J Supercomput (2013) 66:35–56
DOI 10.1007/s11227-013-0955-2

Hierarchical role-based data dissemination in wireless sensor networks

Chen-Che Huang · Tsun-Tse Huang · Jiun-Long Huang · Lo-Yao Yeh

Abstract Data dissemination from multiple sources to mobile sinks is fundamental and challenging in WSN applications due to limited energy supply of sensor nodes and sink mobility. Previous data dissemination protocols either rely on an energy-consuming coordinate system or build an inefficient backbone. In this paper, we propose a hierarchical role-based data dissemination (HRDD) protocol in wireless sensor networks. In HRDD, a small number of sensor nodes are assigned to serve as cluster heads and agents to form the data dissemination backbone and mitigate unnecessary query forwarding. In addition, HRDD designs an efficient data delivery mechanism that provides shorter paths to accelerate data delivery as well as reduce the number of data transmissions. An adaptive backbone maintenance mechanism is also introduced for low-energy cluster heads and agents to reduce their load, thereby prolonging the network lifetime. The experimental results show that HRDD achieves the longer network lifetime, the shorter delay, and the high success ratio compared to the prior work.

Keywords Data dissemination · Mobile sinks · Energy efficiency · Wireless sensor networks

---

C.-C. Huang · T.-T. Huang · J.-L. Huang (✉)
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, ROC
e-mail: jlhuang@cs.nctu.edu.tw

C.-C. Huang
e-mail: cchuang.cs95g@nctu.edu.tw

T.-T. Huang
e-mail: tshuang@cs.nctu.edu.tw

L.-Y. Yeh
System and Network Technology Division, National Center for High-Performance Computing,
Hsinchu, Taiwan, ROC
e-mail: lyyeh@nchc.narl.org.tw

1 Introduction

In recent years, wireless sensor networks (WSNs) have received a significant amount of research interests [10, 16, 26, 30, 37] since WSNs can be applied to a wide variety of applications [2]. A wireless sensor network usually consists of a large number of sensor nodes where each sensor node is a low-cost, tiny, battery-operated device with sensing, processing, and wireless communication capabilities. Examples of WSN applications include military surveillance [38], intrusion detection [34], target tracking [33], and so on. Data dissemination from multiple sources to multiple mobile sinks is a basic and important operation in these WSN applications. A source refers to a sensor node that detects an event of interest and generates the data message of the event for the sinks. A sink is a user that queries the sensor network about the event of interest. A typical example of data dissemination is that a number of moving soldiers (mobile sinks) submit queries about the up-to-date information of the enemy tanks detected by the sensor nodes (sources) deployed in a battlefield.

The naive approach to data dissemination is that a query sink floods the query message to the entire sensor network and the sources receiving the query message reply the desired data to the query sink. However, since the sensor nodes are energy-constrained, the flooding approach is impractical in sensor networks. To address this problem, [9, 12, 22, 25, 28, 29, 32, 36] have proposed the rendezvous area concept on top of a coordinate system. These protocols assumed that each sensor node is aware of its location by either equipping with a GPS device or employing a virtual coordinate system. Over the physical or virtual coordinate system, a group of sensor nodes are selected to form a rendezvous area for data dissemination. Each source reports the detected event to one or more sensor nodes in the rendezvous area and the mobile sinks are able to obtain the data of interest by querying only the nodes in the rendezvous area. These protocols presented different rendezvous area structures to reduce the storage overhead and the number of transmissions. Despite the benefit of the rendezvous area concept, it is cost-ineffective to rely on coordinate systems. Explicitly, using the physical coordinate based on a GPS device is costly and energy-consuming while employing a virtual coordinate system incurs high control overhead for coordinate construction and usually does not offer delivery guarantee (or suffers from high complexity to ensure delivery).

To avoid the disadvantages of adopting a coordinate system, DDB [20] and HCDD [19] proposed to construct a data dissemination backbone by organizing the sensor nodes into a tree or a hierarchical cluster or structure, respectively. Without the need of coordinate information, the backbone formation is carried out via message exchange among sensor nodes. Over the data dissemination backbone, the query message of a sink can be propagated to reach the sources with the desired data. Since the query and data transmissions over the backbone only require a small number of sensor nodes to participate, the energy consumption of data dissemination can be reduced. Although having successfully realized data dissemination at a low cost without any coordinate system, DDB and HCDD suffered from the low query efficiency (excessive number of data and query transmissions to solve queries). Moreover, DDB and HCDD did not tackle the hot spot problem, causing the shorter network lifetime.

In view of this, we propose in this paper a hierarchical role-based data dissemination (HRDD) protocol to provide energy-efficient and scalable data dissemination in wireless sensor networks. The objectives of HRDD are to reduce the numbers of query and data transmissions, accelerate data delivery, and alleviate the hot spot problem by three mechanisms, namely backbone construction, data delivery, and backbone maintenance. First, the backbone construction mechanism organizes sensor nodes into a two-level hierarchical cluster structure as the data dissemination backbone where a small number of sensor nodes serve as the cluster heads. To further reduce the number of query transmissions over the cluster backbone, a few sensor nodes and cluster heads are designated as special roles called agents to mitigate unnecessary query forwarding. Then, on top of the cluster backbone, the data delivery mechanism includes the event notification, query forwarding, and data delivery processes to enable efficient data dissemination. The data delivery mechanism provides shorter data forwarding paths to not only speed up data delivery, but also save data transmissions. Finally, to alleviate the hot spot problem, the backbone maintenance mechanism adaptively transfers the role of low-energy cluster heads and agents to the other nodes with higher residual energy. To evaluate the performance of HRDD, we conduct several experiments and compare HRDD to HCDD. The experimental results demonstrate that HRDD outperforms HCDD substantially in terms of network lifetime, delay, and success ratio.

The rest of this paper is organized as follows. Section 2 reviews related work. In Sect. 3, we describe the design of HRDD in detail. The experimental results are presented in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Related work

SPIN [15], Directed diffusion [17], and DRP [8] are the early works for data dissemination in wireless sensor networks. SPIN is a flooding-based protocol and introduces a high-level data descriptor (metadata) for negotiation. With the negotiation, a source could transmit the data messages only to those interested sinks, reducing data redundancy. Similarly, Directed diffusion and DRP also exploit the data-centric paradigm to enable efficient data dissemination. However, mobile sinks have to continuously update their locations and thus cause substantial energy expenditure since sink mobility is not considered in the design of both protocols. To overcome the problem of costly metadata flooding and location updates, a vast number of data dissemination protocols have been proposed [13] and could be broadly classified into two categories, namely coordinate-aware protocols and coordinate-free protocols. We review these protocols in the following.

2.1 Coordinate-aware protocols

To support sink mobility and conserve the energy of sensor nodes, [9, 12, 22, 25, 28, 29, 32, 36] have proposed the rendezvous area for disseminating data from sources to mobile sinks. These protocols assume that each sensor node obtains a coordinate by equipping with a GPS device or employing some virtual coordinate system [4, 11, 24, 31, 39]. A rendezvous area, which is created per source or shared by all sources, is used to maintain the detected data information or the corresponding metadata. Upon detecting the event of interest, a source uses a greedy geographic routing protocol (e.g., GPSR [18] and GFG [6]) to send the message to the rendezvous area. When a mobile sink intends to query the data of interest, the sink is able to access the data by querying only the sensor nodes in the rendezvous area. TTDD [36] proposes that each source builds a grid as the rendezvous area when sensing a new event. With the grid, a mobile sink can use limited query flooding within the local grid cell and query forwarding along the grid to retrieve the interested data. In [9], Das and Pucha design a Column-Row Location Services (XYLS) for data dissemination. In XYLS, a source creates its rendezvous area by replicating the data in the north and south directions based on its location. A query sink can find the desired data by propagating the query in a horizontal direction, leading the query message to reach the node having the desired data.

