國立交通大學

資訊科學與工程研究所

碩 士 論 文

在可任意移動的Sinks中利用階級的角色方式進行資料散播

Hierarchical Role-based Data Dissemination with Mobile Sinks and Targets

研 究 生：葉國禾

指導教授：黃俊龍　教授

中 華 民 國 九 十 七 年 八 月

在可任意移動的Sinks中利用階級的角色方式進行資料散播

Hierarchical Role-based Data Dissemination with Mobile Sinks and Targets

研 究 生：葉國禾　Student：Guo-He Ye

指導教授：黃俊龍　Advisor：Jiun-Long Huang

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

摘要

在無線感測網路的環境裡, 感測器不只是用來測量眞實世界的資料, 也可以用來儲存、處理和傳送所測量到的資料, 在之前就有一些有關資料散播的協定被提出, 但是大部分的協定對於任意移動的 sinks 仍然存在許多的議題, 在本論文中, 我們提出了在可任意移動的 Sinks 中利用階級的角色方式進行資料散播, 我們利用 hierarchical cluster-based 的結構, 指定兩個 roles 來幫助感測器傳遞資料, 分別是 Indexing Agent 和 Gateway Agent, Indexing Agent 是一個可以降低一些不必要的訊息查詢, 而 Gateway Agent 是用來降低 sink 在做註册時, 所浪費的多餘的訊息傳遞, 我們和之前的所提出的協定一起測量且比較他們之間通訊成本和訊息量, 結果顯示出我們所提出的協定更能降低網路的能源消耗並且可以延長整個網路的壽命。

Abstract

In wireless sensor networks, sensor nodes are capable of not only measuring real world phenomena, but also storing, processing, and transferring these measurements. Many protocols have been proposed for disseminating event data. However, most of them still bring some challenges to multiple mobile sinks. In this paper, we propose the Hierarchical Role-based Data Dissemination approach, named HRDD, with multiple mobile sinks for large-scale wireless sensor networks. In HRDD, we use a hierarchical cluster-based structure to discover and maintain the routing paths for distributing data to the mobile sink. We assign two roles, called Indexing Agent and Gateway Agent to some sensors in the wireless sensor networks. Indexing Agents are used to avoid unnecessarily transferring the query messages, while Gateway Agents are used to decrease energy consumption of broadcasting and number of flooding messages. We evaluate and compare the communication cost and message complexity of HRDD with previous approaches. Our results show that HRDD is able to reduces the energy consumption of the wireless sensor networks and achieve longer network lifetime.

誌謝

兩年的碩士班生活至此已告一段落, 無論是生活、課業及論文撰寫多虧有大家的幫助與陪伴, 讓我有今日收成的喜悅。

這一篇論文能夠順利完成, 首先要誠摯的感謝我的指導教授－黃俊龍老師。在老師辛勞的教導及啓發下, 論文才能夠如期的完成。再來要感謝我的口試委員胡誌麟老師、劉傳銘老師與鄧維光老師, 在口試期間給予寶貴意見, 讓我的論文表達上更加清楚, 在此一併致謝。還要感謝我的同學及好友, 在論文研究期間給予我精神上莫大之鼓勵。

最後要感謝我的家人, 有你們的贊助及支持才能順利的完成學業。另外還有很多曾經幫助過我的朋友, 因爲有大家的幫助, 我才能有今天的成果。

葉國禾 謹誌於交通大學資訊科學與工程研究所
中華民國九十七年八月

Contents

List of Figures

List of Tables

Chapter 1

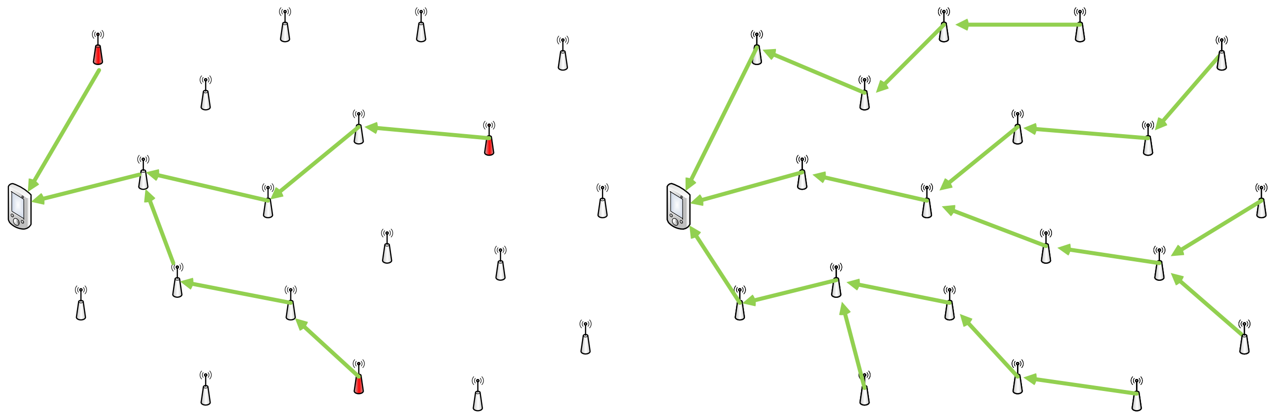
INTRODUCTION

Wireless sensor networks (WSNs) [1] usually contain thousands or millions of sensors that are randomly and widely deployed in an area of interest. The sensor nodes are used to monitor the phenomena in the network, collect interested data, and forward the measured data toward a control center that requests this data. The sensor which generates data reports is called a source node, while the control center is called a sink which generates query messages. The sink which is equipped with a database system, sends queries or control commands to sensor nodes and collects information from the sensors. Each sensor node is a small device with limited resources: limited battery power, small memory, restricted computing capability, and a short-range wireless transceiver.

The primary goal of a WSN is to collect useful information such as battlefield surveillance, habitat monitoring [8] [21], disaster rescue, and traffic tracking. Data dissemination protocols are the ways to distribute queries and data among the sensor nodes. They can be classified into three cases [6] [10]: from source nodes to sink, among neighboring sensors, and from sink to sensors.

(i) From source nodes to sink

When source nodes detect or monitor events, they determine the sink's location and then forward the data to the sink. There are two kinds of applications, event driven and time driven. In event driven, source nodes operate in a silent monitoring state and are programmed to notify about events. In the example shown in Figure 1.1(a), only the source nodes whose readings are above 40^oC will send data to the sink. In time driven, source nodes are always monitoring the physical environment and continuously report measurements to the sink. In the example shown in Figure 1.1(b), all source nodes send temperature at every 30 minutes.



(a) Event driven (b) Time driven

Figure 1.1: From Source Nodes to Sink

(ii) Among neighboring sensors

Sensor data dissemination often happens in some kinds of cooperation requiring information exchange among nodes, like clustering techniques [3] [4] [5] [9] [18]. These clustering techniques associate each sensor with some weight. The weight is calculated from a sensor's local properties, such as node ID, temperature, node degree and residual energy. Initially, sensors broadcast their weight to their neighbors (Figure 1.2). Cluster heads (CHs) are selected from those nodes that have the highest weight in their neighborhoods. After CH election, these cluster heads choose their member respectively to construct clusters in WSNs.

(iii) From sink to sensors

Sink data dissemination is normally used to disseminate some information to sensor nodes. The information makes monitoring node have to perform some specific activities, such as to change the operational mode of part or the entire WSNs, activate/deactivate one or more sensors, and send queries to the network. Figure 1.3 illustrates a sink sends queries to the entire WSNs.

Data dissemination in WSNs has the following challenges:

1. When a source node wants to disseminate a new event to sink, it has to get the sink's information (ex: sink id, sink's location, sink's query, etc...) to determine where to forward the event data. Since not

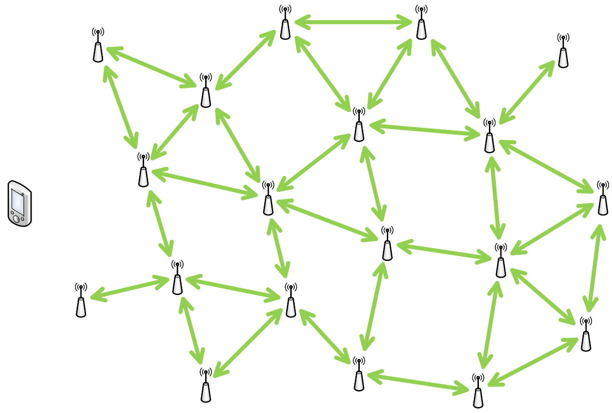


Figure 1.2: Among Neighboring Sensors

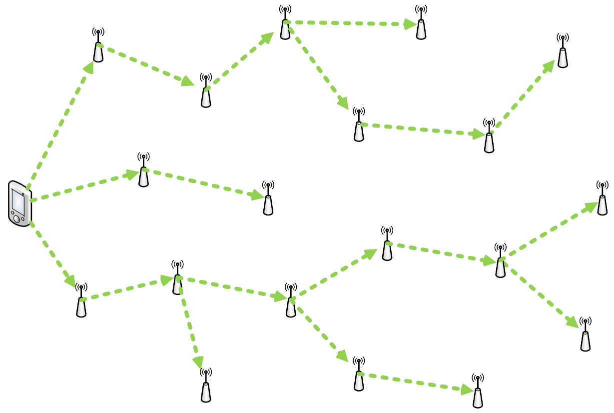


Figure 1.3: From Sink to Sensors

every sensor needs the sink's information, disseminating the sink information to all sensor nodes in [10] will increase messages of WSNs and consume much energy of sensors. So we can select a set of nodes that are most suitable for disseminating information to decrease unnecessary messages. In this way, sink can disseminate some information to the selected nodes rather than all sensor nodes, and source can easily perform some specific activities by these nodes.

2. Many data dissemination schemes like Two-Tier Data Dissemination (TTDD) [26], Scalable Energy-efficient Asynchronous Dissemination (SEAD) [14], and Hierarchical Data Dissemination Scheme (HDDS) [22] require all sensors equipped with Global Positioning System (GPS) devices to build a unique routing structure for each source. Unfortunately, GPS has the following limitations: (a) GPS cannot work indoors or in the presence of dense vegetation; (b) the high power requirements of GPS components drains out the battery of a sensor node pretty fast; and (c) the cost of GPS components can become the most deterrent factor for large scale deployment of WSNs. Thus, we should construct a specific structure to discover and maintain the routing paths without the help of location information for distributing data to the mobile sink.

3. In many situations, a static sink may be unfeasible because of deployment or security constraints. Sink mobility may also improve the lifetime of a WSN [23] [24]. For example, sink mobility could avoid excessive transmission overhead at sensor nodes that are close to the location that would be occupied by the static sink. The sink mobility may be useful for many applications, such as target tracking, emergency preparedness, and habitat monitoring. In these applications, the difficulty is for sensor nodes to efficiently track the sink and report the event data. Hence, the mobile sink and sensors can make use of a virtual infrastructure formed by the set of storing nodes over the physical network for routing, data aggregation and data dissemination. As in [19] [20] [28], infrastructure acts as a rendezvous area for queries and data reports. For example, once a sensor detects an event, a data report is sent towards the rendezvous area. To acquire the data, a sink sends queries instead of to the entire WSN only to the nodes in the rendezvous area, and then the sensor in the rendezvous area sends the requested data back to the sink.

