. An old IA will relinquish its role and change back to a regular sensor node if it has not received any QUERY message from its sink for some query intervals.

3.4.3. Sink's neighborhood contains neither tree nodes nor the current IA

If no response comes back after some more wait time, TIMEa, it means that there are no tree nodes in the one-hop neighborhood of the sink. The sink starts to request a near-by dissemination node to create a tree. The requested dissemination node invokes *TREE* Procedure 3 to construct a tree rooted at itself within its Tree_Area. Once the sink receives the first broadcast message TREE_CNST from the dissemination node, it waits for a certain period of time such that tree nodes in its neighborhood are stable. Then the sink will select a tree node in its neighborhood as the new IA and the dissemination node, the root of the constructed tree, as the new PA. Once the sink finds a new IA and PA, it does what it did in the previous case.

3.5. Network lifetime prolongation

The network lifetime prolongation can be achieved by either saving energy or slowing down energy consumption. TOG employs a traffic load reducing mechanism to save energy and a traffic load balancing mechanism to slow down energy consumption.

3.5.1. Reducing the traffic load

TTDD tries to aggregate queries to reduce the traffic load. However, it requires a dissemination node to wait and synchronize queries from different sinks. In reality, such a synchronization is difficult to achieve because a dissemination node does not know how long it needs to wait to aggregate queries from other sinks.

TOG employs a new query aggregation scheme to reduce the traffic load. It divides the time into a series of aggregation time periods. A dissemination node records then forwards a query when it receives the first query during an aggregation time period. When other queries arrive during the same aggregation time period, the dissemination node checks if it has already received the response of the first query within this aggregation time period. If the dissemination node has received the response of the first query, it sends the response to those sinks directly. Otherwise, it only records those queries and waits for the response of the first query to come back. It then sends the response to all queries. In addition, the query aggregation can also improve the query response time.

3.5.2. Balancing the traffic load

In TOG, some dissemination nodes and relay nodes may have more traffic load. As a result, they will inevitably consume more energy. To prolong the network lifetime, four balancing traffic load techniques listed in the following can be used to slow down energy consumption.

First, when a message needs to be forwarded, a dissemination node will select a path which is used most infrequently to forward it to balance the load. Second, when the Source finds its remaining energy reaching a certain limit, it will broadcast a request to dissemination nodes to select a dissemination node which has more energy left as a new source. In order for queries and reports to correctly be delivered to a new source, some directions of paths need to be reversed. The four quadrants of a new source and the four quadrants of the previous Source divide the field into nine regions (similar to a tic-tac-toe board). Directions of paths needed to be reversed are: the vertical directions of paths in the left, center, and right regions, and the horizontal directions of paths in the top, center, and bottom regions. Third, when the Source finds the remaining energy of most of the dissemination nodes reaching their limits, it will reconstruct a dissemination network by shifting the virtual grid vertically and horizontally away from the original virtual grid structure. Fourth, the remaining energy of relay nodes reaching their limits shall be replaced by low usage neighbors.

From the description above, it can be seen that most of the functions under TOG are distributed. They include constructing the dissemination network, creating a tree rooted at a dissemination node, monitoring targets, and prolonging the network lifetime. Notice that the Source in our application scenario issues an order to construct the dissemination network, stores all the target information sent by reporters, and responds to the requests sent by sinks. It is not hard to see that the nature of these functions done by the Source are naturally centralized. We all know that centralized functions may shorten the network lifetime. Therefore, it is a good idea to reduce centralized functions as much as possible. One thing we do is to provide a mechanism to select a new Source at a suitable time to reduce the load of the current Source to increase the network lifetime as described in the previous paragraph.

4. Simulation results

This section is dedicated to evaluating TOG's performance. We first list the default simulation settings in Table 1 and then describe the definitions of performance metrics. Those metrics are used to study TOG's performance by comparing it with those of TTDD [11], EEGBDD [1], and Lee's Protocol [4]. In the figure legends, the label "IGS" is used to represent Lee's Protocol.

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

Table 1
Default simulation settings.

Parameter	Value
Number of sinks	4
Number of targets	4
Sensor node deployment	Random
Simulation area	2000m × 2000m
Sink mobility model	Random waypoint model
Radio propagation model	Two-ray ground
Radio communication range	200m
Radio sensing range	100m
Sink speed	6m/s with 5s pause
Target speed	Stationary
Sink query interval	1 s
Simulation period	200 s
Transmission power	0.66W
Receiving power	0.395W
Data packet size	64 bytes
Control packet size	36 bytes
Number of sensor nodes	500

4.1. Simulation setting and metrics

The simulation work is done under NS2 [31] version 2.34 network simulator. We use 802.11 DCF as the MAC protocol which supports 1Mbps bandwidth. Each grid structure has its own grid cell size. The grid-cell areas of TOG and TTDD are 600 m × 600 m. However, grid-cell areas of EEGBDD and Lee's Protocol are 130 m × 130 m and 70 m × 70 m, respectively. The number k is set to be five as used by Lee's Protocol to construct the IGS structure. The random topology is generated by NS2 **setdest**. For each configuration, 25 network topologies are generated and simulation results are collected and averaged then shown in their respective figures.

The following are the definitions of performance metrics:

- The average total energy consumption is the average energy consumed by the sensor nodes in transmitting and receiving packets during the simulation period. Packets concerned here can be control packets or data packets. The unit of energy is Joule.
- The average success ratio is the average ratio between the number of data packets correctly delivered to the sink and the number of queries sent from a sink during the simulation period.
- The average end to end delay is the average time from the moment a sink sends a query to the moment the sink receives the query response.
- The average network lifetime is the average period of time from the beginning of a simulation until the first sensor node, which is commonly a sensor node in the hotspot area, runs out its energy. To evaluate this performance, each sensor node is assumed to have an initial electrical energy of 20 Joules.

4.2. Impact of number of sinks

We first evaluate the performance against the number of sinks. The number of sinks varies from 1, 2, 4, 6 to 8. The simulation results are shown in Fig. 11. Their statistical confidence information is collected in Table 2. As can be seen from that table, the 95% confidence intervals (CIs) of TOG do not overlap with those of TTDD, IGS, and EEGBDD on almost all performance metrics except the average success ratio. The 95% CIs of TOG do overlap with those of EEGBDD on the average success ratio performance metric. Therefore, we conclude that the measurement data of TOG are statistically significantly different from those of other protocols on almost all performance metrics except those of EEGBDD on the average success ratio.

The following may contribute to the better performance shown by TOG under the impact of the number of sinks. (1) TOG constructs only one grid network once the first target is detected and its associated source is selected. Other new sources, as new reporters, only need to send the collected information to the first source through a tree path. (2) During the grid network construction, routing paths between dissemination nodes are also created which result in reducing the number of sensor nodes involved in query and data forwarding, facilitating query routing decision, and reducing the number of packet collisions. Other possible reasons are as follows. (1) In TTDD, constructing a separate grid network for each individual source together with local query flooding may contribute to quite a portion of total energy consumption. TTDD also exhibits worse performance on other metrics. It may be due to the fact that more sensor nodes are involved in query and data forwarding. (2) EEGBDD creates more dissemination nodes than that of TOG because the cell size of EEGBDD is about one fifth of TOG's. When a dissemination node of EEGBDD forwards a packet to a designated dissemination node, all neighboring dissemination nodes still need to spend energy to receive the packet to determine whether they are the one designated to forward the query. (3) Lee's Protocol requires an even smaller cell size, about one half of EEGBDD's. Since k is five, the unstructured area is larger than a cell area of TOG. This could cause a lot of flooding activities occurring outside IGS. (4) Moreover, new sources of EEGBDD and Lee's Protocol other than the first one need to flood their information including locations to some sensor nodes and all sinks so that any sink can query any source.