Since per-source rendezvous area creation of TTDD and XYLS incurs high control overhead, Shin et al. [29] present a Railroad system for large-scale sensor networks. Only one Rail, placed in the middle of the network, acts as the rendezvous area. In case of event detection, the source stores the data and sends the corresponding metadata to the nearest station (a group of nodes in the Rail). The query issued by a sink is forwarded to the Rail and then along the Rail in a circular way until it contacts the node with the matadata of interest. Such node in turn relays the query message to the corresponding source with the desired data. Similarly, LBDD [12] uses a virtual line which has width w and is divided into groups of size g as the rendezvous area. By adjusting w and g of the line, LBDD can perform well in event-driven and query-based scenarios. HDDS [32] and QDD [22] introduce a Quadtree hierarchial structure to perform data dissemination. Sources and sinks transmit their data and query messages to the nearest dissemination node in charge of maintaining data and answering queries, respectively. The main difference between HDDS and QDD is that HDDS generates the Quadtree based on sink density and achieves better load balancing. GHT [25] designs a geographic hash table to map the data type into a specific geographic coordinate. A source forwards the sensed data to the home node closest to the hashed coordinate and responsible for storing the data. A mobile sink can easily retrieve the data from the home node according to the GHT. By applying the GHT, Locator [28] proposes that the sensor nodes, located at the specific locations determined by the hashing the sink IDs, are responsible for caching sink locations. By doing so, sources can know the current locations of query sinks, and thus could transmit interested data messages to the sinks at a low cost.

2.2 Coordinate-free protocols

To save the energy of using coordinate systems as well as avoid the flooding cost, DDB [20] and HCDD [19] take advantage of backbones for data dissemination. In DDB, Lu and Valois propose to employ a localized self-organization scheme (LEGOS [21]) to form a virtual backbone and build a dynamic directed dissemination

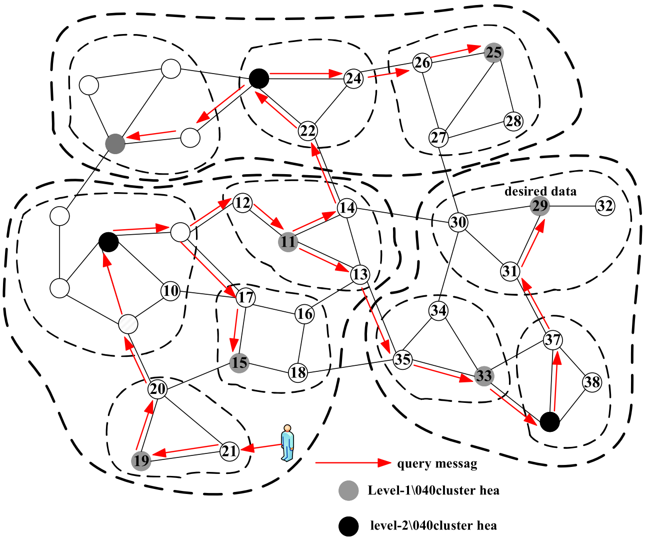


Fig. 1 Query forwarding in HCDD with a 2-level hierarchical cluster structure

tree on top of the backbone. Each sensor node is either a leader, a gateway, or a member. A member is associated with only one leader and any two leaders communicate with each other via a gateway. The backbone is composed of only leaders and gateways. When a sink wants to inject a query message to the network, it transmits the query message over the backbone. When all the leaders and gateways have received the query message, a directed dissemination tree is built with respect to the query message (the sink). When an event is sensed by some source, the source sends the data message to its corresponding leader, which in turn forwards the message to the query sink along the dissemination tree. The main drawback of DDB is that DDB produces a large number of query transmissions and the tree structure results in a single point of failure.

In HCDD, Lin et al. present a hierarchical cluster-based data dissemination scheme to reduce the number of involved sensor nodes. HCDD organizes the sensor nodes into a k-level hierarchical cluster structure as the data dissemination backbone. A mobile sink registers its location to the corresponding highest level cluster head, which in turn sends the location registration message to all the other highest level cluster heads. Each node receiving the location registration message maintains the ID of the sender for setting up a reverse path to the query sink, as shown in Fig. 1. When a source has a data message for a query sink, the data message is delivered along the reverse path to the sink. Although HCDD could save energy by using only a small number of involved sensor nodes for dissemination, the hierarchical cluster backbone causes redundant query transmissions to those cluster without data of interest and longer data paths on the backbone. Table 1 summarizes the above data dissemination protocols.

Table 1 Comparison of existing data dissemination protocols

Protocols	Coordinate awareness	Virtual structure	Per-source creation or all sources sharing	Disseminated information
TTDD	Yes	Grid	Per-source	Data
XYLS	Yes	Random line	Per-source	Data
Railroad	Yes	Rail	All sources	Metadata
LBDD	Yes	Line	All sources	Data
HDDS	Yes	Quadtree	All sources	Data
QDD	Yes	Quadtree	All sources	Data
GHT	Yes	Hashed coordinate	All sources	Data
Locator	Yes	Hashed coordinate	All sources	Sink location
DDB	No	Tree	All sources	Query
HCDD	No	Hierarchical cluster	All sources	Data

3 Hierarchical role-based data dissemination

In this section, we elaborate the proposed hierarchical role-based data dissemination (HRDD) protocol in wireless sensor networks with mobile sinks. HCDD is a coordinate-free protocol, which aims to achieve high efficiency of data efficiency and the long network lifetime. HRDD consists of three mechanisms: backbone construction, data dissemination, and backbone maintenance. In the backbone construction mechanism, HRDD establishes a two-level hierarchical cluster structure as the data dissemination backbone without requiring the location information of sensor nodes. The two-level hierarchical cluster backbone is composed of *high-level cluster heads* and *low-level cluster heads*. In addition, a subset of sensor nodes and cluster heads are selected to act as *indexing agents* and *gateway agents*, respectively. The indexing agents are utilized to forward the query messages from high-level cluster heads to only the low-level cluster heads with data of interest, while the gateway agents are used to avoid redundant query forwarding between adjacent high-level cluster heads.

Next, the data dissemination mechanism includes the event notification, query forwarding, and data delivery processes on top of the constructed cluster backbone. The event notification process enables the sources to diffuse the detected events to the responsible low-level cluster heads and the indexing agents for future queries. When a sink intends to query the sensor network about the data of interest, the data forwarding process directs the query message toward the low-level cluster heads with the desired data over the cluster backbone. To accelerate data delivery and save data transmissions, the data delivery process propagates the data message back to the query sink along shorter data paths. Finally, since the cluster heads as well as the indexing and gateway agents are heavily involved in data dissemination, they are prone to dissipate their energy more quickly. In order to prolong the network lifetime, the adaptive backbone maintenance mechanism is introduced to allow low-energy cluster heads and agents to reduce their load by designating the sensor nodes with high residual energy as the new cluster heads or agents. In what follows, we detail the three mechanisms.

3.1 Backbone construction

3.1.1 Hierarchical cluster formation

Most of the data dissemination protocols require either the global position system or some virtual coordinate system [11, 24, 31] to determine the locations of sensor nodes and then create the rendezvous area for data dissemination. However, the use of GPS devices or the construction of a virtual coordinate system greatly consumes the energy of sensor nodes and thus are cost-ineffective for data dissemination. Thus, we exploit a two-level hierarchical cluster structure as the backbone for data dissemination. Note that HRDD can be easily modified to adopt a k-level hierarchical cluster structure. We determine to use a two-level hierarchical cluster backbone for two reasons. One is that the experimental results of HCDD suggest that a small value of k leads to better performance; the other is that using a two-level hierarchical cluster structure is simple, and thus will be easily realized in practice. To construct the two-level hierarchical cluster backbone, we employ the existing clustering algorithm in the literature since we do not focus on clustering in this paper. Due to the benefits of clustering, a number of clustering algorithms [1, 3, 5, 7, 23] have been proposed. Considering that the Max–Min d-Cluster algorithm [3] is a distributed and low-cost clustering algorithm achieving load balancing among cluster heads, we choose the Max–Min d-Cluster algorithm for the creation of the two-level hierarchical cluster backbone.