In this paper, we propose a Hierarchical Role-based Data Dissemination approach (HRDD) for providing a scalable and energy-efficient data dissemination with multiple mobile sinks in WSNs. A source is a sensor node that detects a stimulus (e.g. a target or an event of interest) and generates data to report the stimulus. A sink is a device that collects these data reports from the sensor network. Both the number of sinks may vary over time. Since hierarchical structure has significant saving in total energy consumption of the wireless sensor network [2], we use clustering techniques to build a hierarchical structure that each mobile sink can easily maintain its data dissemination paths. After clustering techniques, CHs form a virtual backbone and may be used to route packets for nodes in their clusters. These CHs are called low-level CHs and their clusters are called low-level clusters. Then, these low-level CHs organize themselves into high-level clusters and select their high-level CHs. Furthermore, the node whose one of neighbor nodes doesn't belong to the same low-level cluster is called border node, and the low-level cluster whose one of neighbor clusters doesn't belong to the same high-level cluster is called border cluster. In hierarchical cluster-based protocols, creating clusters and assigning special tasks to some CHs can greatly contribute to overall system scalability and lifetime. The high-level CHs save their cluster members' information, and other low-level CHs only save their local information. After that, high-level CHs assign two roles, called Indexing Agent and Gateway Agent to a set of nodes for storage and routing, respectively. Indexing Agents which are the border nodes in the high-level CH's low-level cluster (Figure 1.4. the red nodes) are like rendezvous area to save the event messages of neighboring low-level clusters, and Gateway Agents which are the border clusters belong to high-level cluster (Figure 1.4. the green nodes) allocate broadcasting path to other high-level clusters. When a source detects an event, the sensing data is stored to local CH. After that, local CH informs its Indexing Agent which local CH belongs to with an event message (Figure 1.4. the pink path). The event message includes meta-data such as CH's ID and event type. As a sink issues a

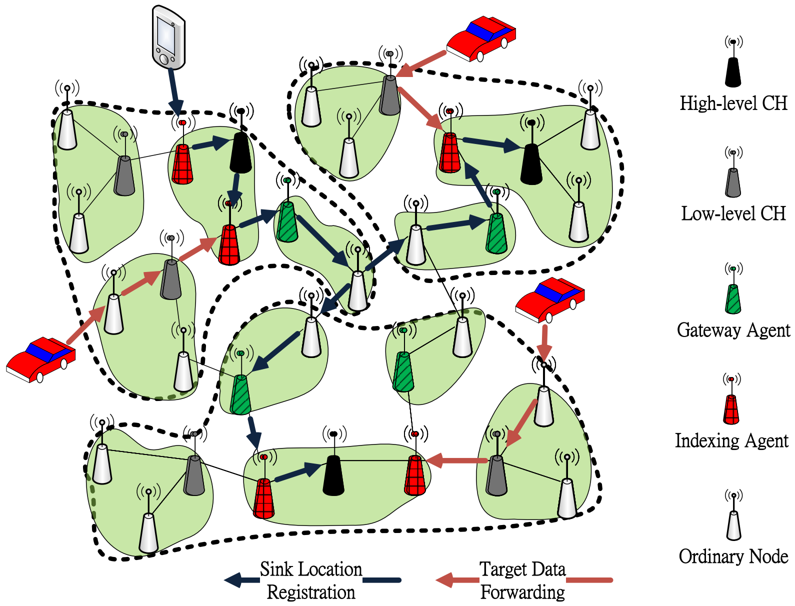


Figure 1.4: An example of the HRDD in WSN

query, it must register its location information to the nearest high-level CH first, and then the local high-level CH broadcasts sink's registration message to other high-level clusters through Gateway Agent (Figure 1.4. the blue path). After registering, high-level CHs local query forwarding to their Indexing Agents. If there is a Indexing Agent which has a relevant data, the Indexing Agent generates a query message and forwards it to CH which stores the source data. Once a CH receives a query message, the CH sends the source data messages by the reverse path to the sink.

In HRDD, the data dissemination challenges in WSNs can easily be solved. First, because of using clustering technique to build a hierarchical structure, we can select high-level CHs for disseminating and saving sink's information. In this way, sink can disseminate some information to high-level CHs rather than all sensor nodes, and source can easily perform some specific activities by high-level CHs. Second, we use a hierarchical cluster-based structure to discover and maintain the routing paths for distributing data to the mobile sink. Since the formation of the hierarchical cluster-based structure does not need the location information of each node, each node only has to locally exchange some information with its neighboring nodes to build a hierarchical cluster-based structure without the location information. Third, some nodes are selected as Indexing Agents which operate as a rendezvous area for events and queries. HRDD stores event messages in Indexing Agents, and sinks retrieve relevant data by searching Indexing Agents. Hence, Indexing Agents can intermediate communication among mobile sinks and sources. Simulation results show that the proposed Hierarchical Role-based Data Dissemination (HRDD) reduces the energy consumption of the WSNs, achieves longer network lifetime, and outperforms Hierarchical Cluster-based Data Dissemination (HCDD) [16] with more 50% total consumed overhead.

The rest of this paper is organized as follows. Section 2 introduces several related works. We depict HRDD in Section 3. Section 4 describes the performance evaluation. Finally, Section 5 draws the conclusions.

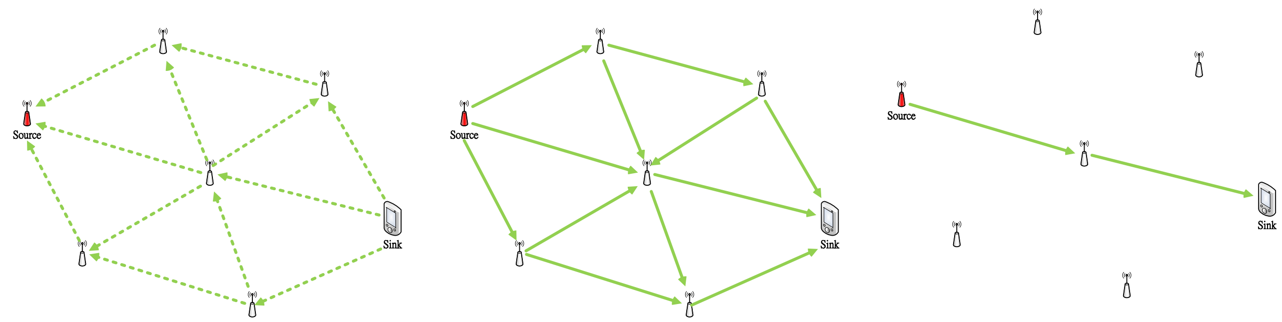
Chapter 2

RELATED WORKS

In large-scale WSNs, scalable and energy-efficient data dissemination schemes are very critical with mobile sinks. This is, because the global flooding and frequently location updating of mobile sinks are energy consuming. Therefore dissemination schemes in WSNs are dedicate to individual nodes to provide a particular structure (rendezvous area) for sinks to reach all sensor nodes. In this section, we dedicated to the study of energy-efficient protocols for data dissemination on the WSNs with mobile sinks. The related works are divided into three categories: source-based approaches [12] [14] [26], index-based approaches [19] [20] [28], and hierarchical-based approaches [16] [22].

Directed Diffusion (DD) uses data-centric routing and data dissemination based on the names described by attribute-value pairs [12]. After a sink broadcasts its interest for a certain type of data (Figure 2.1(a)), the nodes in an area of interest start setting up the gradients which indicate where the replies should be sent (Figure 2.1(b)). The matched data are forwarded back to the sink along the gradients (Figure 2.1(c)). Also, the sink must refresh and reinforce the interest when it starts to receive data from the source. However, the sinks periodically propagate their interests and the flooding messages may result in network congestion.

Two-Tier Data Dissemination (TTDD) [26] provides scalable and efficient data delivery to multiple mobile sinks, as shown in Figure 2.2. Each data source in TTDD proactively builds a grid structure which enables mobile sinks to continuously receive data on the move and to flood queries within a local cell only. Each source transmits data along the nodes on a grid to a sink. However grid construction for each source node and local query flooding may consume much energy. Also, TTDD frequently resumes the entire path to the sinks.



(a) Interest propagation (b) Initial gradients set up (c) Data delivery along reinforced path

Figure 2.1: Directed Diffusion

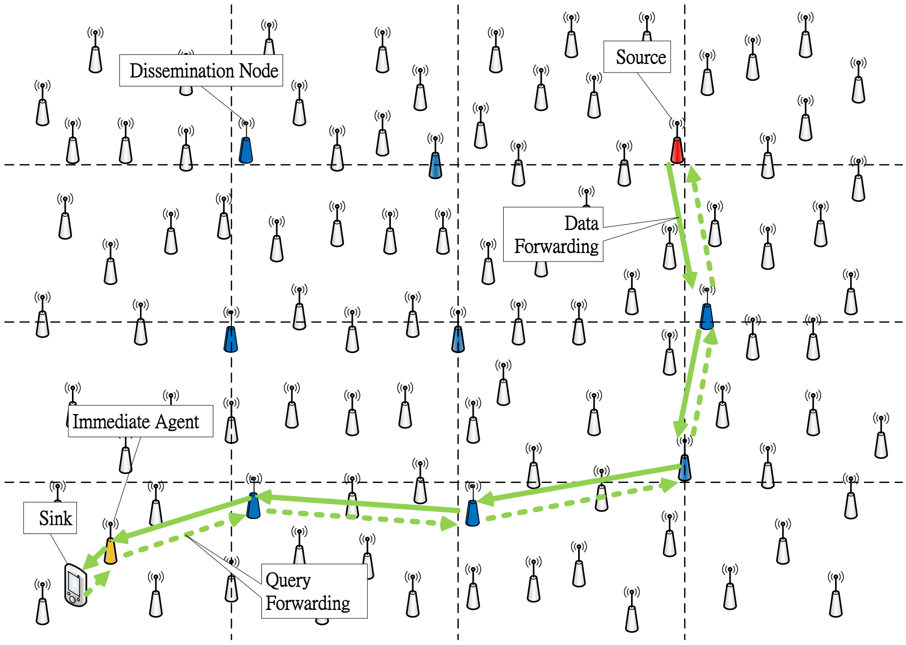


Figure 2.2: Two-Tier Data Dissemination (TTDD)

Scalable Energy-efficient Asynchronous Dissemination (SEAD) [14] considers the distance and the packet traffic rate among nodes to create near-optimal dissemination trees and designates some nodes on the tree as the access nodes, as shown in Figure 2.3. Each mobile sink sends the registration to the access node which is closest to mobile sink, and the access node receiving the registration should forward data to registered mobile sinks. When the sink moves out of the coverage range of the corresponding access node, it should dynamically reregister itself to alternative access node and adjust the tree adaptively. In addition, source data is replicated at selected nodes between the source and sinks. These selected nodes,

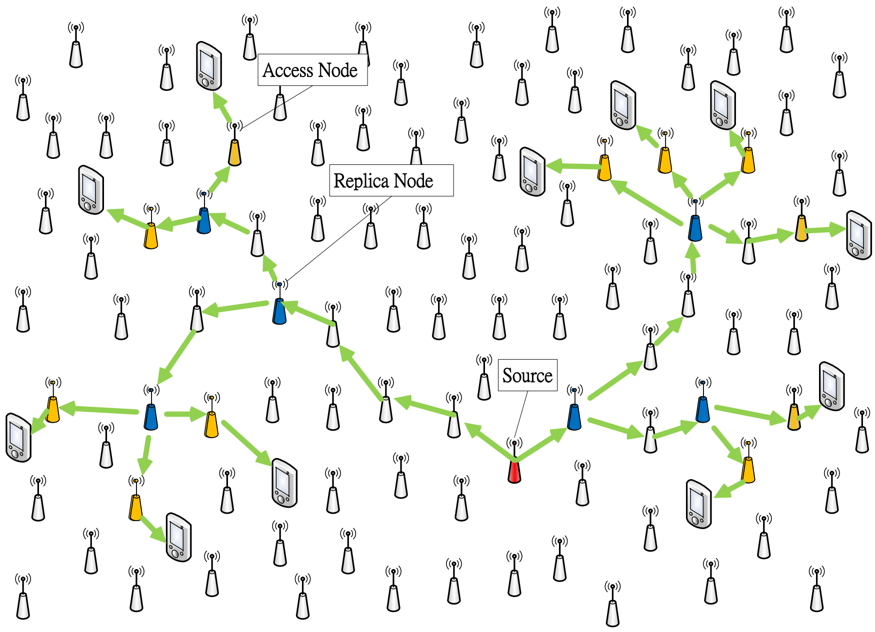


Figure 2.3: Scalable Energy-efficient Asynchronous Dissemination (SEAD)

called replica nodes, store a copy of the source data. The replica node temporarily stores the latest data incoming from the source and asynchronously disseminates it to others along the tree. However, like the TTDD, the SEAD also has to construct separate dissemination trees for multiple sources.