4.3. Impact of sink mobility

In this section, we evaluate the performance at various sink speeds from 0m/s, 2m/s, 4m/s, 6m/s, 8m/s, to 10m/s. The simulation results are shown in Fig. 12. Their statistical confidence information is collected in Table 3. As can be seen from that table, we have the same conclusion as that of the previous one.

For all protocols, sinks need to keep tract of their trajectories once they move away from the current cell. Since the cell size of TOG is greater than EEGBDD's which is larger than that of Lee's Protocol, the chance of moving into a new cell of TOG is less than those of EEGBDD and Lee's Protocol. Therefore, TOG has less work than that of EEGBDD and Lee's Protocol when sinks are moving around. This may be the main reason that TOG has the better performance in the category.

4.4. Impact of network size

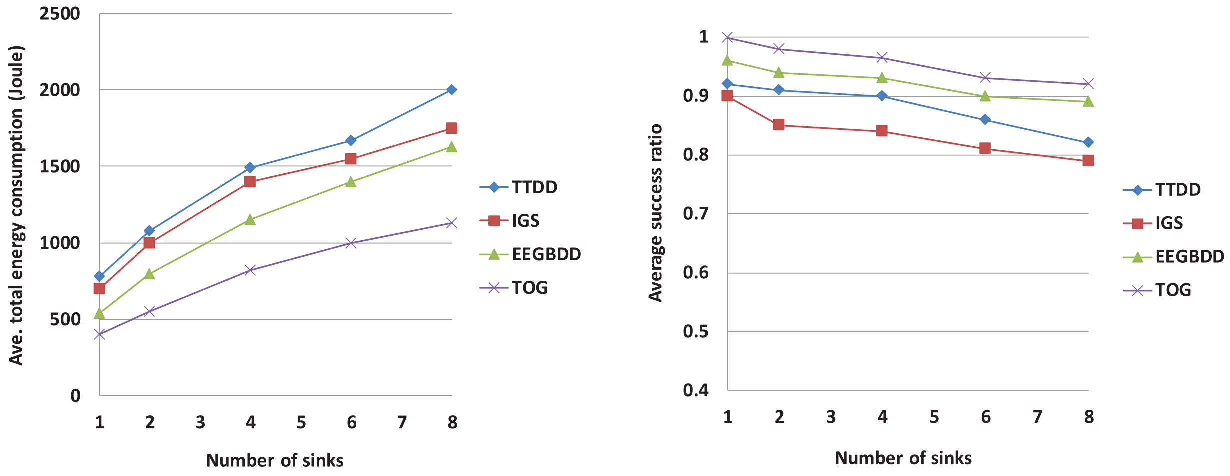
In this simulation, we evaluate the performance impact due to the network size assuming that the grid networks have been established. We assume that the number of sensor nodes deployed in the simulation area varies from 400, 500, 600, 700, to 800. The simulation results are shown in Fig. 13. Their statistical confidence information is collected in Table 4. As can be seen from that table, we have the same conclusion as that of the previous one.

It is not hard to see that the more sensor nodes involved in query and data forwarding, the more traffic load will be generated. Please note that energy consumed by constructing grid networks is not counted in this simulation. Since TOG has larger cell size, it has less dissemination nodes involved in query and data forwarding among all four protocols. Also, since routing paths exist in TOG, there is no need to do routing calculations as other protocols do. Those reasons may contribute to TOG's better performance behavior in this category.

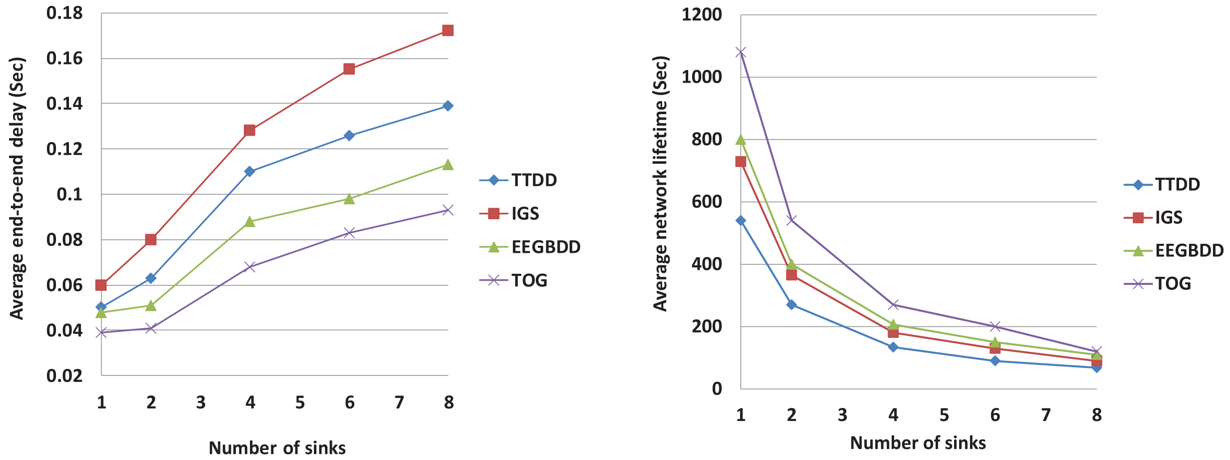
4.5. Impact of query interval

The performance against the query interval will be evaluated at 0.25 s, 0.5 s, 1 s, and up to 4 s. The simulation results are shown in Fig. 14. Their statistical confidence information is collected in Table 5. As can be seen from that table, the 95% CIs of TOG do not overlap with those of other protocols on almost all performance metrics except the average success ratio performance metric. The 95% CIs of

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015



(a) Energy Consumption (b) Success Ratio



(c) Delay (d) Network Lifetime

Fig. 11. Performance impact due to number of sinks.

Table 2
The 95% confidence intervals (CIs) associated with Fig. 11.

# of sinks	1	2	4	6	8
TTDD	[760.4,799.6]	[1055.8,1098.1]	[1460.9,1519.1]	[1637.1,1702.9]	[1960.8,2039.2]
IGS	[691.3,708.6]	[980.4,1019.6]	[1372.5,1427.4]	[1519.8,1580.2]	[1715.5,1784.5]
EEGBDD	[525.1,554.9]	[776.9,813.0]	[1130.6,1179.3]	[1372.5,1427.4]	[1597.8,1662.1]
TOG	[387.4,412.54]	[534.3,565.7]	[801.4,840.6]	[978.1,1021.9]	[1105.6,1154.3]

(a) Energy consumption's 95% CIs of Fig. 11(a).

# of sinks	1	2	4	6	8
TTDD	[0.883,0.956]	[0.874,0.946]	[0.864,0.935]	[0.826,0.894]	[0.787,0.852]
IGS	[0.864,0.935]	[0.816,0.883]	[0.807,0.873]	[0.778,0.842]	[0.759,0.821]
EEGBDD	[0.922,0.998]	[0.903,0.977]	[0.893,0.966]	[0.864,0.935]	[0.855,0.925]
TOG	[0.996,1.000]	[0.960,0.999]	[0.963,0.966]	[0.893,0.966]	[0.883,0.956]

(b) Success ratio's 95% CIs of Fig. 11(b).