To build the two-level hierarchical cluster backbone, each sensor node initially runs the Max–Min d-Cluster algorithm to determine itself as a cluster head or a cluster member. A cluster member also realizes the corresponding cluster head and the neighbor on the shortest path toward the cluster head. These cluster heads are referred to as *low-level cluster heads*. Figure 2(a) shows an example of the clustering result where nodes 1, 5, 11, 15, 19, 22, 25, 29, 33, and 36 become the low-level cluster heads. Then these low-level cluster heads again execute the Max–Min d-Cluster algorithm to elect the *high-level cluster heads*. As shown in Fig. 2(b), low-level cluster heads 5, 22, 36 are elected as high-level cluster heads. The two-level hierarchical cluster backbone is formed and consists of the low-level and high-level cluster heads. Note a *border cluster member* is the cluster member that has adjacent neighbors belonging to different clusters and serves as the communication bridge between two adjacent clusters. For instance, in cluster 19 of Fig. 2(a), cluster member 20 is a border cluster member and is able to help the communication between clusters 5 and 15 and cluster 19. Similarly, a *border low-level cluster head* is a low-level cluster head that connects to other low-level cluster heads associated with different high-level cluster heads and allow two adjacent high-level cluster heads to communicate with each other. An example of the border low-level cluster head in Fig. 2(b) is the low-level cluster head 33 between high-level clusters 36 and 5. In the cluster backbone, the low-level cluster heads are responsible for storing the data detected by the sources, while the high-level cluster heads are in charge of forwarding queries from and data to the sinks. The two-level hierarchical cluster backbone not only facilitates data dissemination but also enables the mobile sinks to query the data of interest at a low transmission cost.

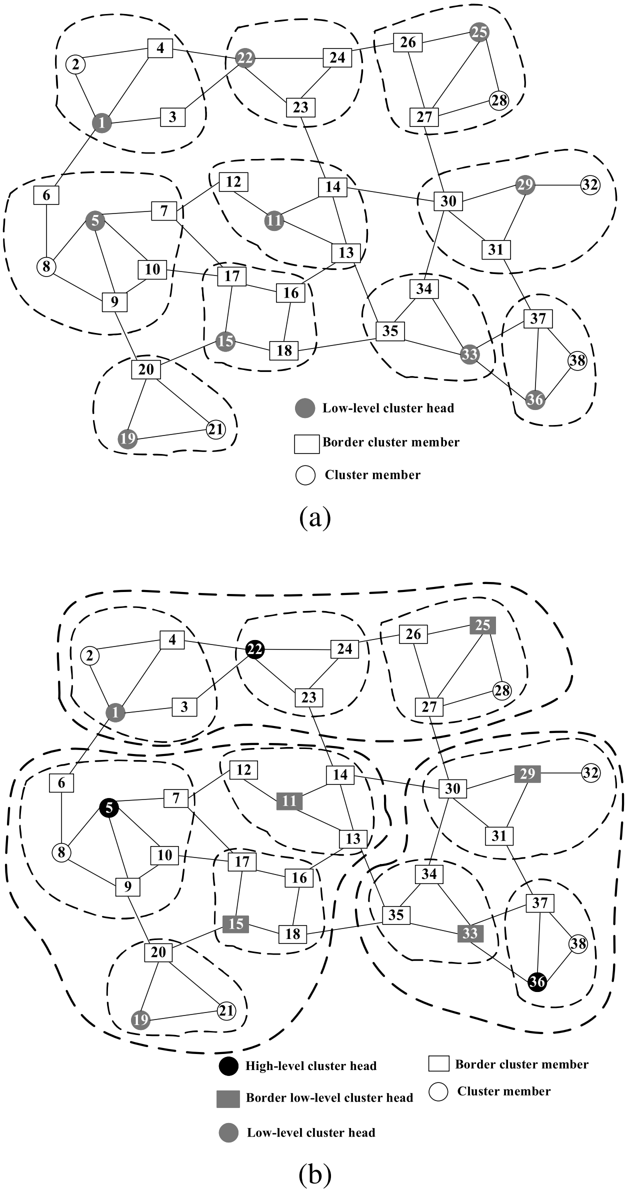


Fig. 2 An example of two-level hierarchical cluster backbone. (**a**) Low-level cluster structure. (**b**) High--level cluster structure

3.1.2 Agent selection

As aforementioned, the two-level hierarchical cluster backbone enables data dissemination to be performed by only a small subset of sensor nodes. To further reduce the number of query transmissions, we propose to designate a small number of sensor

Table 2 An example of the indexing agent candidate table

Agent candidate	Number of adjacent clusters	Adjacent cluster IDs
7	2	11, 15
9	1	19
10	1	15

nodes as the *indexing agents* and *gateway agents*. An indexing agent of a high-level cluster is a border cluster member of the high-level cluster and maintains the metadata information of the data stored at the associated low-level cluster heads. The introduction of the indexing agents is to make the query messages only be forwarded from a high-level cluster head to only the associated low-level cluster heads with the desired data. On the other hand, a gateway agent of a high-level cluster is a border low-level cluster head of the high-level cluster and keeps track of the routing information toward the other high-level cluster heads. The gateway agents are used to eliminate redundant query forwarding among neighboring high-level cluster heads. For a high-level cluster, we propose an agent selection algorithm to determine the indexing agents and gateway agents from its border cluster members and low-level cluster heads, respectively.

Since the indexing agents and the gateway agents are determined based on the same principle, we describe only the indexing agent selection mechanism as follows. After the formation of the cluster backbone, each high-level cluster head broadcasts an *HCH_NOTIFY* message with TTL $= d$ to the cluster members in its cluster. When receiving the *HCH_NOTIFY* message, a border cluster member replies a *BORDER_MEMBER_REPLY* message containing its own ID and the number of its adjacent clusters to the high-level cluster head. On receipt of a *BORDER_MEMBER_REPLY* message, the high-level cluster head updates the *indexing agent candidate table* with a new entry. After collecting all *BORDER_MEMBER_REPLY* messages, the high-level cluster head runs algorithm *AGENT_SELECTION* to determine the indexing agents from agent candidates. The idea of algorithm *AGENT_SELECTION* is to iteratively select the candidate with the largest number of adjacent clusters as an indexing agent in order to minimize the number of indexing agents. The fewer the indexing agents are, the smaller number of query transmissions between a high-level cluster heads and its indexing agents is.