In Data-Centric Storage-based (DCS) [19] scheme, events to be detected are named, and the sensing data of these events are stored at nodes within the network instead of being forwarded to an external storage. As shown in Figure 2.4, the storing node of an event is the node closest to a location calculated by applying a hash function on the event type. No matter whether sensing data are pushed to the storage outside of the network or that within the network, all the data must be transferred regardless of whether they are used by the applications. Hence, they lack flexibility and may introduce lots of unnecessary data transfers.

In Adaptive Ring-based Index (ARI) [28] scheme, as shown in Figure 2.5, the sensing data of an event are stored at the detecting nodes themselves or some nodes close to them (called storing nodes). A storing node sends data to a sink only when it receives a query from the sink. Also, the location information (called index) of the storing nodes is pushed to and maintained at some nodes (called index nodes) based on the event type of the stored data. In ARI, multiple nodes are picked as the index nodes for each event type, and these index nodes are connected with each other to form a ring (called the index ring). The number and locations of the index nodes on an index ring, as well as the shape of the

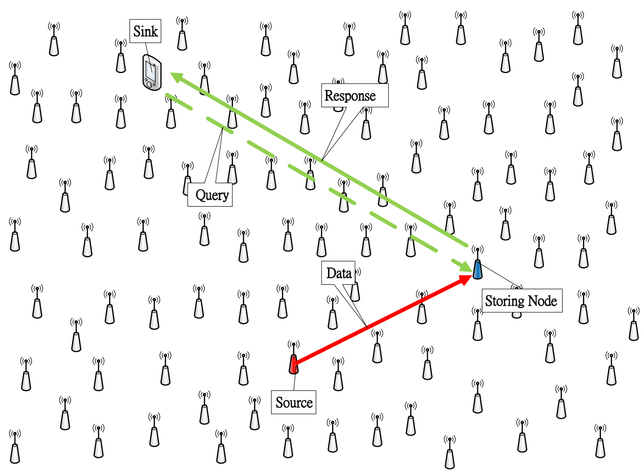


Figure 2.4: Data-Centric Storage Scheme

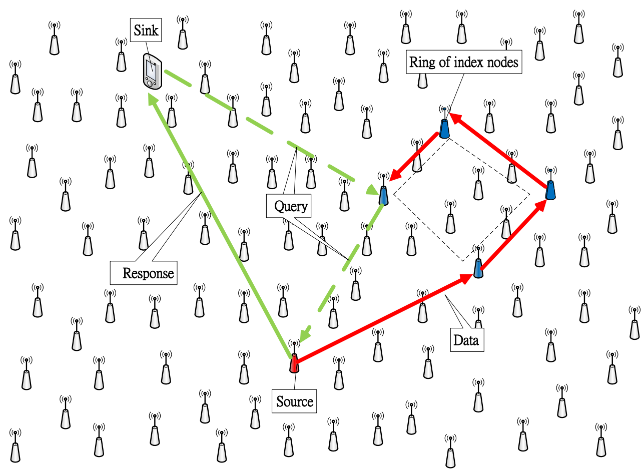


Figure 2.5: Adaptive Ring-based Index Data Scheme

ring, can be adaptively changed to tolerate node failures and achieve load balance. Hence, queries for a particular event are directly routed to the appropriate index nodes, which may then forward the queries to appropriate storing nodes. This scheme avoids both unnecessarily transferring the sensing data and flooding control messages throughout the network.

Railroad [20] builds a virtual infrastructure, called a rail, as shown in Figure 2.6. This rail is placed

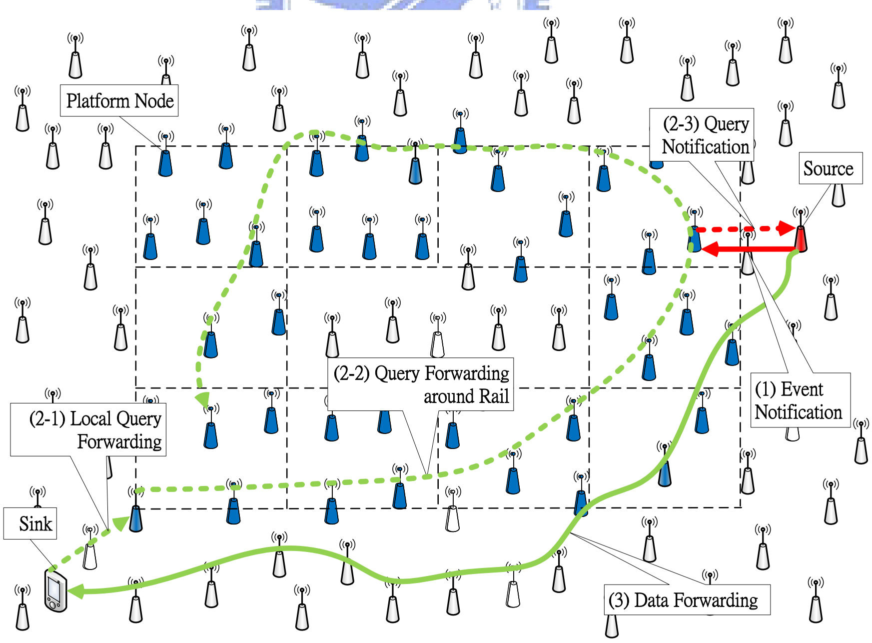


Figure 2.6: Railroad

in the middle area of the sensor field so that each node can easily access it. When a source detects an event, it stores the data and forwards a notification to the nearest neighbor toward Rail (1). When a sink node issues a query, the query is forwarded to the sources in three phases. Each node on the forwarding path transfers the query to the nearest node to Rail (2-1). Once the query enters Rail, it is forwarded along Rail until it reaches the entered point (2-2). If there is a station which has a relevant data, the corresponding platform node generates a query notification message and forwards it to the source node (2-3). Immediately after the reception of a query notification message, the source sends the relevant data messages directly to the sink (3). However Railroad presents a rather high path-stretch which is almost twice the optimal path. This is a direct consequence of the query path length.

In Hierarchical Data Dissemination Scheme (HDDS) [22], as shown in Figure 2.7, source routes

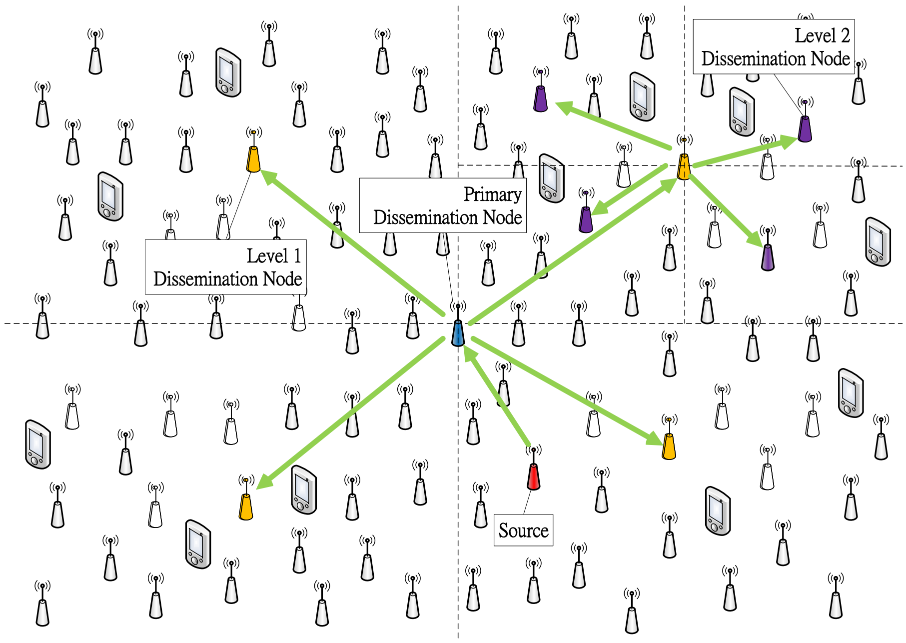


Figure 2.7: Hierarchical Data Dissemination Scheme (HDDS)

data towards sinks using a hierarchy of randomly selected dissemination nodes. Because dissemination nodes have limited resources, whenever a dissemination node is overloaded, it inserts another level of dissemination nodes to reduce its load. For an energy-efficient communication scheme, data forwarding path is close to the shortest route since this path optimization is essential for prolonging lifetime of sensor networks. HDDS follows a data transmission policy that forwards data to the forwarding agent directly. Thus, data may take a shorter path, and total energy consumption and delay can be reduced. However, in TTDD, SEAD, DCS, ARI, Railroad, and HDDS, each node should acquire the location information for data dissemination. TTDD, SEAD, and HDDS use location information to construct specific structure for source to transfer event data to sinks, and Railroad, DCS, and ARI select certain nodes for queries and data reports with location information. Once the location information doesn,t work well in these schemes, sources cannot build a routing structure to send event data for multiple sinks.

Hierarchical Cluster-based Data Dissemination (HCDD) [16], proposes a hierarchical cluster-based structure to discover and maintain the routing paths for distributing data to the mobile sink, as shown in Figure 2.8. In HCDD, all nodes distributed build a cluster structure without the location information,

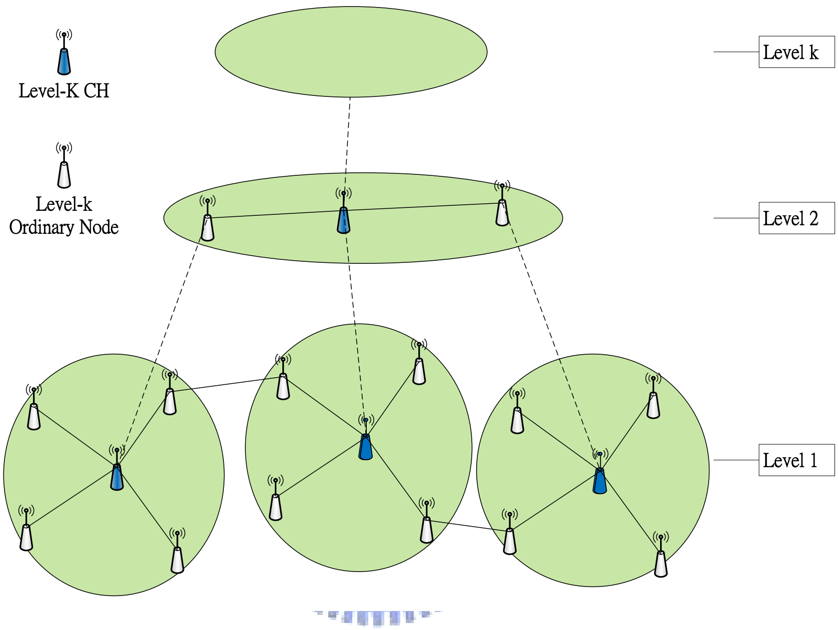


Figure 2.8: Hierarchical Cluster-based Data Dissemination (HCDD)

so each node only has to locally exchange the information with its neighboring nodes. There are three procedures in HCDD. First, in Cluster Construction, all nodes are divided into multilevel clusters by Max-Min *D*-Cluster Formation Algorithm [3], and each cluster will designate a node as the CH. After Cluster Construction, all of CHs in the highest hierarchical level CHs, called Routing Agent, should keep Dynamic Global Information, i.e. the sink information and routing information. Second, in Sink Location Registration, each sink has to register at one of the Routing Agents, which are responsible for the management of the sink information. The sink information should only be broadcast to Routing Agents, but not flooded to all sensor nodes. Finally, in Data Delivery and Path Routing, CHs and Routing Agents cooperate to find the paths from data sources to the sink by the inter-cluster routing and the intra-cluster routing. However, in the HCDD scheme, the broadcasting of Sink Location Registration increases energy consumption and the number of messages. In addition, HCDD is not suitable for multiple mobile sinks. Table 2.1 compares the existing approaches described in section 2 according to these different criteria we have just presented.