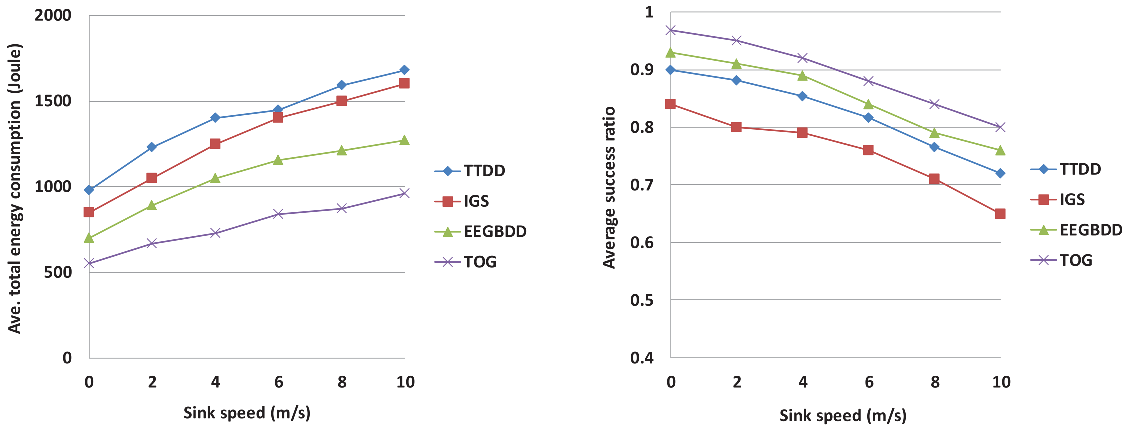
# of Sinks	1	2	4	6	8
TTDD	[0.047,0.053]	[0.059,0.067]	[0.104,0.115]	[0.119,0.132]	[0.132,0.146]
IGS	[0.057,0.062]	[0.076,0.084]	[0.121,0.134]	[0.147,0.163]	[0.163,0.180]
EEGBDD	[0.045,0.050]	[0.050,0.052]	[0.083,0.092]	[0.093,0.103]	[0.107,0.119]
TOG	[0.036,0.041]	[0.039,0.043]	[0.064,0.071]	[0.082,0.083]	[0.088,0.098]

(c) Delay's 95% CIs of Fig. 11(c).

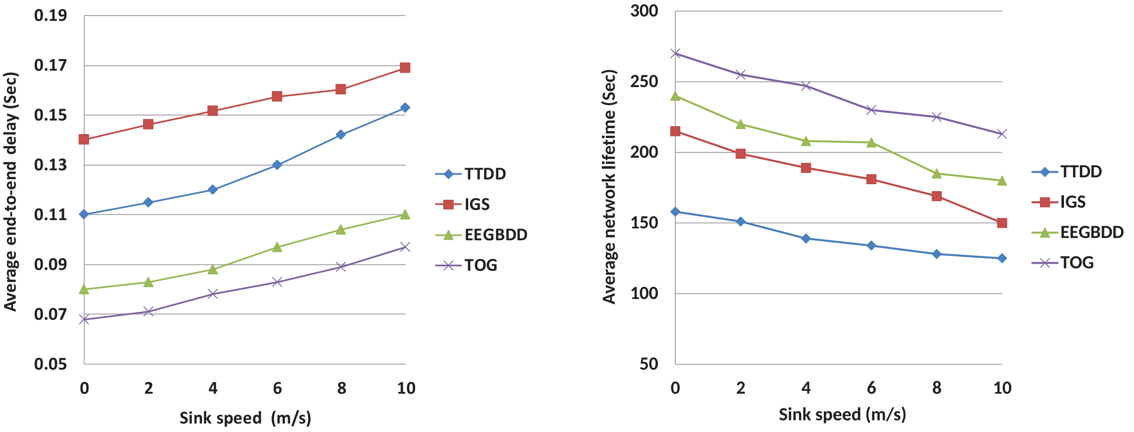
# of sinks	1	2	4	6	8
TTDD	[524.3,555.6]	[258.2,281.7]	[123.8,144.1]	[80.5,99.4]	[64.4,71.5]
IGS	[708.6,749.3]	[349.3,380.6]	[170.1,191.9]	[119.8,140.1]	[86.4,93.5]
EEGBDD	[778.1,821.9]	[382.7,417.2]	[194.4,219.5]	[138.6,161.3]	[106.1,113.9]
TOG	[1054.9,1105.1]	[521.1,558.8]	[256.6,283.3]	[187.4,212.5]	[114.1,125.8]

(d) Network lifetime's 95% CIs of Fig. 11(d).

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015



(a) Energy Consumption (b) Success Ratio



(c) Delay (d) Network Lifetime

Fig. 12. Performance impact due to sink mobility.

Table 3
The 95% confidence intervals (CIs) associated with Fig. 12.

Sink speed	0 m/s	2 m/s	4 m/s	6 m/s	8 m/s	10 m/s
TTDD	[931.0,1029.0]	[1171.2,1288.8]	[1331.4,1468.6]	[1378.2,1521.7]	[1511.6,1668.4]	[1597.6,1762.3]
IGS	[806.8,893.1]	[999.0,1100.9]	[1188.1,1311.9]	[1331.4,1468.6]	[1426.3,1573.6]	[1521.6,1678.4]
EEGBDD	[665.5,734.4]	[846.8,933.1]	[999.0,1100.9]	[1098.5,1211.4]	[1160.6,1259.3]	[1207.6,1332.3]
TOG	[518.6,581.3]	[638.6,701.3]	[694.3,765.6]	[798.8,881.1]	[826.88,913.1]	[912.9,1007.0]

(a) Energy consumption's 95% CIs of Fig. 12(a).

Sink speed	0 m/s	2 m/s	4 m/s	6 m/s	8 m/s	10 m/s
TTDD	[0.861,0.939]	[0.843,0.919]	[0.816,0.891]	[0.780,0.852]	[0.731,0.798]	[0.684,0.755]
IGS	[0.803,0.876]	[0.765,0.835]	[0.755,0.824]	[0.726,0.793]	[0.679,0.741]	[0.621,0.678]
EEGBDD	[0.889,0.970]	[0.870,0.950]	[0.851,0.929]	[0.803,0.876]	[0.755,0.824]	[0.726,0.793]
TOG	[0.937,0.999]	[0.908,0.991]	[0.879,0.960]	[0.841,0.918]	[0.803,0.876]	[0.765,0.834]

(b) Success ratio's 95% CIs of Fig. 12(b).

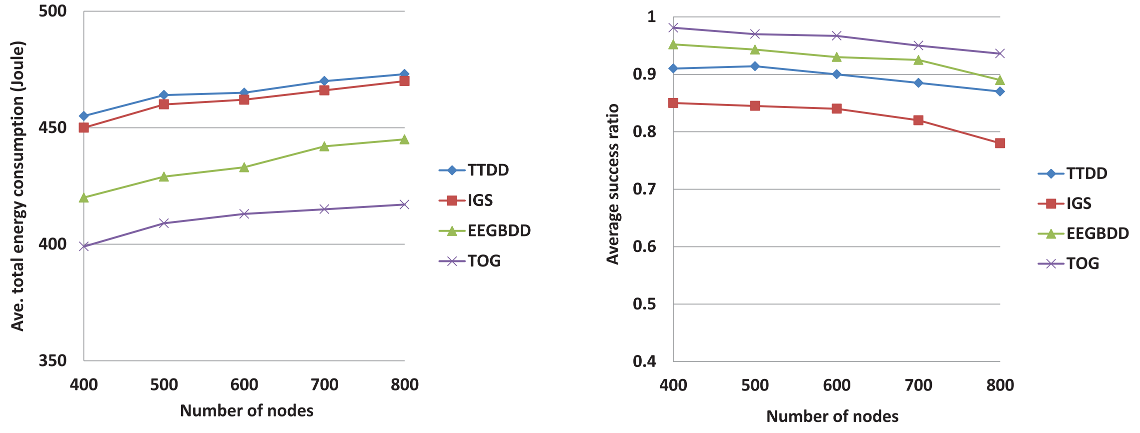
Sink speed	0 m/s	2 m/s	4 m/s	6 m/s	8 m/s	10 m/s
TTDD	[0.105,0.114]	[0.109,0.119]	[0.114,0.125]	[0.124,0.135]	[0.135,0.148]	[0.146,0.159]
IGS	[0.133,0.146]	[0.139,0.152]	[0.144,0.158]	[0.150,0.164]	[0.153,0.167]	[0.161,0.176]
EEGBDD	[0.076,0.083]	[0.079,0.086]	[0.084,0.091]	[0.093,0.101]	[0.099,0.108]	[0.105,0.114]
TOG	[0.065,0.071]	[0.068,0.073]	[0.074,0.081]	[0.079,0.086]	[0.085,0.092]	[0.092,0.102]

(c)Delay's 95% CIs of Fig. 12(c).