We take the high-level cluster head 5 in Fig. 2(b) as an example to illustrate the agent selection mechanism. First, high-level cluster head 5 broadcasts an *HCH_NOTIFY* message with TTL $= 1$ to its cluster members. When border cluster member 7 receives the *HCH_NOTIFY* message, it replies a *BORDER_MEMBER_REPLY* message with its ID 7 as well as the IDs of the cluster heads of its adjacent clusters (i.e., 11 and 15). Similarly, border cluster members 9 and 10 send their *BORDER_MEMBER_REPLY* messages with the ID information back to high-level cluster head 5. With the obtained *BORDER_MEMBER_REPLY* messages, high-level cluster head 5 maintains an agent candidate table, as shown in Table 2. With the agent candidate table, high-level cluster head 5 runs algorithm *AGENT_SELECTION* to determine the indexing agents. First, high-level cluster head 5 selects the candidate 7 as an indexing agent since the candidate 7 has the largest

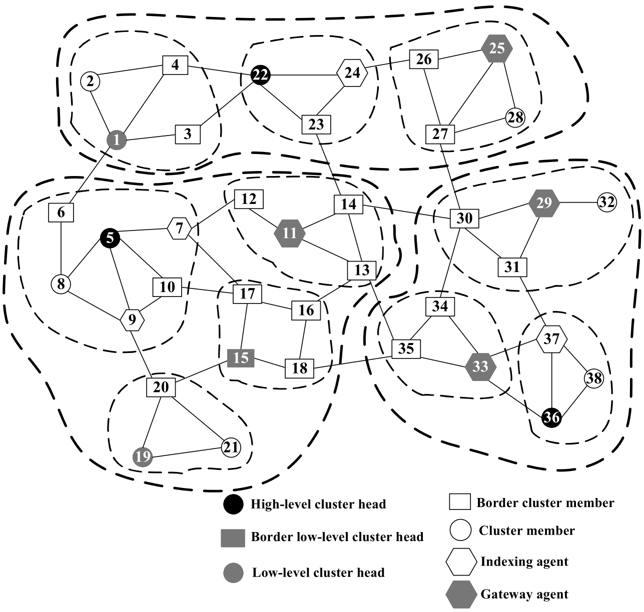


Fig. 3 An example of indexing agent determination

number of adjacent clusters. Note that since the indexing agent 7 is responsible for the adjacent cluster 15, the adjacent cluster 15 of agent candidate 10 is removed, and thus the number of adjacent clusters of candidate 10 is decreased to 0. High-level cluster head 5 proceeds to assign agent candidate 9 as an indexing agent due to the larger number of its adjacent clusters. Since all low-level clusters associated with high-level cluster head 5 have been taken on, algorithm *AGENT_SELECTION* terminates and the indexing agents are nodes 7 and 9, as depicted in Fig. 3. With algorithm *AGENT_SELECTION*, a high-level cluster head is able to employ fewer indexing agents for the associated low-level clusters, thereby reducing the number of query messages sent from a high-level cluster head to the associated low-level cluster heads.

3.2 Data dissemination

To diffuse sensed data from sources to mobile sinks over the two-level hierarchical cluster backbone, we present the event notification, query forwarding, and data delivery processes. These processes aim to achieve high query efficiency by reducing the numbers of query and data transmissions. With these processes, a mobile sink can retrieve the data of interest more quickly while conserving the energy of sensor nodes.

3.2.1 Event notification

Here, we start to introduce the event notification process with the two-level hierarchical cluster backbone. When a sensor node detects an event of interest, the sensor node

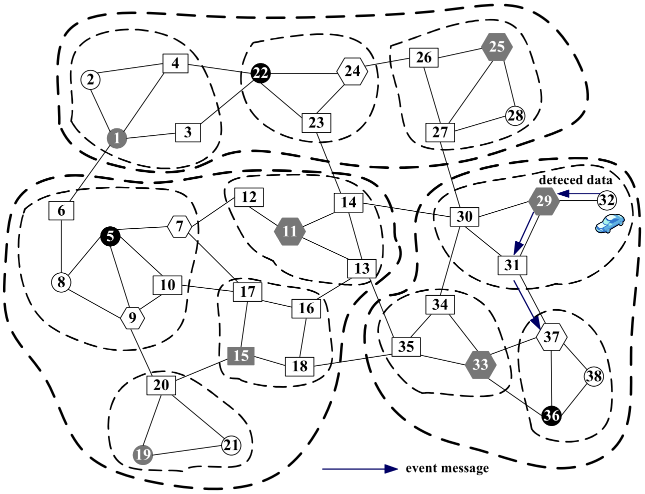


Fig. 4 Low-level cluster head caches the detected event and indexing agent stores the metadata

reports the event to the corresponding low-level cluster head. The low-level cluster head stores the event report and will be responsible for answering the future queries interested in the event. Then the low-level cluster head generates the metadata of the event and delivers the metadata message to the corresponding indexing agent assigned by its high-level cluster head. The corresponding indexing agent maintains the metadata together with the ID of the responsible low-level cluster head. For example, Fig. 4 depicts that low-level cluster head 29 stores the data sensed by node 32 and indexing agent 37 maintains the corresponding metadata. The maintained metadata enables the indexing agent to forward the query message, received from the high-level cluster head, to only the low-level cluster head with the desired data, thereby mitigate unnecessary query forwarding.

3.2.2 Query forwarding

When a mobile sink intends to issue a query about the data of interest, the mobile sink sends the query message to the nearest sensor node acting as a proxy for the sink. On receiving the query message, the sensor node forwards the query message to its low-level cluster head. The low-level cluster head in turn forwards the query message to the corresponding high-level cluster head. Additionally, if the low-level cluster head has the desired data for the sink, the low-level cluster head replies the data message to the sink. When the high-level cluster head receives the query message, it executes the two operations below. Note these two operations are also executed by all other high-level cluster heads.

1. *Operation 1.* The high-level cluster head forwards the query message to the indexing agents to find whether the associated low-level cluster heads has the data of interest. Thanks to algorithm *AGENT_SELECTION* and the metadata at the indexing agents, the number of query transmissions between the high-level cluster

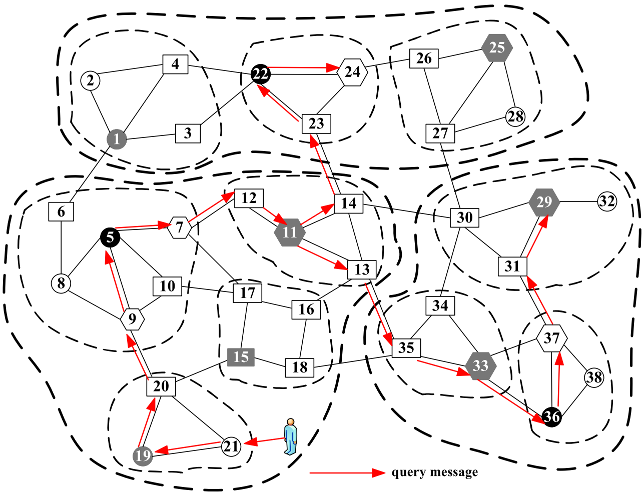


Fig. 5 An example of query forwarding

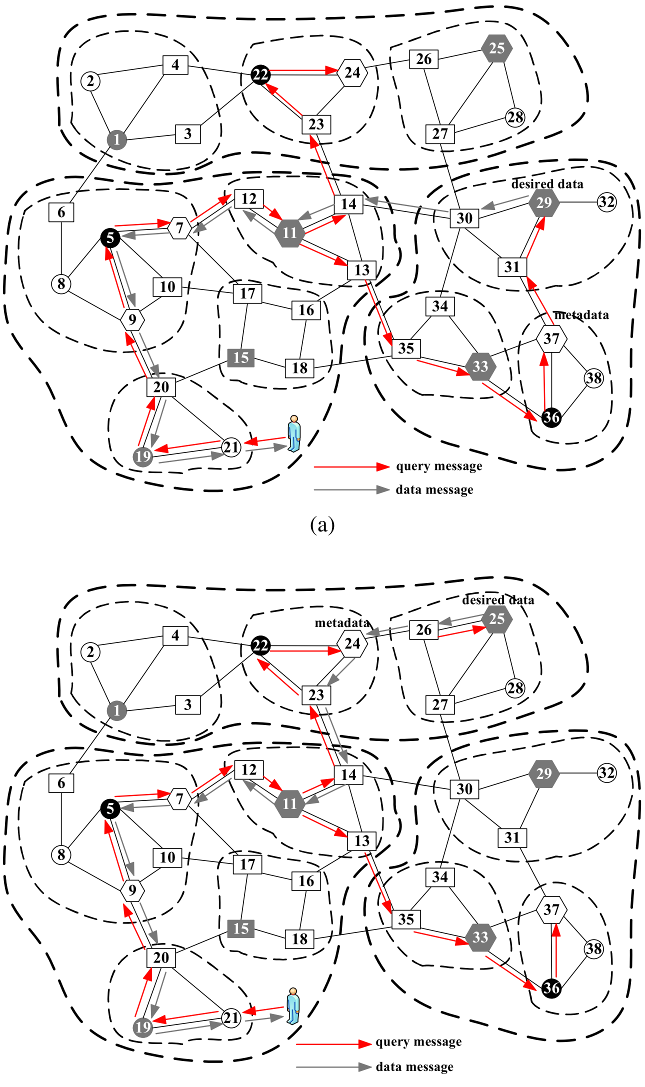
head and the indexing agents is minimized and the query transmissions only occur when the associated low-level cluster heads have the interested data.