Protocol	How to send sink's query?	Equip positioning device or not?	Exist rendezvous area or not?
DD [12]	Flooding [15]	no	no
TTDD [26]	Grid structure	yes	no
SEAD [14]	D-tree structure	yes	no
DCS [19]	GPSR [13]	yes	yes
ARI [28]	GAF [25] and GPSR	yes	yes
Railroad [20]	Rail structure	yes	yes
HDDS [22]	Hierarchical structure	yes	no
HCDD [16]	Hierarchical structure	no	no
HRDD	Hierarchical structure	no	yes

Table 2.1: Data Dissemination Protocols

In this paper, we propose the Hierarchical Role-based Data Dissemination (HRDD) approach to assign two roles, called Indexing Agent and Gateway Agent in the WSNs. Indexing Agents which are like rendezvous area for source data reports and sink queries could avoid unnecessarily transferring the query messages, and Gateway Agents could decrease energy consumption of broadcasting and number of flooding messages. The objectives of our work are to (1) avoid using location information to construct cluster structure and dissemination path, (2) build an energy-efficient data dissemination for WSNs to reduce registration messages, and (3) construct a rendezvous area that is convenient for source data reports and sink queries.

Chapter 3

HIERARCHICAL ROLE-BASED DATA DISSEMINATION

3.1 Overview of the HRDD

We assume that a wireless sensor network has the following properties:

- After randomly deployment, sensor nodes remain stationary at their initial locations.
- The sensor nodes are homogeneous and location unaware.
- The sensor nodes can make decisions without global information.
- There are multiple mobile sinks which generate queries in the WSNs.
- The sensor nodes communicate with sinks by delivering data across multiple hops.

The hierarchical structure can reduce communication overhead and data redundancy in the sensor networks. We use clustering technique to build a hierarchical structure that each mobile sink can easily maintain its data dissemination path. Figure 3.1 illustrates the fundamental concept of a cluster hierarchy, and the number in the figure expresses the ID of sensor node. All sensor nodes organize themselves into low-level clusters via a CH election process [3] [4] [5] [9] [18]. The low-level clusters, in turn, organize themselves into high-level clusters.

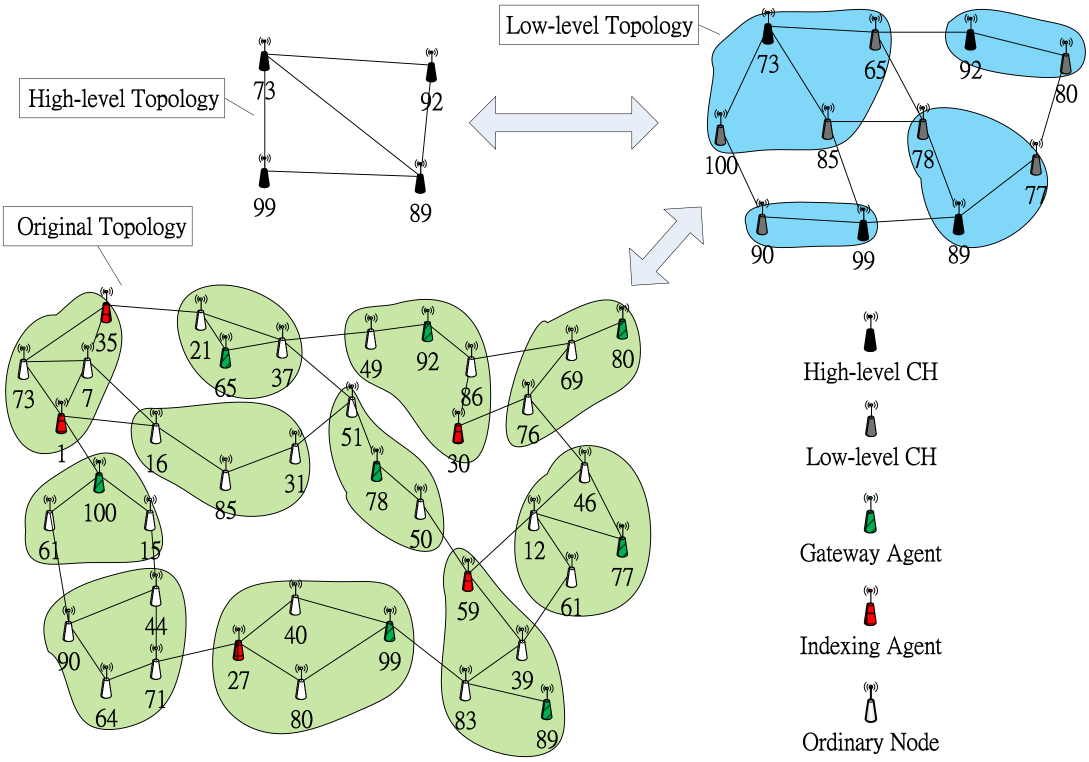


Figure 3.1: Example of Hierarchical Role-based Data Dissemination (HRDD)

In HCDD, when a sink registers its location information, sink's high-level CH will broadcast registration message to other high-level clusters through border clusters as shown in Figure 3.2. Since one of border clusters may overlap the neighboring high-level clusters of another border cluster, high-level CH may get reduplicate sink's registration messages. It will generate redundant messages and make entire sensor network consume much energy. In addition, as a sink issues a query, each high-level CHs local query forwarding to their low-level CHs. In the example shown in Figure 3.3, since the sensing data are stored to low-level CH, each high-level CHs must forward query to them to get the relevant data. However there may not be any relevant data in the low-level CH, it will consume much energy caused these unnecessary messages. Since the registration of mobile sink's location increases much energy consumption and lots of messages, and data delivery from source to sink may induce unnecessary query messages. So, we assign two roles, Indexing Agents and Gateway Agents to some nodes. Indexing Agents, which are the border nodes in high-level cluster (Figure 3.5. the red nodes), should store the event messages of neighboring low-level clusters, and Gateway Agents, which are the border clusters belong to high-level cluster (Figure 3.4. the green nodes), should allocate broadcasting path

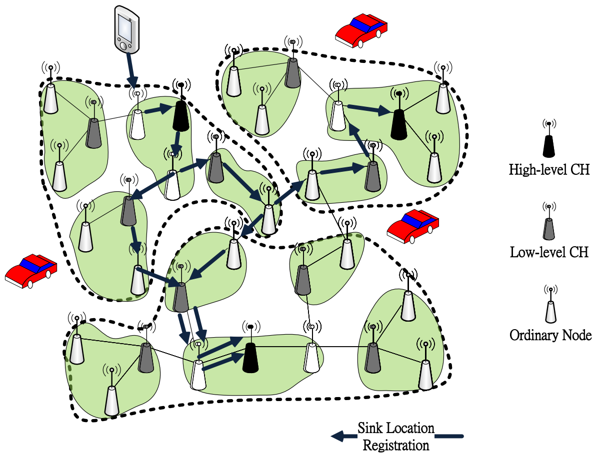


Figure 3.2: Sink Location Registration in HCDD

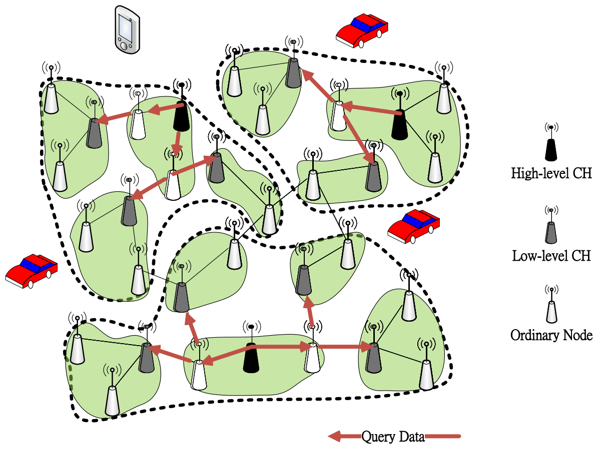


Figure 3.3: Query Data in HCDD

to other high-level clusters. Once a source detects an event, the sensing data are stored to its low-level

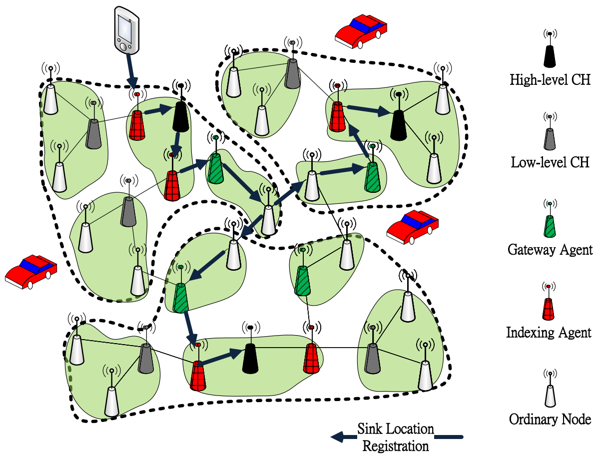


Figure 3.4: Sink Location Registration in HRDD

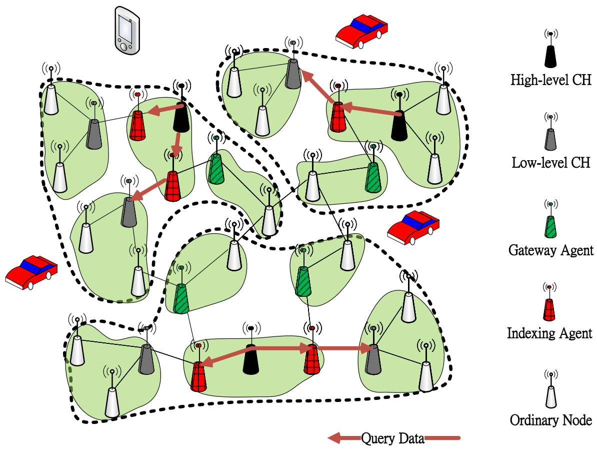


Figure 3.5: Query Data in HRDD

CH and then an event message is forwarded to the Indexing Agent which the low-level CH belongs to. When a sink issues a query, it must broadcast its registration message to all high-level clusters through Gateway Agent for source data forwarding. In this way, sink can easily get the relevant data by querying Indexing Agents, instead of querying to all low-level CHs, and sink lightly sends its registration message direct through Gateway Agent to other high-level clusters, not through the border clusters belong to high-level cluster. Hence, sink queries by Indexing Agents could avoid unnecessarily transferring the query messages between source and sink for data lookup, and sink registration through Gateway Agent could decrease energy consumption of broadcasting and number of flooding messages.