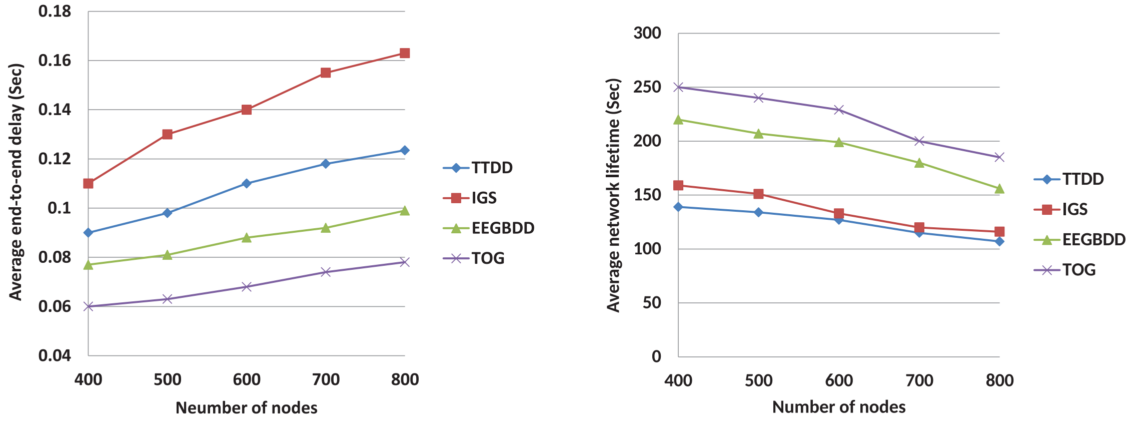
Sink speed	0 m/s	2 m/s	4 m/s	6 m/s	8 m/s	10 m/s
TTDD	[148.9,167.0]	[142.7,159.2]	[131.1,146.8]	[126.5,141.4]	[120.9,135.1]	[118.3,131.6]
IGS	[202.8,227.1]	[188.0,209.9]	[178.4,199.5]	[171.2,190.8]	[159.5,178.4]	[141.7,158.2]
EEGBDD	[226.6,253.3]	[207.8,232.1]	[196.2,219.7]	[195.6,218.3]	[174.8,195.1]	[170.2,189.8]
TOG	[254.7,285.2]	[240.8,269.1]	[233.2,260.7]	[217.1,242.9]	[212.4,237.5]	[201.2,224.7]

(d) Network lifetime's 95% CIs of Fig. 12(d).

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015



(a) Energy Consumption (b) Success Ratio



(c) Delay (d) Network Lifetime

Fig. 13. Performance impact due to network size.

Table 4
The 95% confidence intervals (CIs) associated with Fig. 13.

# of nodes	400	500	600	700	800
TTDD	[445.5,464.4]	[453.4,474.5]	[453.2,476.7]	[457.1,482.9]	[458.8,487.1]
IGS	[441.1,458.9]	[449.8,470.1]	[450.8,473.1]	[453.2,478.7]	[455.8,484.1]
EEGBDD	[412.3,427.6]	[420.1,437.8]	[423.0,443.1]	[430.8,453.1]	[432.6,457.3]
TOG	[391.3,406.6]	[400.1,417.8]	[403.3,422.6]	[403.8,426.1]	[404.6,429.3]

(a) Energy consumption's 95% CIs of Fig. 13(a).

# of nodes	400	500	600	700	800
TTDD	[0.870,0.949]	[0.874,0.953]	[0.861,0.939]	[0.846,0.923]	[0.832,0.907]
IGS	[0.812,0.887]	[0.808,0.881]	[0.803,0.876]	[0.784,0.855]	[0.746,0.813]
EEGBDD	[0.911,0.993]	[0.901,0.984]	[0.889,0.970]	[0.889,0.959]	[0.869,0.947]
TOG	[0.961,0.999]	[0.950,0.989]	[0.947,0.986]	[0.932,0.967]	[0.920,0.951]

(b) Success ratio's 95% CIs of Fig. 13(b).

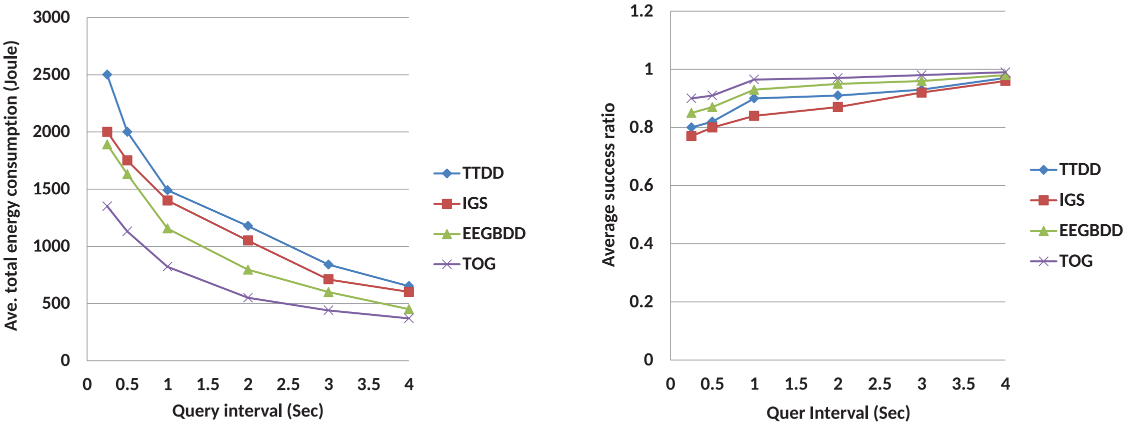
# of nodes	400	500	600	700	800
TTDD	[0.086,0.093]	[0.093,0.102]	[0.105,0.114]	[0.112,0.123]	[0.118,0.128]
IGS	[0.105,0.114]	[0.124,0.135]	[0.133,0.146]	[0.148,0.161]	[0.155,0.170]
EEGBDD	[0.073,0.080]	[0.077,0.084]	[0.084,0.091]	[0.088,0.095]	[0.094,0.103]
TOG	[0.057,0.062]	[0.060,0.065]	[0.065,0.070]	[0.070,0.077]	[0.074,0.081]

(c) Delay's 95% CIs of Fig. 13(c).