2. *Operation 2*. The high-level cluster head forwards the query message to the adjacent high-level cluster heads via the gateway agents. When a gateway agent receives the query message, it relays the message to the associated high-level cluster heads to reach those low-level cluster heads with the desired data in other high-level clusters.

This process continues until all high-level cluster heads have processed the query message. Note that each sensor node participating in the query forwarding process stores the ID of the node from which it receives the query message to set up the reverse data path. An example of the query forwarding process is given in Fig. 5. From Fig. 5, it can be seen that high-level cluster head 5 does not incur any redundant query transmission to adjacent high-level cluster heads 22 and 36 due to the help of gateway agent 11. Besides, thanks to the indexing agents, only low-level cluster head 29 receives the query message since it is the only low-level cluster head with the interested data. Figure 5 clearly demonstrates that HRDD effectively enhances the query efficiency.

3.2.3 Data delivery

To accelerate data delivery as well as reduce the number of data transmissions, we design an efficient data delivery process to provide shorter data delivery paths. To achieve our goal, the sink location as well as the ID of the corresponding gateway agents are attached to the query message. Based on such information, when a low-level cluster head with the desired data receives the query message, the data message may be able to be sent to the gateway agent directly instead of being transmitted along the reverse path. Similarly, an indexing agent obtaining the data message from the



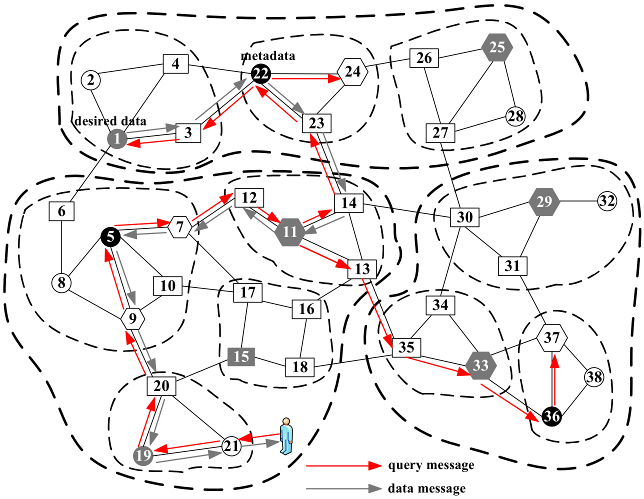
(a)

(b)

Fig. 6 Data delivery process. (**a**) Low-level cluster head knows the gateway agent. (**b**) Indexing agent knows the gateway agent. (**c**) Low-level cluster head reply along the reverse path

low-level cluster head may be able to relay the data message to that gateway agent directly. In such cases, since the data message reaches that gateway agent without being forwarded to the corresponding high-level cluster head, the data delivery paths are shorter and thus the data delivery time and the number of data transmissions are reduced. The data delivery process is composed of three cases and is detailed below.

- *Case 1*: When the low-level cluster head with the data of interest receives the query message forwarded by the indexing agent, it checks the ID of the gateway agent



(c)

Fig. 6 (*Continued*)

contained in the query message. If the gateway agent is its neighboring low-level cluster head, the low-level cluster head sends the data message to the gateway agent directly without forwarding to its high-level cluster head. As shown in Fig. 6(a), low-level cluster head 29 can communicate with gateway agent 11 with border cluster head 30. Thus, the number of data transmissions can be reduced from 7 (reverse path 29-31-37-36-33-35-13-11) to 3 (path 29-30-14-11).

- *Case 2*: When receiving the data message from the low-level cluster head, the indexing agent examines whether the ID of the gateway agent of the query message is identical to one of its responsible low-level cluster heads. If so, the indexing agent directly forwards the data message to the gateway agent without transmitting to the high-level cluster head. For example, in Fig. 6(b), since indexing agent 24 and gateway agent 11 can exchange messages via border cluster member 23, indexing agent 24 sends the data message directly to gateway agent 11 without high-level cluster head 22 being involved, saving one data transmission.
- *Case 3*: If no shorter path in the above two cases is available, the data message is delivered to the query sink along the reverse path, as shown in Fig. 6(c).

Note that we employ the well-known progressive-footprint-chaining strategy [27] to guarantee data delivery when query sinks continue the movement.

3.3 Backbone maintenance

In HRDD, the low-level and high-level cluster heads as well as the indexing and gateway agents are prone to dissipate their energy more quickly due to heavy loads on these nodes. As such, we propose an adaptive backbone maintenance mechanism for low-energy cluster heads and agents to extend the network lifetime. The idea of the backbone maintenance mechanism is that the low-energy cluster heads and agents will delegate the sensor nodes with high residual energy to serve as the new cluster heads and agents. In other words, the load on low-energy cluster heads or agents will be transferred to those sensor nodes having higher residual energy. According to the different roles, the backbone maintenance mechanism consists of three cases, which are described in detail below.

1. *Low-level cluster head*. When a low-level cluster head is aware of its current energy lower than the predefined threshold, it sends an *ENERGY_INQUIRY* message to its cluster members. On receiving the *ENERGY_INQUIRY* message, the cluster member replies its remaining energy to the cluster head. After collecting the energy information of all cluster members, the low-level cluster head selects the cluster member with the highest residual energy as the new cluster head. Finally, the low-energy cluster head sends a *NEW_LCH_NOTIFY* message to notify its cluster members, neighboring low-level cluster heads, and the corresponding high-level cluster head and indexing agent of the new low-level clusters.
2. *Indexing/Gateway agent*. If an indexing agent or a gateway agent has the residual energy below the predefined threshold, it sends an *AGENT_CHANGE_REQUEST* message to the corresponding high-level cluster head. When receiving the *AGENT_CHANGE_REQUEST* message, the high-level cluster head will look up the agent candidate table to find an available agent candidate to replace the low-energy agent. The high-level cluster head in turn issues a *NEW_AGENT_NOTIFY* message to request the selected agent candidate to become the new agent and to inform the associated low-level cluster heads. However, in case of the absence of available agent candidates, the high-level cluster head will transmit an *AGENT_CONT* message to request the low-energy agent to continue to serve as an agent.
3. *High-level cluster head*. When a high-level cluster head realizes its current energy lower than the predefined threshold or has received the *AGENT_CHANGE_REQUEST* messages from at least half of the indexing agents, the low-energy high-level cluster head starts the high-level cluster head changing procedure. First, the low-energy high-level cluster head issues an *ENERGY_INQUIRY* message to the associated low-level cluster heads. After the high-level cluster head has received the energy information from all the low-level cluster heads, it selects the low-level cluster head with the highest remaining energy as the new high-level cluster head. The high-level cluster head proceeds to sends a *NEW_HCH_NOTIFY* message to notify the selected new low-level cluster head, the associated low-level cluster heads and the neighboring high-level cluster heads of the new high-level cluster head. At last, the new high-level cluster head executes the agent selection mechanisms (Sect. 3.1.2) to determine the new indexing and gateway agents.