There are five procedures in HRDD. First, in Cluster Construction, all nodes are grouped into multilevel clusters, and each cluster will elect a node as the CH. Second, in the Selection of Indexing Agent and Gateway Agent, each high-level CH has to select a set of nodes as Indexing Agents or Gateway Agents by Agent Selecting Algorithm. Third, in Event Detection, the sensing data is stored to its low-level CH and an event message send to Indexing Agent for events and queries. Fourth, in Sink Location Registration, each sink registers its location information to all high-level CHs through Gateway Agent, Finally, in Query Data Forwarding, when an Indexing Agent has relevant data which a sink queries, the data will be forwarded to the sink by high-level CHs and low-level CHs along the reverse path. The detail of each procedure will be presented in the following subsections.

3.2 Cluster Construction

There are a number of clustering techniques been proposed in the literature [3] [4] [5] [9] [18]. In the Linked Cluster Algorithm (LCA) [4], a node becomes the CH if it has the highest identity among all nodes within one hop of itself or among all nodes within one hop of one of its neighbors. The Distributed Clustering Algorithm (DCA) [5] uses generic application-dependent weights associated with nodes to elect CHs. The Weighted Clustering Algorithm (WCA) [9] elects a node as a CH if it has the highest weight among its one-hop neighbors based on a combination of node characteristics that include node degree, transmission power, mobility and batter power of node considered. The Voting-based Clustering Algorithm (VCA) [18] votes for their neighbors to elect suitable CHs based on the local information of each node, such as residual energy and node degree.

All of the above algorithms [4] [5] [9] [18] generate one-hop clusters, require synchronized clocks and have $O(n)$ complexity, where n is the number of nodes in the network. This makes them suitable only for networks with a small number of nodes. The Max-Min D-cluster algorithm proposed in [3] generates D hops clusters based on node-ids with a run-time of $O(D)$ rounds and only $O(D)$ messages per node, where the value of D is selected by users. Hence, in our work, we group the sensors by Max-Min D-cluster algorithm, which formed by node IDs along the wireless links and can provide load-balanced clustering for extending the lifetime of the WSNs. Max-Min D-Cluster Formation Algorithm consists of a distributed CHs election algorithm, guaranteeing that no node is more than D hops away from its CH. The algorithm has four logical stages: The first stage uses D rounds to propagate the largest node ID which nodes heard from other neighbor nodes. The second stage uses D rounds to propagate the smallest node ID which nodes heard from other neighbor nodes. In the third stage, each node selects its CH based on the information received in the first and second stages. Finally, each non-CH node communicates with its CH to join the CH's cluster in the fourth stage. After the Max-Min D-cluster algorithm, CHs form a virtual backbone and may be used to route packets for nodes in their cluster. These CHs are called low-level CHs. Then, Max-Min D-cluster algorithm is performed on the low-level CHs to form high-level clusters.

3.3 The Selection of Indexing Agent and Gateway Agent

In HRDD, Indexing Agents are the rendezvous area to save the event messages of neighboring low-level clusters, and Gateway Agents allocate broadcasting path to other high-level clusters. After Cluster Construction, the high-level CHs are aware of local information, i.e. the information of low-level clusters and the neighboring high-level clusters. They use these local information to select Indexing Agents and Gateway Agents by Agent Selecting Algorithm. The main idea of Agent Selecting Algorithm is that selecting a set of nodes plays two roles. These nodes take place in the border nodes of high-level cluster or the border clusters of high-level cluster. In this way, when a sink issues a query, it is relaxed to communicate to other high-level clusters through Gateway Agents and is easily to query the relevant data by Indexing Agents, rather than using heavily broadcasting to other high-level clusters and searching hardly for all the low-level clusters of high-level CH.

There are two input parameters for Agent Selecting Algorithm, one is agent candidates, another is agent candidates' neighboring clusters. For Indexing Agents, agent candidates are the border nodes in the high-level CH's low-level cluster (in Figure 3.6. high-level CH 73's Indexing Agent candidates are node 1, 7, and 35), and neighboring clusters are the neighbor low-level cluster of high-level CH's low-level cluster (in Figure 3.6. high-level CH 73's neighboring low-level clusters are cluster 65, 85, and 100). For Gateway Agents, agent candidates are the border clusters belong to high-level cluster (in Figure 3.6. high-level CH 73's Gateway Agent candidates are cluster 65, 85, and 100), and neighboring clusters are the high-level CH's neighbor high-level clusters (in Figure 3.6. high-level CH 73's neighboring high-level clusters are cluster 89, 92, and 99).

The Agent Selecting Algorithm consists of the following two phases:

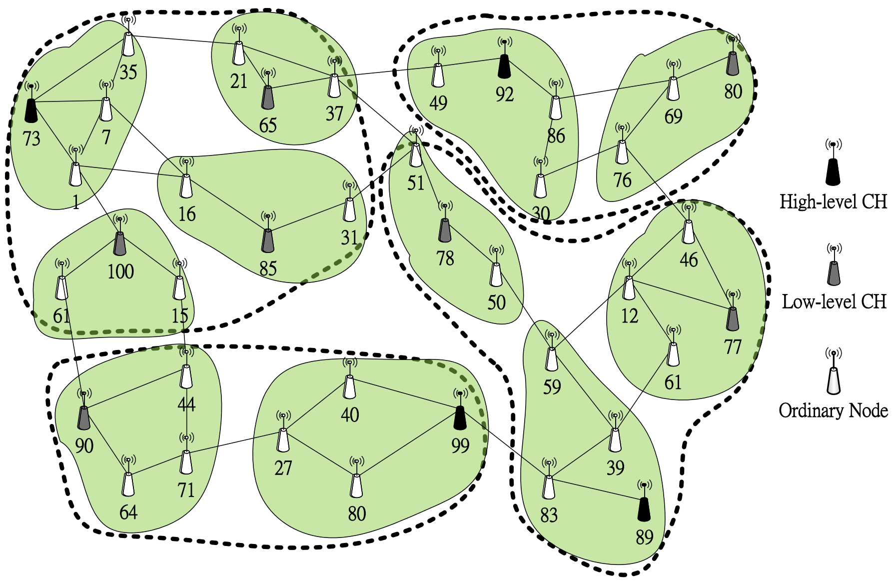


Figure 3.6: Example of selecting Agents

Phase I: Agent table sets up

- The high-level CHs set up the relationship between agent candidates and neighboring clusters, and then build the agent table with these relationships. In Figure 3.7, the Indexing agent candidates of high-level CH 73 are node 35, 7, and 1. Their neighboring low-level clusters are, respectively, cluster 65, cluster 85, and cluster 85, 100. According to these relationships, high-level CH 73 sets up the agent table for next phase to select the Indexing Agents.

High-level CH 73	
Agent candidate	Neighboring cluster
35 →	65
7 →	85
1 → (85, 100)	100

Figure 3.7: High-level CH 73's Indexing Agent table

Phase II: Agent selecting

- Figure 3.8 illustrates the example of the Agent Selecting Algorithm for Indexing Agents in High-level CH 73. There are five steps to select agents.

Step 1: Transform the agent table into an array.

- The high-level CH transforms the agent table into an array which agent candidates are column of array and neighboring clusters are row of array.
- If there is a neighboring low-level cluster is next to one of agent candidates, the high-level CH sets 1 to array. If not, set 0 to array (Figure 3.8(a)).

Step 2: Select an agent.

- The high-level CH counts number of 1 in every row first.
- After that, the high-level CH selects a row which has the maximum number of 1 and records it (Figure 3.8(b)).

Step 3: ANDNOT operation.

- When high-level CH records a row which has the maximum number of 1, it uses this row to do ANDNOT operation for the entire row of array (Figure 3.8(c)).

Step 4: Repeat agent selecting.

- After ANDNOT operation, high-level CH repeats the step 2 to the step 3 (Figure 3.8(d)).

Step 5: Final agent decision.

- If high-level CH finds that all elements of the array is set 0, it ends the Agent Selecting Algorithm (Figure 3.8(e)).
- The results of high-level CH 73's Indexing Agents are node 1 and node 35 (Figure 3.1. the red nodes in high-level CH 73). The Agent Selecting Algorithm for Gateway Agents is the same as above.

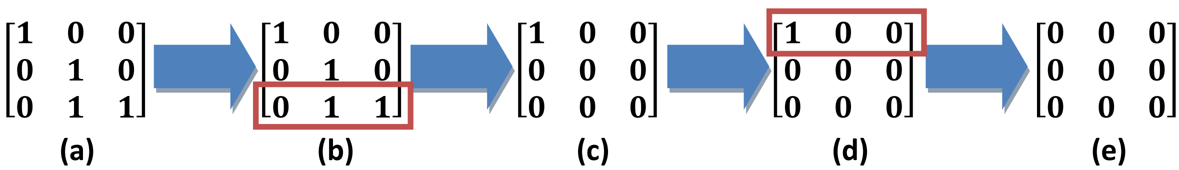


Figure 3.8: Agent selecting for high-level CH 73's Indexing Agents

When the Agent Selecting Algorithm is end, high-level CH sends ROLE message to those nodes or clusters which are selected to be agent. Once some nodes get ROLE message which is Indexing Agent message, they will start to collect the information of neighboring low-level clusters which they are responsible for. Another some nodes get ROLE message which is Gateway Agent message, they will wait for registration messages and then transmit these messages to neighboring high-level clusters.

3.4 Event Detection

HRDD proactively exploits Indexing Agents to support target mobility as shown in Figure 3.9. When a source detects an event, the sensing data are sent towards the local CH. After that, local CH informs its Indexing Agent which local CH belongs to with an event message which includes meta-data such as CH's ID and event type (1). When the an event moves, the source is changed accordingly. However, the new source still stores the sensing data at the same local CH (2) until the source belongs to different local CH, and then the sensing data is sent towards the new local CH. When an old source found that an event disappeared, the old source will delete the sensing data of the target and informs its Indexing Agent. Also, the new local CH should inform its event message at the Indexing Agent which it belongs to for the event (3 and 4). When a sink wants to query the detailed sensing data of an event, it sends a query message to an Indexing Agent. On receiving the message, the Indexing Agent transmits the request to the corresponding local CH which sends a response to the querying sink.