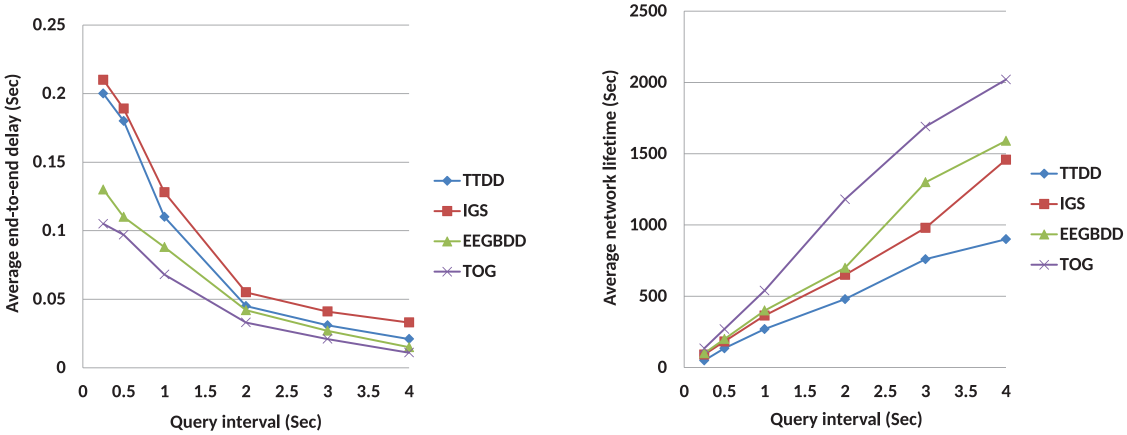
# of nodes	400	500	600	700	800
TTDD	[131.1,146.8]	[126.5,141.4]	[119.9,134.1]	[108.5,121.4]	[101.0,112.9]
IGS	[150.1,167.8]	[142.5,159.4]	[125.5,140.4]	[113.2,126.7]	[109.5,122.4]
EEGBDD	[207.6,232.3]	[195.4,218.5]	[187.8,210.1]	[169.9,190.1]	[147.2,164.7]
TOG	[236.0,263.9]	[226.5,253.4]	[216.1,241.8]	[188.8,211.2]	[174.6,195.3]

(d) Network lifetime's 95% CIs of Fig. 13(d).

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015



(a) Energy Consumption (b) Success Ratio



(c) Delay (d) Network Lifetime

Fig. 14. Performance impact due to query interval.

Table 5
The 95% confidence intervals (CIs) associated with Fig. 14.

Query interval	0.25 s	0.5 s	1 s	2 s	3 s	4 s
TTDD	[2360.0,2639.9]	[1887.1,2112.8]	[1405.7,1574.2]	[1111.1,1242.8]	[792.9,887.0]	[615.5,688.4]
IGS	[1887.8,2112.1]	[1652.0,1848.0]	[1321.6,1478.4]	[991.2,1108.8]	[670.0,749.9]	[567.3,634.6]
EEGBDD	[1784.1,1995.8]	[1538.7,1721.2]	[1090.3,1219.6]	[750.5,839.4]	[566.4,633.5]	[424.9,475.1]
TOG	[1274.4,1425.6]	[1066.7,1193.2]	[775.0,866.9]	[519.2,580.7]	[415.3,464.6]	[349.3,390.6]

(a) Energy consumption's 95% CIs of Fig. 14(a).

Query interval	0.25 s	0.5 s	1 s	2 s	3 s	4 s
TTDD	[0.765,0.834]	[0.784,0.855]	[0.861,0.939]	[0.870,0.949]	[0.890,0.969]	[0.927,1.000]
IGS	[0.736,0.834]	[0.765,0.834]	[0.803,0.876]	[0.832,0.907]	[0.879,0.960]	[0.918,1.000]
EEGBDD	[0.812,0.887]	[0.832,0.907]	[0.889,0.970]	[0.908,0.991]	[0.918,1.000]	[0.937,1.000]
TOG	[0.861,0.939]	[0.870,0.949]	[0.922,1.000]	[0.927,1.000]	[0.937,1.000]	[0.946,1.000]

(b) Success ratio's 95% CIs of Fig. 14(b).

Query interval	0.25 s	0.5 s	1 s	2 s	3 s	4 s
TTDD	[0.191,0.208]	[0.172,0.187]	[0.105,0.114]	[0.043,0.046]	[0.029,0.032]	[0.020,0.022]
IGS	[0.201,0.219]	[0.181,0.197]	[0.122,0.133]	[0.052,0.057]	[0.039,0.042]	[0.031,0.034]
EEGBDD	[0.124,0.135]	[0.105,0.114]	[0.084,0.091]	[0.040,0.043]	[0.025,0.028]	[0.014,0.015]
TOG	[0.100,0.109]	[0.092,0.101]	[0.065,0.071]	[0.031,0.034]	[0.020,0.021]	[0.011,0.011]

(c) Delay's 95% CIs of Fig. 14(c).

Query interval	0.25 s	0.5 s	1 s	2 s	3 s	4 s
TTDD	[47.2,52.7]	[127.5,142.4]	[254.9,285.1]	[453.1,506.8]	[717.4,802.5]	[849.6,950.3]
IGS	[85.9,96.1]	[172.8,193.1]	[344.6,385.3]	[613.6,686.3]	[925.1,1034.8]	[1376.4,1539.5]
EEGBDD	[94.3,105.6]	[188.8,211.1]	[377.6,422.3]	[660.8,739.2]	[1227.2,1372.7]	[1501.0,1678.9]
TOG	[127.5,142.4]	[254.9,285.1]	[509.8,570.1]	[1113.9,1246.1]	[1595.5,1784.4]	[1906.8,2133.1]

(d) Network lifetime's 95% CIs of Fig. 14(d).

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

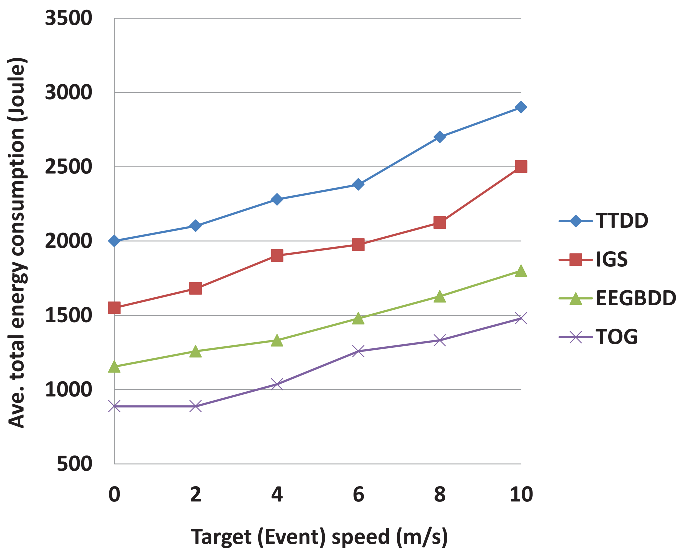


Fig. 15. Energy consumption vs. target speed.

TOG do overlap with those of EEGBDD on the average success ratio all query intervals. However, they do overlap with those of TTDD, IGS, and EEGBDD on the average success ratio at query intervals, which are greater than 2 s. Therefore, we conclude that the measurement data of TOG are statistically significantly different from those of other protocols on almost all performance metrics except those of TTDD, IGS, and EEGBDD on the average success ratio at certain query intervals.

The shorter query period will result in more control and data packets generated in the network. It can be treated as more sinks in the network. It looks like that the performance impact under the query interval may be treated the same as that of the number of sinks as discussed in Section 4.2.

4.6. Impact of target mobility

Now, we evaluate the performance at various target speeds from 0m/s, 2m/s, 4m/s, …, up to 10m/s. TTDD, IGS, and EEGBDD only briefly mention that how target (or event) mobility can be supported. Since they do not provide the detailed implementation, we only evaluate energy consumption performance for this category. The simulation results are shown in Fig. 15. Their statistical confidence information is collected in Table 6. As can be seen from that table, the 95% CIs of TOG do not overlap with those of TTDD, IGS, and EEGBDD on all performance metrics. Therefore, we conclude that the measurement data of TOG are statistically significantly different from those of other protocols on all performance metrics.