4 Performance evaluation

In this section, we evaluated the performance of HRDD and compared HRDD with HCDD [19]. All the experiments were conducted using the ns-2 simulator. The sinks and sensor nodes were deployed in a 2000 m × 2000 m sensor field. The default numbers of sinks and sensor nodes were set to 3 and 100, respectively. The maximum

Table 3 Default simulation parameters

Parameter	Value
Sensor field	2000 m × 2000 m
Number of sinks	3
Number of nodes	100
Query interval	50 s
Transmission range	100 m
Initial energy	2 J
Query message size	48 bytes
Data message size	48 bytes

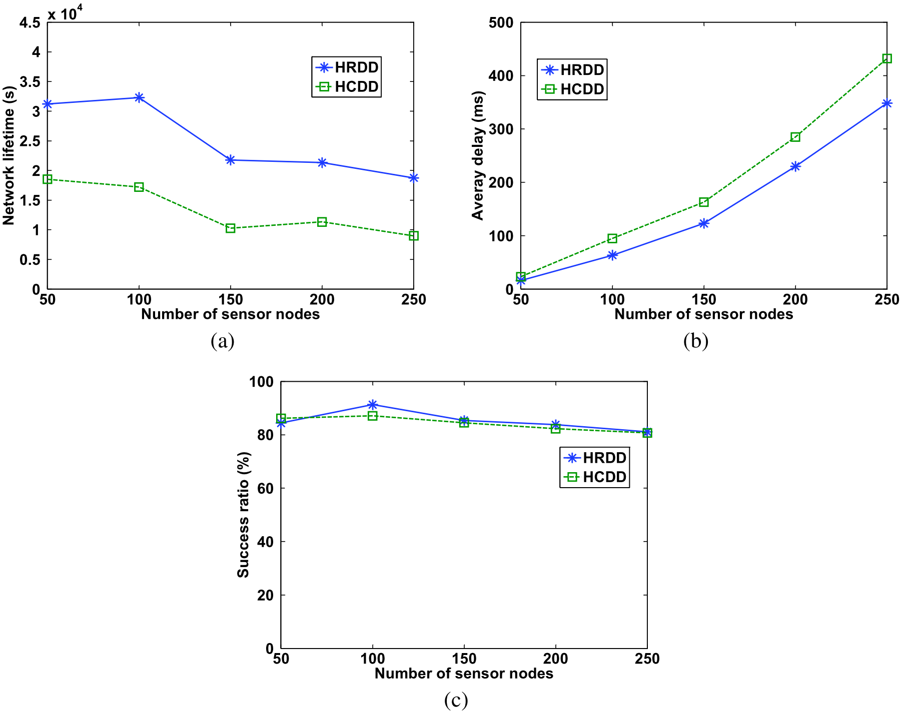
number of hops d between a sensor node and its corresponding cluster head was fixed at 2. We employed the random waypoint model as the sink mobility model. The two-ray ground model was the radio propagation model. The transmission range of each sensor node was 250 m. For measuring the energy consumption, we adopted the first-order radio model that was proposed in [14] and widely used in the literature. In this model, the energy spent by a sensor node to transmit a message of s bits over the distance d was $s \cdot (E_{\text{elec}} + E_{\text{amp}} \cdot d^q)$ J where $E_{\text{elec}} = 50$ nJ/bit was the energy spent by transmitter electronics for one-bit data, $E_{\text{amp}} = 0.0013$ pJ/bit/m^4 was the energy spent by transmitting amplifier for one-bit data, and $q = 4$ was the propagation loss exponent. To receive a message of s bits, a receiver consumes $s \cdot \alpha$ J. Both the sizes of a query message and a data message were set to 48 bytes [35]. The initial energy was 2 J for all sensor nodes. The default interval of query generation for each sink was set to 50 s. The event frequency of sensor nodes was 10 % (i.e., 10 % of sensor nodes report events per minute). Table 3 summarizes the default parameters used during simulation.

The performance metrics are listed below.

- *Network lifetime*: the time until the first sensor node runs out of energy.
- *Average Delay*: the average time between the moment a source transmits a data message and the moment a sink receives the data message over all the source-sink pairs.
- *Success Ratio*: the ratio of the number of data messages successfully received at a sink to the total number of data messages generated by a source, averaged over all source-sink pairs.

4.1 Effect of the number of sensor nodes

In this experiment, we evaluate the scalability of HRDD and HCDD by varying the number of sensor nodes. The number of sensor nodes increases from 50 to 250 in increments of 50. Figure 7(a) shows that the network lifetime of both HRDD and HCDD decreases as the number of sensor nodes increases. The reason is two-fold. First, the increase in the number of sensor nodes leads to more data messages since the number of sources becomes larger in the setting of 10 % event frequency. Second, more sensor nodes result in more high-level cluster heads, and thus cause more query



(a) (b) (c)

Fig. 7 Effect of the number of sensor nodes. (**a**) Network lifetime. (**b**) Average delay. (**c**) Success ratio

transmissions. Such larger numbers of data and query messages cause sensor nodes to consume more energy, thereby reducing the network lifetime. From Fig. 7(a), we also can see that HRDD extends the network lifetime by 68 % to 111 % compared with HCDD because (1) the proposed indexing and gateway agents and the data delivery process are effective in reducing the numbers of query messages and data messages, respectively, and (2) the adaptive backbone maintenance mechanism effectively extends the lifetime of low-energy cluster heads and agents by shifting their load to the sensor nodes having high remaining energy.

Next, Fig. 7(b) depicts that the larger number of sensor nodes gives rise to the longer average delay. It is because that the more high-level cluster heads resulted from the increase in the number of sensor nodes lead to longer query and data forwarding paths. Due to the shorter data paths of HRDD, HRDD reduces the average delay of HCDD by 19 % to 33 %, as shown in Fig. 7(b). It is worthwhile to mention that the improvement of HRDD over HCDD in average delay becomes more substantial with the number of sensor nodes increasing since more sensor nodes are beneficial for the proposed query forwarding and data delivery processes. Finally, for success ratio, we can observe from Fig. 7(c) that both HRDD and HCDD have high success ratios and the success ratio stays nearly constant regardless of the number of sensor nodes. Such result mainly benefits from the effectiveness of the two-level hierarchical cluster backbone.

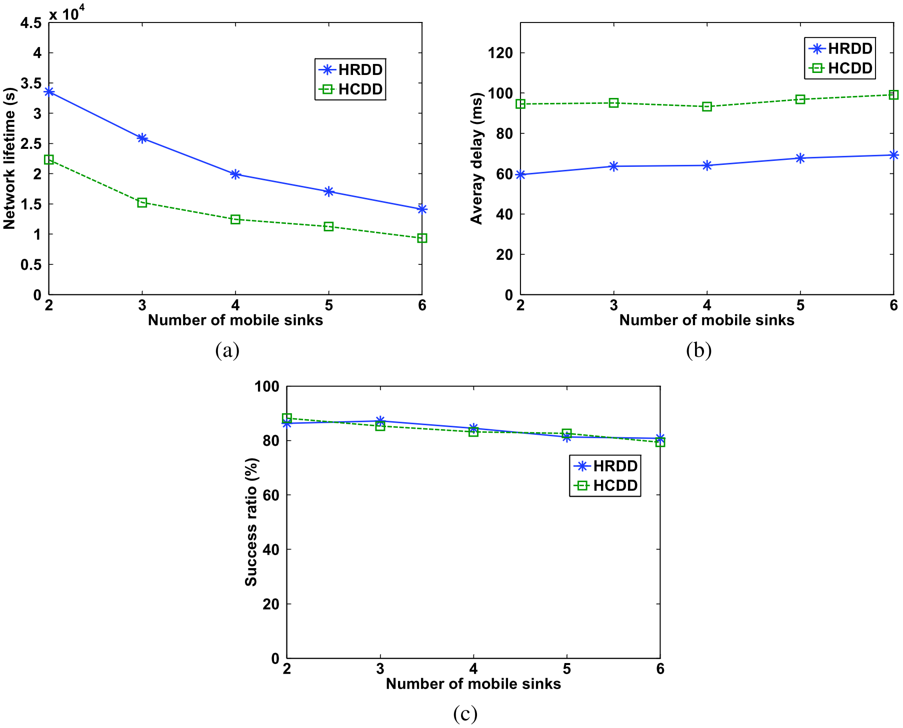


Fig. 8 Effect of the number of mobile sinks. (**a**) Network lifetime. (**b**) Average delay. (**c**) Success ratio