3.5 Sink Location Registration

The high-level CHs maintain the information of mobile sinks and save previous high-level CH which they receive registration message from in our HRDD. The sink registration is divided into two stages, high-level local CH registration and high-level global CH registration. When a mobile sink issues a query, it first sends registration message to its high-level CH as shown in Figure 3.10 (the pink path). The sink sends registration message to its low-level local CH, node 100. Node 100 in turn transmits the registration message to its high-level CH, node 73. Then, the sink's high-level local CH will send the registration message to all other high-level CHs through Gateway Agents. At the same time, when other high-level CHs receive the registration message in the second stage, they should save upper high-

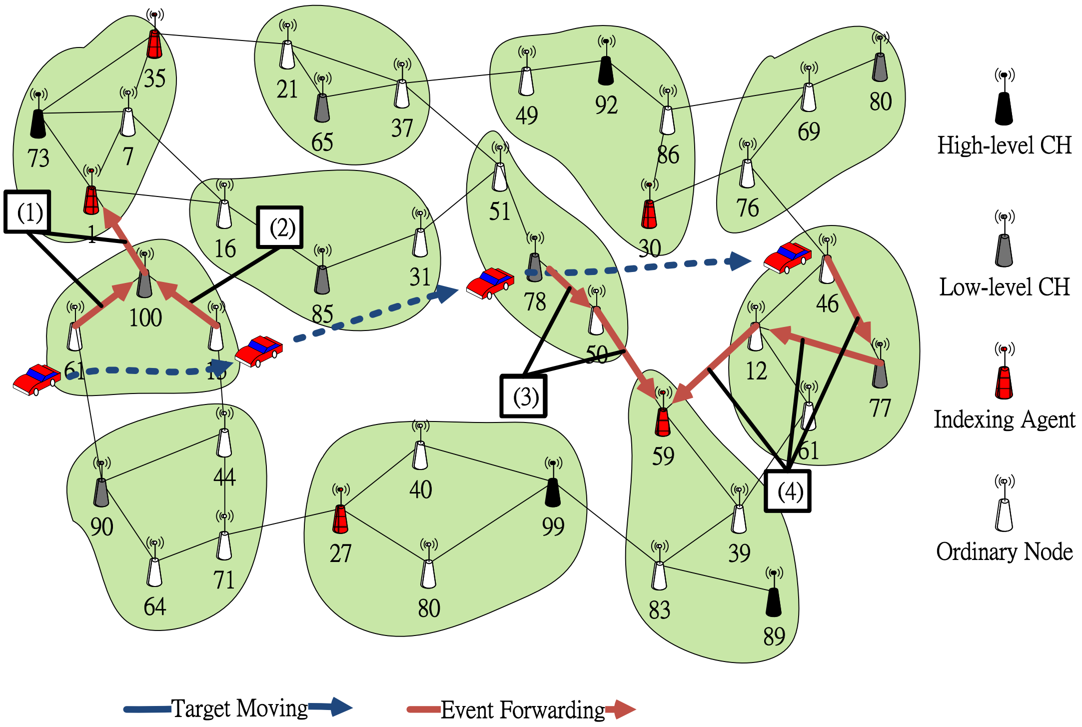


Figure 3.9: Event Detection

level CH which they receive the registration message from to assure that the source data can easily be forwarded to the sink along the reverse path. Figure 3.10 (the blue path) illustrates the example of high-level global CH registration. Node 73 sends the registration message to its Gateway Agents, node 65 and 100. Finally, node 65 and 100 broadcast the registration message to their neighboring high-level clusters, including high-level cluster 89, 92, and 99. If the sink's high-level local CH sends the registration message to other high-level CHs not through Gateway Agents, it may increase unnecessary registration messages. If not through Gateway Agents, node 73 sends registration message to other high-level CHs through its border clusters (node 65, 85, and 100). Since node 85's neighboring high-level clusters (high-level cluster 89 and 99) overlap the node 65 and 100's neighboring high-level clusters, node 85 may send redundant registration messages. Thus, Gateway Agent could decrease energy consumption of broadcasting and number of flooding messages.

If a mobile sink moves out of the range of its high-level local CH, it should redo the Sink Location Registration to ensure the paths to the mobile sink are available. On the contrary, a mobile sink moves in the range of its high-level local CH, it only updates its location information for high-level local CH

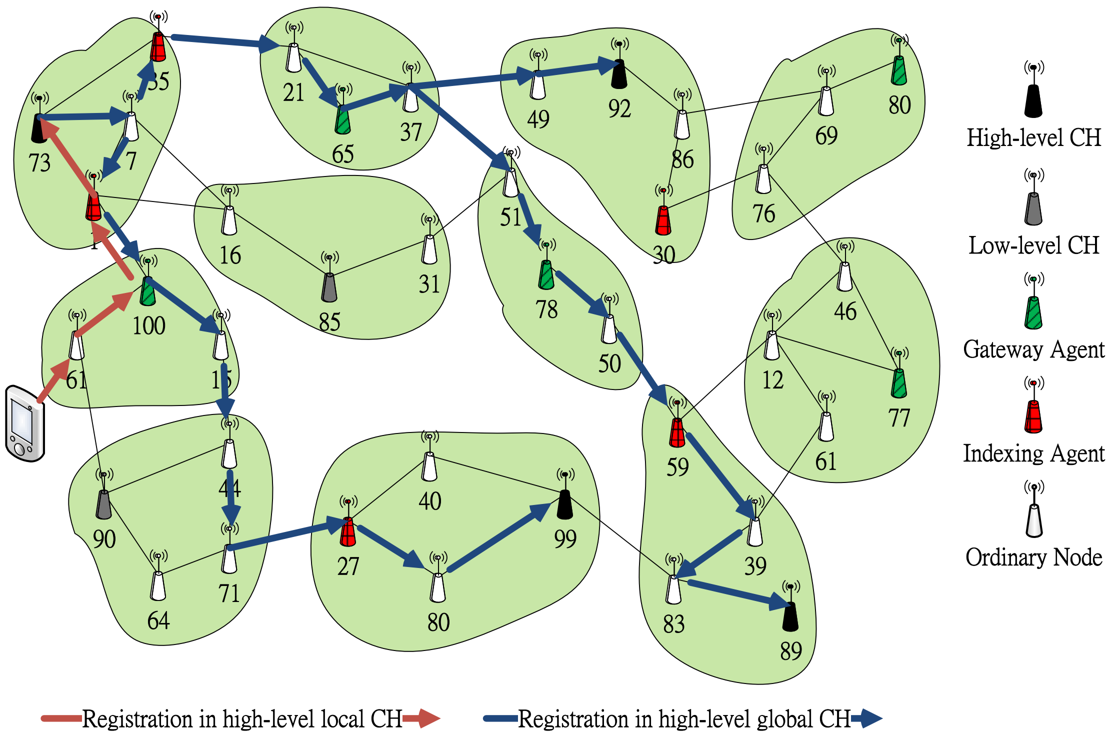


Figure 3.10: Sink Location Registration

and low-level local CH, so that it can decrease unnecessary registration messages.

3.6 Query Data Forwarding

After the Sink Location Registration, each high-level CH starts to proceed data delivery. The data delivery is also divided into two stages, query data searching and query data delivery. In query data searching, each high-level CH forwards sinks' queries to all Indexing Agents to acquire the interesting data. Figure 3.11 (the pink path) illustrates the example of query data searching. If there is a Indexing Agent which has a relevant data, the Indexing Agent sends the request to the corresponding low-level CH which stores the source data, and then the low-level CH proceeds next stage.

In query data delivery stage, the low-level CH first sends event data to its high-level local CH, and the high-level CH forwards event data to the sink's high-level CH. Therefore, high-level CH can relay data by the reverse path to the sink as shown in Figure 3.11 (the blue path). When sink's high-level CH (node 73) receives the event data from other high-level CHs (node 89), and it relays the data to

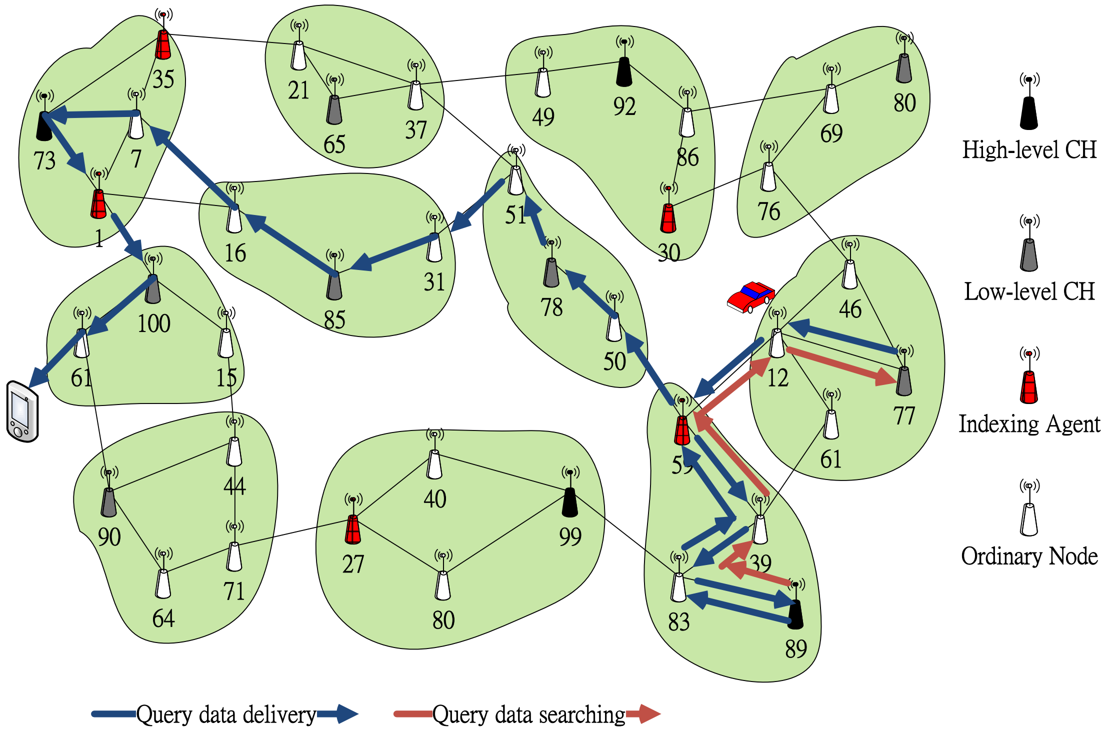


Figure 3.11: Query Data Forwarding

the node 61 which is near the sink. Finally, the sink can get the event data from node 61. However, when a sink issues a query without Indexing Agents, high-level CHs must query forwarding to their low-level CHs to acquire the interesting data. For example, in query data searching stage, the high-level local CH (node 89) which doesn't have any Indexing Agent receives sink queries, it should query forwarding to its low-level CHs (node 77 and 78) to acquire the interesting data. It will increase number of unnecessarily query messages for these low-level CHs in query data searching stage. Hence, Indexing Agent is convenient for data storage and lookup.

Chapter 4

PERFORMANCE EVALUATION

In this section, we evaluate the performance of HRDD through simulations. We first describe our simulator implementation, simulation environment and metrics in Section 4.1. Then we evaluate how environmental factors and control parameters affect the performance of HRDD in Sections 4.2 to 4.4 with the performance of HCDD. The results confirm the efficiency and scalability HRDD to deliver data from sources to multiple mobile sinks.

4.1 Simulation Environmental and Metrics

We develop a simulator based on JSIM [17] to evaluate and compare HRDD to HCDD. In this simulator, the default simulation setting has 6 sinks, 30 targets, and 1500 sensor nodes randomly distributed in a $2500 \times 2500\ m^2$ field. The maximum number of wireless hops between a node and its CH (D) was set to 2. Each simulation run lasts for 18000 seconds, and each result is averaged over three random network topologies. Sinks and targets' mobility follows the standard Random WayPoint Model [7]. Each sink generates queries at every 100 seconds. Each query packet has 76 bytes and each data packet has 59 bytes. We use a simplified model shown in [11] [27] for the radio hardware energy dissipation. Both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models are used in the model, depending on the distance between the transmitter and receiver. The energy spent for transmission of a l-bit packet over distance d is:

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4, & d \geq d_0 \end{cases}$$

The first item presents the energy consumption of radio dissipation, while the second presents the energy consumption for amplifying radio. When receiving this data, the radio expends: $E_{Rx}(l) = lE_{elec}$. Table 4.1 lists most of the simulation parameters.