TTDD, Lee's Protocol, and EEGBDD handle target mobility differently. TTDD treats a new source associated with a moving target differently. No new grid network will be constructed for such a new source. It applies the same technique used by a sink to handle target mobility. If a new source has data to send, it locally floods a packet to locate an immediate dissemination node. In Lee's Protocol, if a target leaves the sensing area of its original source, a new source will be selected. The selected new source acts as a regular source. It needs to send its information to all sensor nodes and sinks. EEGBDD handles target mobility in the same way as that of Lee's Protocol. TOG applies the same technique used by its sinks to handle target mobility. This may explain why TOG performance behavior is better than others' in this category.

5. Conclusions and future studies

Mobility of target and sink brings challenges in WSNs. In particular, a target detection application needs a reliable data dissemination scheme. The previous solutions to target detection are inefficient due to skewed longer data dissemination paths, local query flooding, and new sources information flooding. Mobility makes them infeasible.

In this study, we propose a novel data dissemination framework, which is called Tree Overlay Grid, to handle mobile target detections when multiple mobile sinks appear in wireless sensor networks. A grid network, which is built on top of the structure, helps distribute traffic flow along the grid line. Trees, which are constructed at the bottom grid cells, are used to collect information of mobile targets and sinks efficiently. In addition, two mechanisms are introduced to prolong the network lifetime. First, data aggregation is implemented to lower a traffic load to save energy. Second, four techniques are implemented to balance a traffic load to slow down energy consumption. The simulation results validate that our proposed framework consumes less total energy, performs well on the query and response work, and has a longer network lifetime among all.

When collecting performance data by implementing TOG on real sensor nodes, we need to pay a special attention to batteries. Since batteries for sensor nodes are designed to run for years, it is not possible to implement TOG on real sensor nodes to measure the true network lifetime. As we all experienced, some batteries' drain rates may be faster than others' even though they operate under the exact same conditions. Nguyen et al. [32] pointed out that a sensor node with a half dead battery may still be able to transmit messages, but not be able to receive any messages. Based on the two facts above, the collected measurement data, especially for success ratios, may not be correct if the implemented network has included some nodes with such batteries.

Proposition 1 in Section 3.1.3 implies that given a convex area covered by sensor nodes, a dissemination network constructed by *GFB* is actually a grid-like network. Namely, the inner area away from the boundary the grid lines are formed by grid strips R, but the area around the boundary the grid lines are formed by grid-like strips R' such that all directed paths of the dissemination network are located inside strips, R and R'. Please refer to Section 3.1 for the definitions of R and R'. Proposition 1 can be easily derived from the proved facts in Appendix A. It seems that *GFB* produces an optimal grid-like dissemination network in the sense that *GFB* uses the least amount of sensor nodes to form the dissemination network based on the selected grid structure. It is an interesting future study to prove or disprove *GFB* as an optimal algorithm. Another interesting future study would be whether we can always find an algorithm to construct a grid-like dissemination network of any area covered by sensor nodes such that all directed paths are located in either R or R'.

Appendix A. Facts related to dissemination network construction

The following derived facts show us that directed paths, which are constructed in Section 3, are indeed located inside narrow strips

Table 6
The 95% confidence intervals (CIs) associated with Fig. 15.

Sink speed	0 m/s	2 m/s	4 m/s	6 m/s	8 m/s	10 m/s
TTDD	[1888.0,2111.9]	[1984.4,2219.6]	[2152.3,2407.6]	[2246.7,2513.2]	[2548.8,2851.2]	[2737.7,3062.2]
IGS	[1463.3,1636.6]	[1585.9,1774.1]	[1795.4,2008.5]	[1856.8,2077.1]	[2005.2,2242.7]	[2360.1,2639.9]
EEGBDD	[1090.3,1219.6]	[1187.5,1328.4]	[1257.5,1406.4]	[1397.1,1562.8]	[1536.8,1719.1]	[1699.2,1900.4]
TOG	[837.8,938.1]	[838.6,937.3]	[977.9,1094.0]	[1187.5,1328.4]	[1257.4,1406.6]	[1397.3,1562.8]

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

along grid lines. They are derived from assumptions made at the beginning of Section 3.

Lemma 2. *If a point P inside the monitored field is covered by a sensor node, say X, then X can select from its one-hop neighbors including itself a sensor node N with the least ID such that N is closest to P.*

Proof. Since sensor node X covers the point P, sensor nodes closest to P must be located in the closed disk of $C_d(P)$ which is contained in $C_{r_s}(P)$, where d is the distance between X and P. Since $d + r_s < 2r_s = r_c$, $C_{r_s}(P) \subseteq C_{r_c}(X)$. Namely, all sensor nodes closest to P must be one-hop neighbors of X. Since each sensor node knows its one-hop neighbors, sensor node X can select from its one-hop neighbors including itself a sensor node which is closest to P. The sensor node N with the least ID wins the tie-break and achieves the uniqueness. □

Lemma 3. *If a point P is inside the monitored field and a sensor node X is on the circumference of $C_{r_s}(P)$, then there exists at least one sensor node covering P and sensor nodes covering P must be one-hop neighbors of X.*

Proof. Since P is a monitored point, there must be a sensor node, say Y, which covers it. Since the distance between X and P is r_s, and $2r_s = r_c$, $C_{r_s}(P)$ is inside $C_{r_c}(X)$, the neighborhood of X. Including Y, all sensor nodes covering P must be inside $C_{r_s}(P)$. Therefore, we conclude that there are sensor nodes covering P and all those sensor nodes are one-hop neighbors of X. □

Lemma 4. *If a point P is outside the monitored field and N is a sensor node having the shortest distance to P, then $C_{r_s}(N)$ contains at least two monitored boundary points.*

Proof. Since $C_{r_s}(N)$, a sensing area of N, is relatively small compared to the monitored field and N is inside the field, $C_{r_s}(N)$ is either fully contained inside the monitored field or intersects the monitored boundary at more than one point. If $C_{r_s}(N)$ was contained inside the monitored field, let X be the intersection point between the circumference of $C_{r_s}(N)$ and the line segment connecting P and N. Since X is on the circumference of $C_{r_s}(N)$, it is inside the monitored field. There must exist a sensor node, say N', inside $C_{r_s}(X)$ to cover X. We obtain a contradiction that N' is a sensor node closer to P than N is. □

Theorem 5. *Given a regular strip $R(\widetilde{SD})$ or $R'(\widetilde{SD})$, a directed path starting at any sensor node in $C_{r_s}(S)$ and ending at a sensor node covering D can always be constructed by GFB. The constructed path is located inside the regular strip R or R′.*

Proof. Let A_1 be any sensor node inside $C_{r_s}(S)$ and be the first path-node. Since S is in the monitored field, A_1 exists. If $C_{r_s}(A_1)$ contains the point D, we are done; otherwise, $C_{r_s}(A_1)$ may intersect the curve $\widetilde{SD}$ at one or two points. Let $A_1{}'$ be the intersection point which is closer to D. By Lemma 3 sensor node A_1 can select from R or R' its one-hop neighbors a sensor node, say A_2 such that the circle $C_{r_s}(A_2)$ may intersect $\widetilde{SD}$ at a point $A_2{}'$ whose curve length to D is the shortest and shorter than that from $A_1{}'$. Therefore, we conclude that a path from A_1 to A_2 located inside R or R' can be constructed by *GFB* and A_2 is closer to D than A_1 is. Continue this process. As can be seen that each time a new path-node, say N, is chosen, one intersection point of $C_{r_s}(N)$ and $\widetilde{SD}$ has shorter curve length to D. Therefore, the newly selected path-node will progress towards D and eventually cover D. □