4.2 Effect of the number of mobile sinks

In this experiment, we study the effect of the number of mobile sinks on HRDD and HCDD with the number of mobile sinks ranging from 2 to 6. Figure 8(a) demonstrates that both HRDD and HCDD suffer from shorter network lifetime with the number of mobile sinks increasing, agreeing with the intuition that the more query messages arising from the increase in the number of sinks result in more energy consumption of sensor nodes. Besides, we find from Fig. 8(a) that the performance gain of HRDD over HCDD in terms of network lifetime is at least above 50 % regardless of the number of mobile sinks, indicating the effectiveness of the proposed data dissemination and backbone maintenance mechanisms. In terms of average delay and success ratio, the impact of the number of mobile sinks on both HRDD and HCDD is almost negligible thanks to the effectiveness of the hierarchical cluster backbone, as illustrated in Figs. 8(b) and 8(c), respectively. Because of the shorter data delivery paths of HRDD, HRDD achieves 30 % less average delay in comparison with HCDD, as shown in Fig. 8(b).

4.3 Effect of sink speed

In this experiment, we explore the effect of sink speed on the performance of HRDD and HCDD by increasing the sink speed from 4 m/s to 10 m/s. From Fig. 9(a), we can

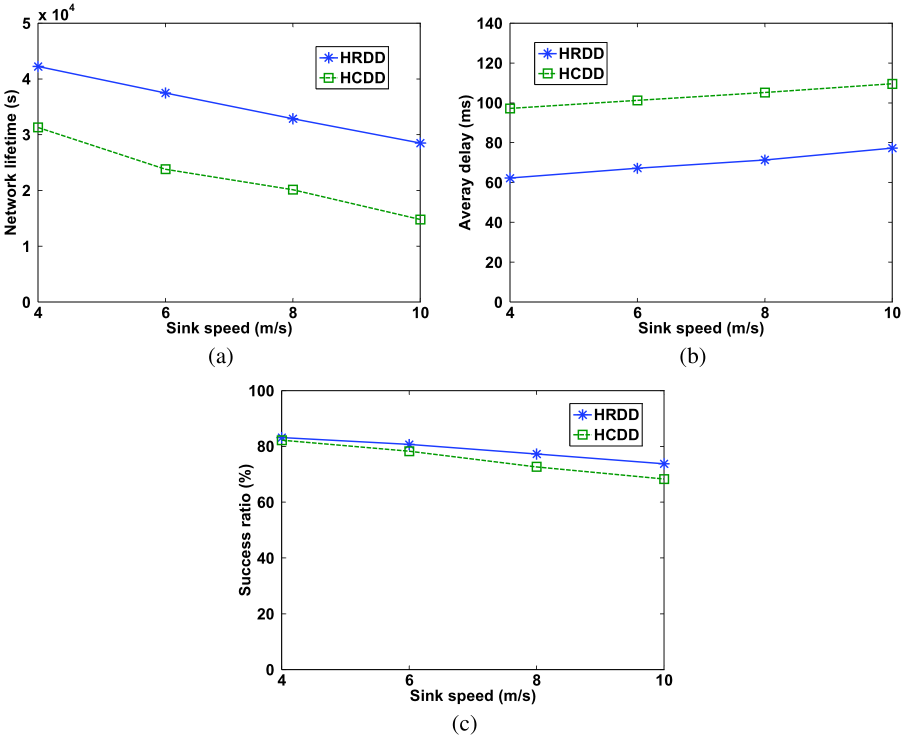
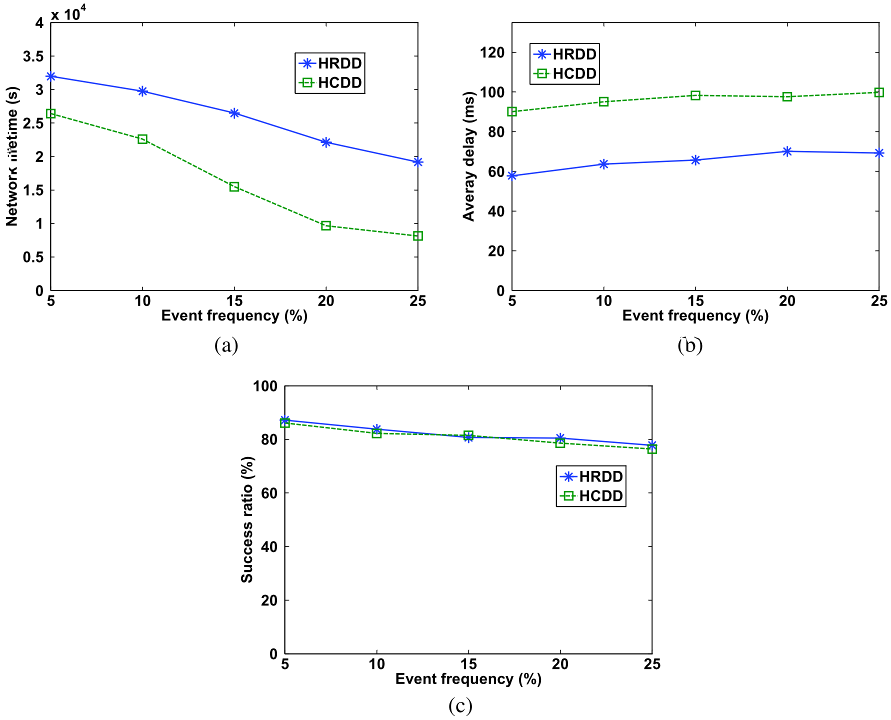


Fig. 9 Effect of sink speed. (**a**) Network lifetime. (**b**) Average delay. (**c**) Success ratio

see that the network lifetime of both HRDD and HCDD becomes shorter as the sink speed increases. Two reasons account for this result. One is that the higher speed of the sinks causes the sinks to move to different clusters at a higher probability and thus have to issues new query messages. The other is that the data delivery paths are longer in order to reach the moving sinks. The longer data delivery paths are also the main reason of longer average delays of HRDD and HCDD, as depicted in Fig. 9(b). Due to the advantages of the proposed data delivery process, HDRR constantly outperforms HCDD at all sink speeds. Figure 9(c) shows that the success ratios of HRDD and HCDD drop from 86 % to 68 % and from 87 % to 73 %, respectively. This is because the higher moving speed of the sinks leads to longer data paths and thus increases the probability of message loss. Besides, it is found from Fig. 9 that HRDD is more resilient to the increase in sink speed than HCDD in terms of success ratio, resulting from the shorter data delivery paths of HRDD.

4.4 Effect of event frequency

We investigate the effect of event frequency for both HRDD and HCDD in this experiment. We vary the event frequency from 5 % to 25 %. As expected, Fig. 10(a) shows that the network lifetime of both HRDD and HCDD decreases with the event frequency increasing. The performance gain of HRDD over HCDD rises from 21 %



(a) (b) (c)

Fig. 10 Effect of event frequency. (**a**) Network lifetime. (**b**) Average delay. (**c**) Success ratio

to 135 % with the increase in event frequency since the shorter data delivery paths of HRDD is more effective in reducing the number of data message transmissions. From Fig. 10(b), we observe that the higher event frequency results in longer average delay for both HRDD and HCDD due to the fact that the larger number of data transmissions increases the queueing time as well as channel contention. Finally, the success ratios of both HRDD and HCDD drop in the presence of higher event frequency, as shown in Fig. 10(c). This is because the higher event frequency causes more data transmissions, and thus leads to more message losses.