Parameter	Value
Field size	$1000 \times 1000\ m^2$
Number of nodes	$500, 1000, 1500, 2000, 2500$
Number of sinks	$2, 4, 6, 8, 10$
Number of targets	$10, 20, 30, 40, 50$
Speed of sink and target	$5 - 10\ m/sec$
D hops between a node and its CH	$1, 2, 3, 4, 5$
Query packet size	$76\ bytes$
Data packet size	$59\ bytes$
Initial energy	$2\ J$
E_{elec}	$50\ nJ$
ε_{fs}	$10\ pJ/bit/m^2$
ε_{ap}	$0.0013\ pJ/bit/m^4$
Threshold distance (d_0)	$75\ m$
Query generation rate	100.0 sec
Simulation time for each experiment	18000.0 sec

Table 4.1: Simulation Parameters

The metrics used for evaluating the performance of HRDD are overall message, energy consumption, and network lifetime. Overall message is the amount of packets transmitted with registration packets in Sink Location Registration procedure and data packets in Query Data Forwarding procedure. Energy consumption is the energy consumed of the network in transmitting and receiving a data report. Network lifetime, which is defined as the duration from the beginning of the simulation to the time once one sensor runs out its energy.

4.2 Impact of the Number of Nodes

We first study the impact of number of nodes on HRDD's performance. In this experiment, the number of sensor nodes is varied from 500, 1000, 1500, 2000, to 2500 and Figure 4.1 illustrates the number of high-level CHs with different number of sensor nodes by Max-Min *D*-cluster algorithm. Due to

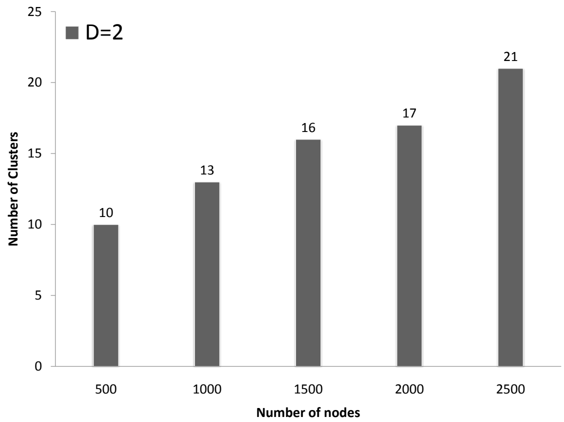


Figure 4.1: Number of high-level CHs with different number of nodes

the more nodes, the number of high-level CHs becomes more. More high-level CHs may increase the generated of registration packets and data packets greatly. Figure 4.2 shows the overall message at different numbers of sensor nodes. In Sink Location Registration procedure, since the usage of Gateway Agent could decrease number of flooding messages, the registration packets in HRDD are less than in HCDD. In Query Data Forwarding procedure, the HRDD is less than HCDD in data packets. When high-level CHs forward sink queries to acquire the interesting data, they forward these queries to Indexing Agents in HRDD rather than forward these queries to low-level CHs in HCDD. Furthermore, HCDD will increase the data packets for forwarding queries to low-level CHs while the number of high-level CHs increases. But in HRDD, the usage of Indexing Agent makes the data packets increase slightly.

Figure 4.3 and Figure 4.4 show the energy consumption and network lifetime respectively. The entire sensor network consumes much energy caused by the increase of registration packets and data packets. This result influences the network lifetime indirectly. Because HRDD has less the generated of registration packets and data packets than HCDD, it has the better performance of lower energy consumption and longer network lifetime.

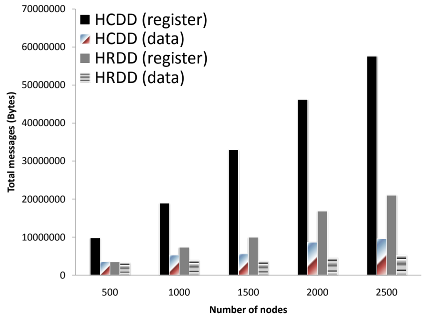


Figure 4.2: Overall message for different number of nodes

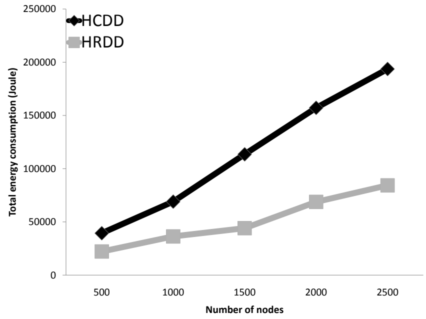


Figure 4.3: Energy consumption for different number of nodes

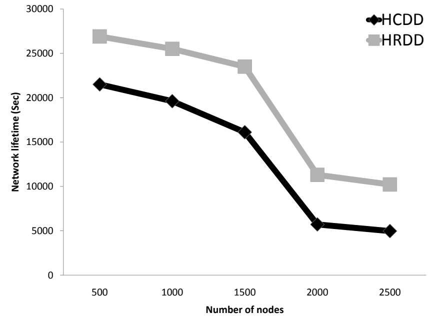


Figure 4.4: Network lifetime for different number of nodes

4.3 Impact of the Number of Sinks and Targets

We study the impact of the number of sinks and targets affect HRDD and HCDD. The number of sinks varies from 2, 4, 6, 8, to 10 and the number of targets varies from 10, 20, 30, 40, to 50. Figure 4.5 shows the overall message at different numbers of sinks. Since the increase of sinks, the generated of registration packets also increases in HCDD and HRDD. Though the number of targets is changeless, the data packets in HRDD and HCDD are slightly increasing with the increase of sinks. In this experiment, the Gateway Agent plays an important role for the increase of mobile sinks. No matter how mobile sinks move or number of sinks increase, sinks register their information through Gateway Agent can reduce the generated of registration packets greatly. Figure 4.6 and Figure 4.7 show the energy consumption and network lifetime with the increase of sinks. In the hierarchical structure, the registration packets all send to high-level CHs. Once the registration packets increase, it does not only consume much energy in the entire sensor network but also increase the overhead of high-level CHs, and then affect the network lifetime. Hence, the usage of Gateway Agent could not only consume less energy but also decrease the overhead of high-level CHs, and then prolong the network lifetime.

In Figure 4.8, Figure 4.9, and Figure 4.10 show the overall message, the energy consumption, and the network lifetime at different number of targets, respectively. Because the number of sinks is fixed, the number of the registration packets is similar at different number of targets. When the number of targets increases, the data packets which are forwarded to sinks will also increase. For energy consumption and network lifetime, since the number of registration packets is similar, the trend of curve is determined by the number of data packets. Though Indexing Agent does not affect energy consumption and network lifetime greatly, it slightly decrease the number of data packets in query data

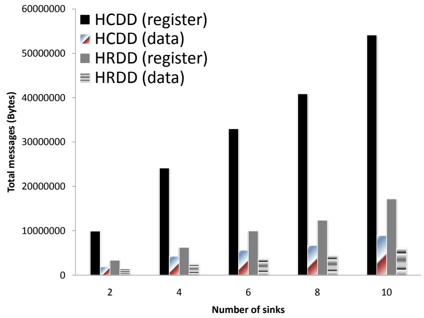


Figure 4.5: Overall message for different number of sinks

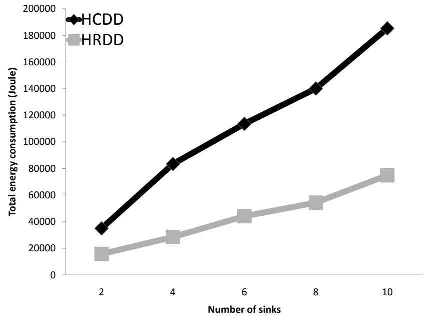


Figure 4.6: Energy consumption for different number of sinks

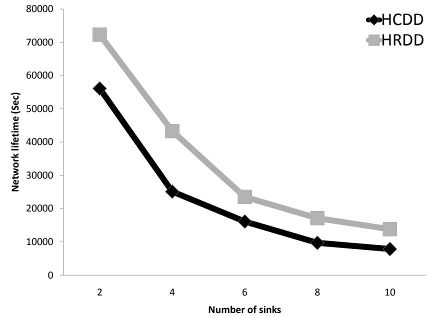


Figure 4.7: Network lifetime for different number of sinks

searching stage.

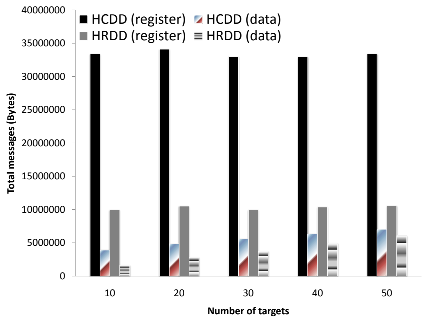


Figure 4.8: Overall message for different number of targets

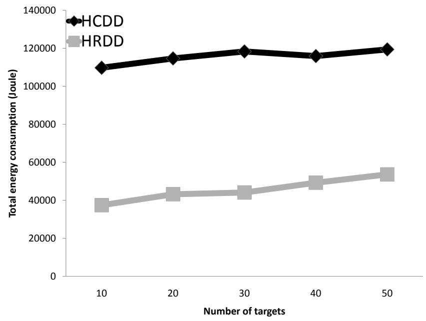


Figure 4.9: Energy consumption for different number of targets

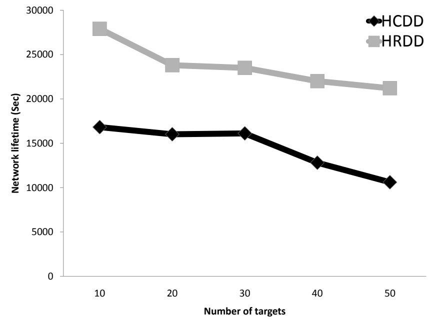


Figure 4.10: Network lifetime for different number of targets

4.4 Impact of *D* Hops between a Node and its CH

In this section we evaluate how hops between a node and its CH affect HRDD and HCDD. In this experiment, the number of hops is varied from 1, 2, 3, 4, to 5 and Figure 4.11 illustrates the number of high-level CHs with different number of hops by Max-Min *D*-cluster algorithm. When *D* hops are more and more large, the number of high-level CHs will be less and less little until there is only one high-level CH. Figure 4.14 shows the overall message at different numbers of *D* hops. In Sink Location Registration procedure, since decrease of high-level CHs, the average link of high-level CH become more and more until *D* hops equal four, as shown in Figure 4.12. The link of high-level CH is that the links between high-level CH and its neighboring high-level CHs. In HCDD, high-level CH broadcasts registration message to other high-level CHs through border clusters, so the generated of registration

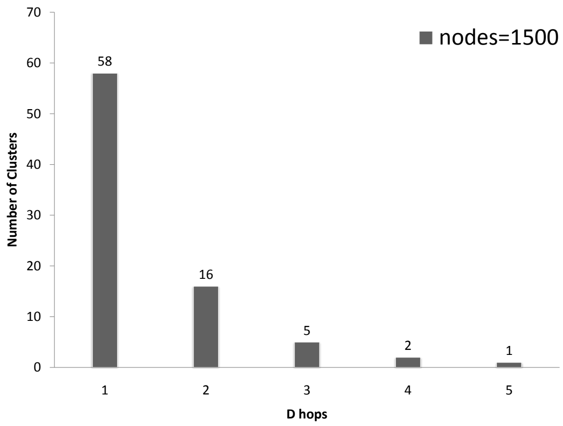


Figure 4.11: Number of high-level CHs for different number of *D* hops

packets was decided by the number of link between high-level clusters. On the contrary, HRDD sends register packet through Gateway Agent, and therefore the generated of registration packets was decided by the degree of high-level CH, as shown in Figure 4.13. The degree of high-level CH is the number of high-level CH's neighboring high-level CHs. With the increase of *D* hops, the size of high-level cluster