Corollary 6. *If necessary, the last path-node of the directed path created in Theorem 5 can be used to select a sensor node with the least ID such that the selected sensor node is closest to point D.*

Proof. Let L be the last path-node which contains the point D. By Lemma 2, sensor node L can select from its one-hop neighbors including itself a sensor node with the least ID, say N, which is closest to D. □

Corollary 7. *Any sensor node inside regular strip R or R′ defined at the beginning of Section 3 broadcasts with the restriction that only sensor nodes inside R or R′ do re-broadcast. All sensor nodes inside R or R′ will receive the broadcast.*

Proof. It is easy to see that by modifying the proof of the Theorem 5 a little bit, we can claim that there exists a communication path between any two sensor nodes inside strip R or R' such that the communication path is located inside its regular strip. □

Corollary 8. *Corollary 7 holds even when the long side curves of the strip R or R′ are crossed by the monitored boundary.*

Corollary 9. *Let $R'(\widetilde{SD})$ be a regular strip, and P be a point outside the monitored field. The angle between $\overrightarrow{SD}$ and $\overrightarrow{SP}$ is less than 90°. A sensor node N with the least ID can be found inside R′ such that N is shortest to P. Furthermore, a directed path from a sensor node covering S to N can be constructed by GFB such that the constructed path is located inside R′.*

Proof. Since S is in the monitored field, there exists a sensor node, say T, covering S. Let m be $dist(T, P)$, the distance between T and P. Let A and B be intersection points of the circumference of $C_m(P)$ and the monitored boundary curve. Let N be the shortest sensor node to P and have the least ID. It is clear that this unique sensor node N is in the closed area of $C_m(P)$. By Lemma 4, N is inside $R'(\widetilde{AB})$. By Corollary 7, we can locate the unique sensor node N. The uniqueness is guaranteed by the least ID tie-break. As mentioned before, by modifying the proof of Theorem 5, a directed path from a sensor node T covering S to N can be constructed such that the path is located inside R'. □

Appendix B. Facts related to TREE Procedure 3

In this appendix, we show that *TREE* Procedure 3 indeed produces trees.

Lemma 10. *Any sensor node (x_1, y_1), in the Referenced Tree_Area (Fig. 9) of a referenced grid point P_r, will receive the broadcast message sent by a type 1 dissemination node $N_d = (x_d, y_d)$ of P_r with the restriction that only the sensor nodes inside the Tree_Area rebroadcast.*

Proof. First, we assume $N_d = (x_d, y_d)$, as shown in Fig. 9(a), is inside the Referenced Tree_Area of P_r. **Case(a):** if $|x_1 - x_d| < 2r_s$ or $|y_1 - y_d| < 2r_s$, then we can construct a vertical or a horizontal strip R such that the strip R is located inside the Tree_Area and contains both (x_1, y_1) and (x_d, y_d). By Corollary 7 or Corollary 8 depending on whether the Referenced Tree_Area is crossed by the monitored boundary, sensor node (x_1, y_1) will receive the broadcast message sent by sensor node (x_d, y_d). **Case(b):** if $|x_1 - x_d| \geq 2r_s$ and $|y_1 - y_d| \geq 2r_s$, then we can construct two disjoint vertical strips R_1 and R_2 and two disjoint horizontal strips R_3 and R_4 such that all four strips are located inside the Tree_Area. The strips R_1 and R_3 contain (x_1, y_1) and strips R_2 and R_4 contain (x_d, y_d), respectively. Since the monitored field is convex and bigger than a grid cell, either $R_1 \cap R_4$ or $R_2 \cap R_3$ intersects the monitored field. And this intersection must contain at least one sensor node since it is a square of size $2r_s \times 2r_s$. Again, by Corollary 7 or Corollary 8, sensor node (x_1, y_1) will receive the broadcast message sent by sensor node (x_d, y_d).

Now, if $N_d = (x_d, y_d)$, as shown in Fig. 9(b), it is outside the Referenced Tree_Area. Let R' be its associated boundary strip containing the dissemination node (x_d, y_d) and some sensor nodes which are inside the Referenced Tree_Area. By Corollary 7, those sensor nodes in the Referenced Tree_Area will receive a broadcast sent by dissemination node, (x_d, y_d). □

It is not hard to see the following proposition holds, and it can be used to find a dissemination node of a given referenced grid point.

Proposition 11. *Let $N_d = (x_d, y_d)$ be the type 1 dissemination node of a referenced grid point P_r. $N_d = (x_d, y_d)$ will receive the broadcast*

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

message sent by any sensor node located inside the Referenced Tree_Area of P_r, with the restriction that only the sensor nodes inside the Tree_Area rebroadcast.

Corollary 12. *TREE Procedure 3, which is called by a type 1 dissemination node N_d, creates a tree in its Tree_Area and any sensor node in the corresponding Referenced Tree_Area must be a tree node of the tree.*

Proof. By Lemma 10, any sensor node in the Referenced Tree_Area will receive the broadcast sent by a tyype 1 dissemination node N_d. Therefore, by Line 29 to Line 35 of TREE Procedure 3, all sensor nodes in the Referenced Tree_Area will call TREENODE Subroutine 2 to create tree nodes. It is easy to see that those sensor nodes and the root, N_d, are connected. Let T be this connected set. Each sensor node except N_d in T has a parent node. Assume sensor nodes C and F_1 are in T, and F_1 is a parent of C. If sensor node $F_2 \in T$ were another parent of C, C might receive TREE_CNST from sensor node F_2. If the hop-count carried by TREE_CNST is higher than Tree-hop of C, then the program will go through Line 15 to Line 20 of TREE Procedure 3. Therefore, sensor node F_2 cannot be a parent of C. If the hop-count in TREE_CNST is less than Tree-hop of C, then the program will go through Line 21 to Line 28 of TREE Procedure 3. In this case, the parent F_1 will be removed and replaced by F_1. This means the connected set T contains no cycles. Now it is clear that *TREE* Procedure 3 indeed creates a tree in a Tree_Area.

By Lemma 10 any sensor node in the Referenced Tree_Area will receive the broadcast sent by the dissemination node N_d. Also, by TREE Procedure 3 any sensor node in the Referenced Tree_Area will call TREENODE Subroutine 2 to create a tree node. Therefore, any sensor node in the Referenced Tree_Area is a tree node of the tree rooted at this dissemination node N_d. □

Lemma 13. *A reporter is either a dissemination node or a tree node of a tree created by TREE Procedure 3.*

Proof. Under the scenario described in Section 3.3.3, the reporter is a dissemination node. By the scenario described in Section 3.3.4, a type 1 dissemination node and the reporter can communicate with each other. Namely, if the type 1 dissemination node is a tree root then the reporter would be a tree node. By Corollary 12, there is a tree rooted at a type 1 dissemination node if the dissemination node exists. □

Appendix C. Facts related to dissemination network discovery process

Now, we show that the dissemination network discovery process can indeed be used to determine whether a dissemination network exists in a wireless sensor network.