5 Conclusion

In this paper, we proposed a hierarchical role-based data dissemination (HRDD) protocol in wireless sensor networks with mobile sinks. In view of the drawbacks of previous protocols (i.e., the high cost of using coordinates for coordinate-aware protocols and low query efficiency, longer data paths, and the hot spot problem of coordinate-free protocols), HRDD is a coordinate-free protocol employing three mechanisms: backbone construction, data dissemination, and backbone maintenance. The backbone construction mechanism selects a small number of sensor nodes to act as high-level and low-level cluster heads to form the data dissemination backbone.

Besides, to further decrease the number of query transmissions, indexing and gateway agents are introduced to enable the query messages to be forwarded from a high-level cluster head to only those low-level cluster heads with the desired data and to adjacent high-level cluster heads, respectively. Leveraging the two-level hierarchical cluster backbone, the data delivery mechanism provides shorter data delivery paths to speed up data delivery as well as reduce the number of data transmissions. To extend the network lifetime by alleviating the hotspot problem, an adaptive backbone maintenance mechanism is proposed to transfer the load of low-energy cluster heads and agents to those sensor nodes with high residual energy. The extensive experimental results show that HRDD significant outperforms HCDD in terms of network lifetime and average delay.

Acknowledgements The authors thank the Taiwan Ministry of Economic Affairs and Institute for Information Industry for financially supporting this research (Industry and Government Application Empirical Environment Development Project).

References

1. Abbasi AA, Younis M (2007) A survey on clustering algorithms for wireless sensor networks. Comput Commun 30(14–15):2826–2841
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40(8):102–114
3. Amis AD, Prakash R, Vnong TH, Huynh DT (2000) Max–min d-cluster formation in wireless ad hoc networks. In: Proc IEEE INFOCOM, pp 32–34
4. Awad A, German R, Dressler F (2011) Exploiting virtual coordinates for improved routing performance in sensor networks. IEEE Trans Mob Comput 10(9):1214–1226
5. Basagni S (1999) Distributed clustering algorithm for ad hoc networks. In: Prof. ISPAN, pp 310–315
6. Bose P, Morin P, Stojmenović I, Urrutia J (2001) Routing with guaranteed delivery in ad hoc wireless networks. Wirel Netw 7(6):609–616
7. Chatterjee M, Das SK, Turgut D (2000) An on-demand weighter clustering algorithm (WCA) for ad hoc networks. In: Proc IEEE GLOBECOM, pp 1697–1701
8. Coffin DA, Hook DJV, McGarry SM, Kolek SR (2000) Declarative ad-hoc sensor networking. In: Proc SPIE, pp 109–120
9. Das SM, Pucha H, Hu YC (2005) Performance comparison of scalable location services for geographic ad hoc routing. In: Proc IEEE INFOCOM, pp 1228–1239
10. Dhurandher SK, Obaidat MS, Gupta M (2012) An acoustic communication based aqua-glomo simulator for underwater networks. Hum-Centr Comput Inf Sci 2(3):1–14
11. Fang Q, Gaoi J, Guibast LJ, de Silva V, Zhang L (2005) GLIDER: gradient landmark-based distributed routing for sensor networks. In: Prof. IEEE INFOCOM, pp 339–350
12. Hamida EB, Chelius G (2008) A line-based data dissemination protocol for wireless sensor networks with mobile sink. In: Proc IEEE ICC, pp 2201–2205
13. Hamida EB, Chelius G (2008) Strategies for data dissemination to mobile sinks in wireless sensor networks. IEEE Wireless Commun Mag 15(6):31–37
14. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proc HICSS
15. Heinzelman WR, Kulik J, Balakrishnan H (1999) Adaptive protocols for information dissemination in wireless sensor networks. In: Proc ACM MobiCom, pp 174–185
16. Huang C, Cheng R-H, Chen S-R, Li C-I (2010) Enhancing network availability by tolerance control in multi-sink wireless sensor networks. J Org Chem 1(1):15–22
17. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. IEEE/ACM Trans Netw 11(1):2–16
18. Karp B, Kung HT (2000) GPSR: greedy perimeter stateless routing for wireless networks. In: Proc ACM MobiCom, pp 243–254

19. Lin C-J, Chou P-L, Chou C-F (2006) HCDD: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink. In: Proc IWCMC, pp 1189–1194
20. Lu J-L, Valois F (2007) On the data dissemination in WSNS. In: IEEE WiMob, p 58
21. Lu J-L, Valois F, Barthel D (2007) Low-energy self-organization scheme for wireless ad hoc sensor networks. In: Proc WONS, pp 138–145
22. Mir ZH, Ko Y-B (2006) A quadtree-based data dissemination protocol for wireless sensor networks with mobile sinks. In: PWC
23. Qin M, Zimmermann R (2005) An energy-efficient voting-based clustering algorithm for sensor networks. In: Proc SNPD-SAWN, pp 444–451
24. Rao A, Ratnasamy S, Papadimitriou C, Shenker S, Stoica I (2003) Geographic routing without location information. In: Proc ACM MobiCom, pp 96–108
25. Ratnasamy S, Karp B, Shenker S, Estrin D, Govindan R, Yin L, Yu F (2003) Data-centric storage in sensornets with GHT, a geographic hash table. Mob Netw Appl 8(4):427–442
26. Sarkar P, Saha A (2011) Security enhanced communication in wireless sensor networks using reed-muller codes and partially balanced incomplete block designs. J Org Chem 2(1):22–30
27. Shen C-C, Srisathapornphat C, Jaikaeo C (2001) Sensor information networking architecture and applications. IEEE Pers Commun 8(4):52–59
28. Shim G, Park D (2006) Locators of mobile sinks for wireless sensor networks. In: Proc Intl Conf Parallel Proc Wksps, pp 159–164
29. Shin J-H, Kim J, Park K, Park D (2005) Railroad: virtual infrastructure for data dissemination in wireless sensor networks. In: Proc PE-WASUN, pp 168–174
30. Singh B, Lobiyal DK (2012) A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks. Hum-Centr Comput Inf Sci 2(13)
31. Tsai M-J, Yang H-Y, Liu B-H, Huang W-Q (2009) Virtual-coordinate-based delivery-guaranteed routing protocol in wireless sensor networks. IEEE/ACM Trans Netw 17(4):1028–1041
32. Visvanathan A, Youn J-H, Deogun J (2005) Hierarchical data dissemination scheme for large scale sensor networks. In: Proc IEEE ICC
33. Wang X, Ma J, Wang S, Bi D (2010) Distributed energy optimization for target tracking in wireless sensor networks. IEEE Trans Mob Comput 9(1):73–86
34. Wang Y, Wang X, Xie B, Wang D, Agrawal D (2008) Intrusion detection in homogeneous and heterogeneous wireless sensor networks. IEEE Trans Mob Comput 7(6):698–711
35. Wu Y, Fahmy S, Shroff NB (2008) On the construction of a maximum-lifetime data gathering tree in sensor networks—np-completeness and approximation algorithm. In: Proc IEEE INFOCOM, pp 1013–1021
36. Ye F, Luo H, Cheng J, Lu S, Zhang L (2002) A two-tier data dissemination model for large-scale wireless sensor networks. In: Proc ACM MobiCom, pp 148–159
37. Ye G-H, Yeh L-Y, Huang J-L (2010) Hierarchical role-based data dissemination for large-scale wireless sensor networks with mobile sinks. In: Proceedings of the 6th international workshop on mobile peer-to-peer computing
38. Yick J, Mukherjee B, Ghosal D (2005) Analysis of a prediction-based mobility adaptive tracking algorithm. In: Proc BROADNETS, pp 809–816
39. Zhao Y, Chen Y, Li B, Zhang Q (2007) Hop ID: a virtual coordinate-based routing for sparse mobile ad hoc networks. IEEE Trans Mob Comput 6(9):1075–1089