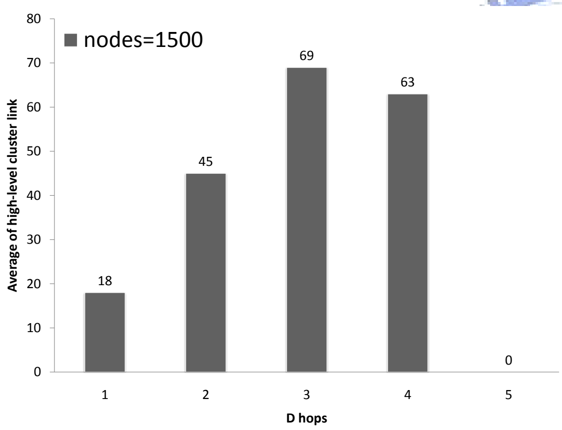


Figure 4.12: Number of high-level CH link for different number of d hops

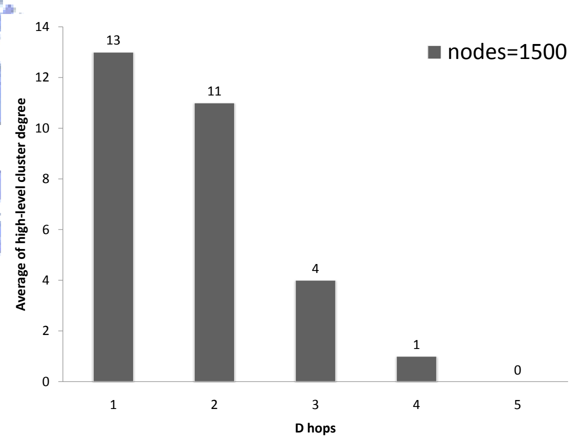


Figure 4.13: Number of high-level CH degree for different number of d hops

is more and more large and even the same as entire sensor network. For HRDD, though Gateway Agent could decrease number of flooding messages to other high-level CHs, sinks mobility still make registration packets increase in the large size of high-levle cluster. Once there is only one high-level CH in WSN, the number of registration packets in HCDD and HRDD is the same for sink registration. In Query Data Forwarding procedure, due to the increase of high-level cluster size, the number of data packets also increases little by little. At the beginning of the experiment, the data packets in HRDD are more than in HCDD caused by the number of Indexing Agents. For HRDD, when a source detects an event or a target, source's local CH informs its Indexing Agent with an event message except that source's local CH stores the sensing data. Hence, when *D* hops between a node and its CH are small, HRDD generates more data packets with many Indexing Agents than HCDD. Because of the increase of *D* hops, the number of Indexing Agents decreases, and therefore HRDD generates less data packets than HCDD. Until *D* hops equal to five, the number of data packets decreases in HRDD and HCDD since there is only one high-level CH in WSN.

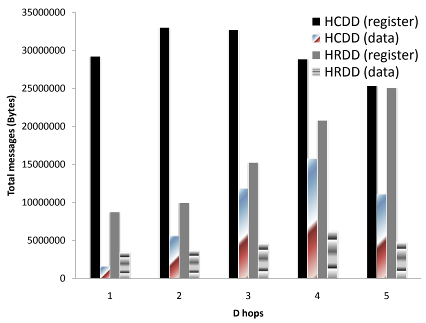


Figure 4.14: Overall message for different number of *D* hops

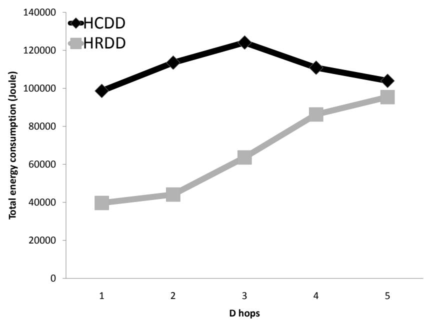


Figure 4.15: Energy consumption for different number of *D* hops

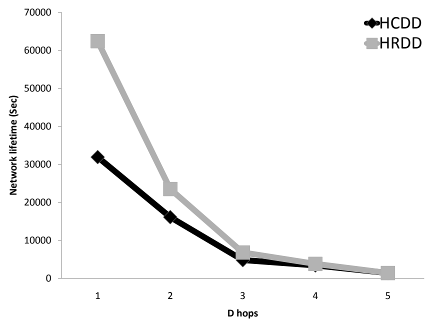


Figure 4.16: Network lifetime for different number of *D* hops

Figure 4.15 and Figure 4.16 illustrate the energy consumption and network lifetime with the different number of *D* hops. For energy consumption, the trend of curve relates to the number of register packets, so the curve of energy consumption is similar to Figure 4.14 with register packets of HRDD and HCDD respectively. The network lifetime with large size of high-level cluster causes heavier load in the high-level CHs, which are for sink registration and data forwarding, and therefore has shorter network lifetime. HRDD perform better than HCDD because of less overhead consumed update the sink information.

4.5 Comparison with Different Location of Indexing Agent

In this section we compare the performance of different location of Indexing Agent. In addition to original Indexing Agent, we also let high-level CH play Indexing Agent. In order to show that the Indexing Agent in the border nodes of high-level cluster is better than in high-level CH, we apply the same scenarios on both different location of Indexing Agent to study the impact of different numbers of targets. For simplifing the name of different location of Indexing Agent, we called the Indexing Agent in the border nodes of high-level cluster is HRDD-IA and in high-level CH is HRDD-CH. All simulations have 1500 sensor nodes and 6 sinks. The simulation results are shown in Figure 4.17 - Figure 4.19.

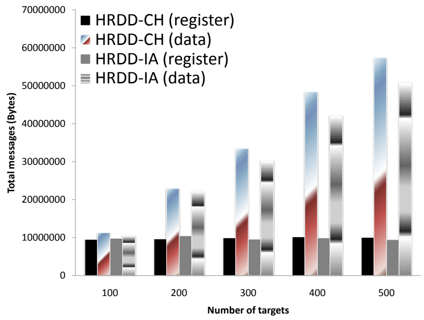


Figure 4.17: Overall message for different location of Indexing Agent

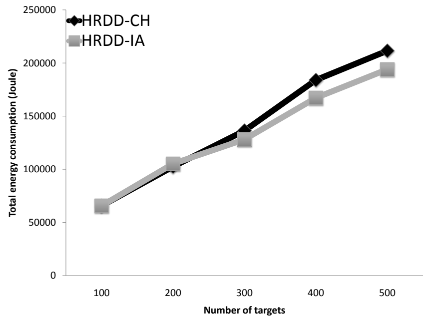


Figure 4.18: Energy consumption for different location of Indexing Agent

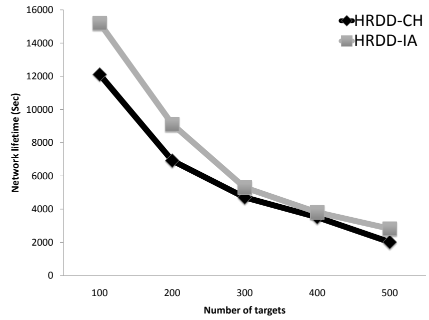


Figure 4.19: Network lifetime for different location of Indexing Agent

We first look at overall message at different numbers of targets, shown in Figure 4.17. For the same number of sinks, HRDD-CH and HRDD-IA have similar number of registration packets in Sink Location Registration procedure. In Query Data Forwarding procedure, more data packets are generated as the number of targets increases, but the data packets of HRDD-CH are more than HRDD-IA. In Event Detection procedure of HRDD-CH, because low-level CH has to send extra event message to inform high-level CH, this brings about more data packets in HRDD-CH. Figure 4.18 plot the energy consumption for HRDD-CH and HRDD-IA. The entire sensor network consumes much energy caused by the increase of data packets. When the number of targets is large, HRDD-IA consumes less energy than HRDD-CH. This is because that data packets of HRDD-IA are less than HRDD-CH. Figure 4.19 plot the network lifetim by HRDD-CH and HRDD-IA, respectively. Since low-level CH has to send extra event message to inform high-level CH, this may make high-level CH get heavier load in the high-level CH. Therefore, the network lifetime of HRDD-IA is longer than HRDD-CH. From above, we can obtain that the Indexing Agent in the border nodes of high-level cluster has the better performance than in high-level CH.

Chapter 5

CONCLUSIONS

In this paper we described HRDD for developing an energy-efficient data dissemination with multiple mobile sinks in WSNs. We assign two roles, called Indexing Agent and Gateway Agent in the WSNs. Indexing Agent could avoid unnecessarily transferring the event data, and Gateway Agent could decrease energy consumption of broadcasting and number of flooding messages. In addition, HRDD solves the data dissemination challenges in WSNs, such as using clustering techniques to build a hierarchical structure without location information, selecting high-level CHs for disseminating and saving sink's information, and constructing Indexing Agents as a rendezvous area for events and queries. Simulations results have shown that HRDD consequently conserves the battery energy of the sensor node efficiently while registering sinks' information to high-level CHs by Gateway Agent, and delivering data to mobile sinks by Indexing Agent.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, 2002.

[2] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. IEEE Journal of Wireless Communications, 2004.

[3] A. D. Amis, R. Prakash, T. H. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. Proceedings of the 19th IEEE INFOCOM Conference, 2000.

[4] D. J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications, 1981.

[5] S. Basagni. Distributed clustering algorithm for ad hoc networks. Proceedings of the 4th InternationalSymposium on Parallel Architectures, Algorithms and Networks, 1999.

[6] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2004.

[7] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. Proceedings of the Wireless Communications and Mobile Computing, 2002.

[8] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. ACM SIGCOMM Computer Communication Review, 2001.

[9] M. Chatterjee, S. K. Das, and D. Turgut. An on-demandweighted clustering algorithm (wca) for ad hoc networks. Proceedings of the IEEE Global Telecommunications Conference, 2000.

[10] M. do Val Machado, O. Goussevskaia, R. A. F. Mini, C. G. Rezende, A. A. F. Loureiro, G. R. Mateus, and J. M. S. Nogueira. Data dissemination in autonomic wireless sensor networks. IEEE Journal on Selected Areas in Communications, 2005.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, 2002.

[12] C. Intanagonwiwat, R. Govindan, D. Estrin, and J. Heidemann. Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on Networking, 2003.

[13] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, 2000.

[14] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, 2003.

[15] J. Kulik, W. Rabiner, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999.

[16] C.-J. Lin, P.-L. Chou, and C.-F. Chou. Hcdd: Hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink. Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing, 2006.

[17] J. A. Miller, R. S. Nair, Z. Zhang, and H. Zhao. Jsim: A java-based simulation and animation environment. Proceedings of the 30th Annual Simulation Symposium, 1997.

[18] M. Qin and R. Zimmermann. An energy-efficient voting-based clustering algorithm for sensor networks. Proceedings of the 6th International Conference on Software Engineering, 2005.

[19] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. ACM SIGCOMM Computer Communication Review, 2003.

[20] J.-H. Shin, J. Kim, K. Park, and D. Park. Railroad: Virtual infrastructure for data dissemination in wireless sensor networks. Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, 2005.

[21] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culle. An analysis of a large scale habitat monitoring application. Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004.

[22] A. Visvanathan, J.-H. Youn, and J. Deogun. Hierarchical data dissemination scheme for large-scale sensor networks. Proceedings of the IEEE International Conference on Communications, 2005.

[23] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, 2005.

[24] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005.

[25] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, 2001.

[26] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, 2002.

[27] O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing, 2004.

[28] W. Zhang, G. Cao, and T. L. Porta. Data dissemination with ring-based index for wireless sensor networks. IEEE Transactions on Mobile Computing, 2007.