Lemma 14. *All sensor nodes in the Discovery Area, which is described in Section 3.3.4, shall receive the DISSEM DISCOVERY message sent by the candidate when the candidate invokes the dissemination network discovery process.*

Proof. Let D be any sensor node inside the Discovery Area, and S be the candidate. If D is inside the inner square, the curve $\widetilde{SD}$ is the line segment connecting S and D. It is easy to see that the monitored area of the strip $R(\widetilde{SD})$ is entirely located inside the Discovery Area. By Theorem 5, the sensor node D will receive the broadcast sent by the candidate. If the sensor D is outside the inner square, rename it as D'. The new D is redefined as an intersection point between the circle $C_{r_s}(D')$ and the sides of the inner square. The curve $\widetilde{SD}$ is defined as the line segment between S and the new D. Again, the monitored area of the strip $R(\widetilde{SD})$ is entirely located inside the Discovery Area. By Theorem 5, we can find a communication path in strip R starting at S and ending at a sensor node located inside a circle center at D with radius r_s. Since the distance between the sensor node D' and the new point D is one half communication range, the sensor node D' can receive the broadcast sent by the candidate. □

Lemma 15. *If there are neither tree nodes nor path nodes in a Discovery Area of a candidate, then there does not exist any dissemination network.*

Proof. If there existed a dissemination network, the candidate of the Discovery Area must be located inside a unique grid cell. Let C_g be such a grid cell. Since C_g contains the candidate, a sensor node of the network, by the definition of the dissemination node, four dissemination nodes and their corresponding directed paths must exist. Let P_r be a referenced grid point of C_g and inside the inner square of the Discovery Area. If it is located inside the monitored field, the dissemination node of P_r is located inside $C_{r_s}(P_r)$, which is clearly located inside the Discovery Area. Now, assume P_r is located outside the monitored field. Let $C_r(P_r)$ be a circle centered at P_r. It is tangent to the monitored boundary at a point, N. Since the monitored field is convex, it is not hard to see that N is inside the square of the Discovery Area. Therefore, the dissemination point of P_r is located inside the Discovery Area. For both cases, there are contradictions in that there are no path nodes in the Discovery Area. In other words, we conclude that if there are neither tree nodes nor path nodes in a Discovery Area, then there does not exist any dissemination network. □

Proposition 16. *A dissemination network discovery process, which is described under Section 3.3.4, can determine whether a dissemination network exists or not.*

Proof. If the candidate receives a FOUND message then either a tree node or a path node is found in the Discovery Area. If the process is stopped and no FOUND message is received by the candidate, by Lemma 14, it means that there are neither tree nodes nor path nodes in the Discovery Area. Therefore, by Lemma 15, we conclude that there is no dissemination network for this sensor network. □

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.comcom.2015.08.015 .

References

[1] P. Singh, R. Kumar, V. Kumar, An energy efficient grid based data dissemination routing mechanism to mobile sinks in wireless sensor network, in: Proceedings of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014, pp. 401–409.

[2] H.S. Kim, T.F. Abdelzaher, W.H. Kwon, Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks, in: Proceedings of the In ACM Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 193–204.

[3] X. Wang, J. Wang, Y. Xu, Data dissemination in wireless sensor networks with network coding, EURASIP J. Wirel. Commun. Netw. 2010 (2010) 14. Article ID 465915

[4] E. Lee, S. Park, H. Park, Y. Yim, S. Kim, Data dissemination protocol based on independent grid structure in wireless sensor networks, in: Proceedings of the In the 22nd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2011, pp. 2264–2268.

[5] K. Kweon, H. Ghim, J. Hong, H. Yoon, Grid-based energy-efficient routing from multiple sources to multiple mobile sinks in wireless sensor networks, in: Proceedings of the International Symposium on Wireless Pervasive Computing (ISWPC), 2009, pp. 1–5.

[6] M. Guerroumi, N. Badache, S. Moussaoui, Networked digital technologies, in: Sink Mobile for Efficient Data Dissemination in Wireless Sensor Networks, in: Communications in Computer and Information Science, Springer Berlin Heidelberg, 2012, pp. 635–645.

[7] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, TTDD: a two-tier data dissemination model for large-scale wireless sensor networks, in: Proceedings of the 8th annual International Conference on Mobile Computing and Networking (MobiCom), 2002, pp. 148–159.

[8] E.B. Hamida, G. Chelius, Strategies for data dissemination to mobile sinks in wireless sensor networks, IEEE Wirel. Commun. 15 (6) (2008) 31–37.

[9] C.J. Lin, P.L. Chou, C.F. Chou, HCDD: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink, in: Proceedings of the 2006 international conference on Wireless communications and mobile computing (IWCMC), 2006, pp. 1189–1194.

[10] C. Chen, J. Ma, J. Salomaa, Simulation study of cluster based data dissemination for wireless sensor networks with mobile sinks, in: Proceedings of the 10th International Conference on Advanced Communication Technology, (ICACT), 2008, pp. 231–236.

[11] S. Sharma, S.K. Jena, Data dissemination protocol for mobile sink in wireless sensor networks, J. Comput. Eng. 2014 (2014) 10. Article ID 560675

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015

[12] Y. Faheem, S. Boudjit, K. Chen, Data dissemination strategies in mobile sink wireless sensor networks: a survey, in: Proceedings of the 2009 2nd IFIP Wireless Days (WD), 2009, pp. 1–6.

[13] M. Mauve, J. Widmer, H. Hartenstein, A survey on position- based routing in mobile ad-hoc networks, IEEE Netw. 15 (6) (2001) 30–39.

[14] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, Z.M. Wang, controlled sink mobility for prolonging wireless sensor networks lifetime, Wirel. Netw. (2008) 831–858.

[15] E. Lee, S. Park, F. Yu, Y. Choi, M.-S. Jin, S.-H. Kim, A predictable mobility-based data dissemination protocol for wireless sensor networks, in: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, 2008, pp. 741–747.

[16] K. Li, C.-C. Shen, G. Chen, Energy-constrained bi-objective data muling in underwater wireless sensor networks, in: Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2010, pp. 332–341.

[17] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet, SIGOPS Oper. Syst. Rev. 36 (5) (2002) 96–107.

[18] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, P. McDonald, When tcp breaks: delay- and disruption- tolerant networking, IEEE Internet Comput. 10 (4) (2006) 72–78.

[19] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, J. Porter, Luster: wireless sensor network for environmental research, in: Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems, 2007, pp. 103–116. Nov.

[20] Z.J. Haas, J.Y. Halpern, L. Li, Gossip-based ad hoc routing, IEEE/ACM Trans. Netw. (TON) 14 (3) (2006) 479–491.

[21] P.-S. Wang, G. Chen, M.-T. Sun, Energy-efficient wireless sensor networks for mobile target detections, in: Proceedings of the 41st International Conference on Parallel Processing Workshops, 2012, pp. 480–487.

[22] Y. Wang, J. Gao, J.S.B. Mitchell, Boundary recognition in sensor networks by topological methods, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Sep., 2006, pp. 122–133.

[23] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, Global Positioning System: Theory and Practice, 4th ed., Springer Verlag, 1997.

[24] X. Ji, H. Zha, Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling, in: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Mar., 2004, pp. 2652–2661.

[25] L. Lazos, R. Poovendran, Serloc: robust localization for wireless sensor networks, ACM Trans. Sensor Netw. 1 (1) (Aug. 2005) 73–100.

[26] G. Mao, B. Fidan, B.D.O. Anderson, Wireless sensor network localization techniques, Comput. Netw. (2007) 2529–2553.

[27] S. Delaet, P.S. Mandal, M.A. Rokicki, S. Tixeuil, Deterministic secure positioning in wireless sensor networks, in: Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS), Jun. 2008, pp. 469–477.

[28] J. Garcia-Alfaro, M. Barbeau, E. Kranakis, Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes, Ann. Telecommun. 66 (9-10) (2011) 535–552.

[29] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG), Aug., 1999, pp. 51–54.

[30] F. Willson, Theoretical and Practical Graphics, BiblioBazaar, LLC, 66.

[31] T. Issariyakul, E. Hossain, Introduction to Network Simulator NS2,, second ed., Springer, 2011.

[32] H.A. Nguyen, A.F. Forster, D. Puccinelli, S. Giordano, Sensor node lifetime: an experimental study, in: Proceedings of the 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011, pp. 202–207.

Please cite this article as: G. Chen et al., Energy-efficient mobile targets detection in the presence of mobile sinks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.015