# 國 立 交 通 大 學

## 資訊科學與工程研究所

## 碩 士 論 文

無線感測網路下

之高效能叢集式資料散播技術

An Efficient Cluster-based Data Dissemination Scheme

in Wireless Sensor Networks

研 究 生：劉人仲

指導教授：王國禎 博士

中 華 民 國 九 十 九 年 六 月

# 無線感測網路下
# 之高效能叢集式資料散播技術

# An Efficient Cluster-based Data Dissemination Scheme in Wireless Sensor Networks

研 究 生：劉人仲　　Student：Ren-Jhong Liu

指導教授：王國禎　　Advisor：Kuochen Wang

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institutes of Computer Science and Engineering

Department of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in Computer Sciencmae

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

# 無線感測網路下
# 之高效能叢集式資料散播技術

學生：劉人仲　　　指導教授：王國禎 博士

國立交通大學資訊科學與工程研究所

## 摘 要

一般而言，無線感測網路受限於有限的能力和非充電式電池，所以如何降低各個節點的能源消耗是一個重要的研究議題。現存的資料散播協定採用泛播機制來傳遞興趣與尋找傳送路徑會造成大量的能源消耗。為了減輕這個問題，叢集式資料散播可以被使用來降低能源消耗與延長網路壽命。能源階層式被動叢集(ELPC)技術是基於叢集式資料散播來平衡感測節點的能源消耗與增加網路壽命。但是ELPC仍然採用泛播探索性資料訊息來選擇一條傳送路徑，這會造成額外的能源消耗和延遲。因此我們提出一個高效能叢集式資料散播(ECDD)技術來解決這個問題。在我們的方法中，除了附

加控制訊息到興趣中來執行隨選式被動叢集，為了幫助感測節點去選擇下一個到匯集點最少跳躍數的傳輸節點，我們也利用額外的控制訊息來對每一個感測節點設定一個跳躍數。利用這個方法，可以找到一條最短路徑來傳輸感測資料給匯集點。對資料散播而言，避免使用探索性資料訊息可以有降低感測節點的能源消耗與達到低延遲時間的優點。實驗結果顯示，我們的方法在平均耗損能量上可以改善直接擴散(DD)協定達到61.30%，另外也可改善ELPC達到22.33%。此外，我們的方法在平均延遲時間上相對於DD與ELPC，分別可降低57.45%和23.49%的延遲時間。ECDD可運作於需長期性監控與即時性回報的應用，如社區看護系統和災區救援方面。

**關鍵詞**：叢集、資料散播、附加控制訊息、無線感測網路。

# An Efficient Cluster-based Data Dissemination Scheme in Wireless Sensor Networks

**Student：Ren-Jhong Liu　　Advisor：Dr. Kuochen Wang**

Department of Computer Science
National Chiao Tung University

## Abstract

Generally, the wireless sensor network (WSN) is restricted with limited capability and un-rechargeable battery power. Therefore, how to reduce energy consumption of sensor nodes becomes an important research issue. Existing data dissemination protocols adopted flooding to propagate interests and find forwarding paths in WSNs, which cause large energy consumption. To relieve this problem, cluster-based data dissemination was adopted to reduce energy consumption and prolong network lifetime. The energy level-based passive clustering (ELPC) scheme is based on cluster-based data dissemination to balance energy consumption of nodes and prolong network lifetime. However, ELPC still has to use exploratory data messages to select a forwarding path by flooding, which causes extra energy consumption and delay. Therefore, we propose an *efficient cluster-based data dissemination* (ECDD) scheme to resolve this problem. In the proposed ECDD, besides piggyback control information into interests to perform on-demand passive clustering, we also use control information to set each node a hop count for assisting a node to select next forwarding node with the least hop count to the sink. In this way, a shortest path to forward sensed data back to the sink can be found. The advantages of avoiding exploratory data messages for data dissemination are reducing energy consumption of sensor nodes and achieving low delay. Simulation results show that

ECDD is 61.30% better than DD, a classical approach, and 22.33% better than ELPC in terms of average dissipated energy. Furthermore, our approach is 57.45% and 23.49% better than DD and ELPC, respectively, in terms of average delay. The proposed ECDD is feasible for applications of long-term monitoring and real-time responding, such as a community health care system and rescue in a disaster area.

**Keywords**: Cluster-based, data dissemination, piggybacked control information, wireless sensor network.

# Acknowledgements

Many people have helped me with this thesis. I am in debt of gratitude to my thesis advisor, Dr. Kuochen Wang, for his intensive advice and guidance. I would also like to show my appreciation for all the classmates in the *Mobile Computing and Broadband Networking Laboratory* for their invaluable assistance and inspirations. The support by the National Science Council under Grants NSC 97-3114-E-009-001 and NSC 98-2218-E-009-010 is also gratefully acknowledged. Finally, I thank my father, my mother and my friends for their endless love and support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Data dissemination in wireless sensor networks

A wireless sensor network (WSN) consists of a large number of small sensor nodes with low cost. A typical wireless sensor network model is depicted in Figure 1. Typically, sensor nodes are deployed throughout an area to sense events that we are interested in. The sink will send request messages to sensor nodes if the sink is interested in some types of events. The request messages which the sink sends to the sensor nodes are called *interests*. When sensor nodes sense event data that match the interests, the sensed data will be forwarded back to the sink through multi-hop paths. Because sensor nodes have the characteristics of low cost and well adaptability to environments, the WSN technology can be applied to various domains,

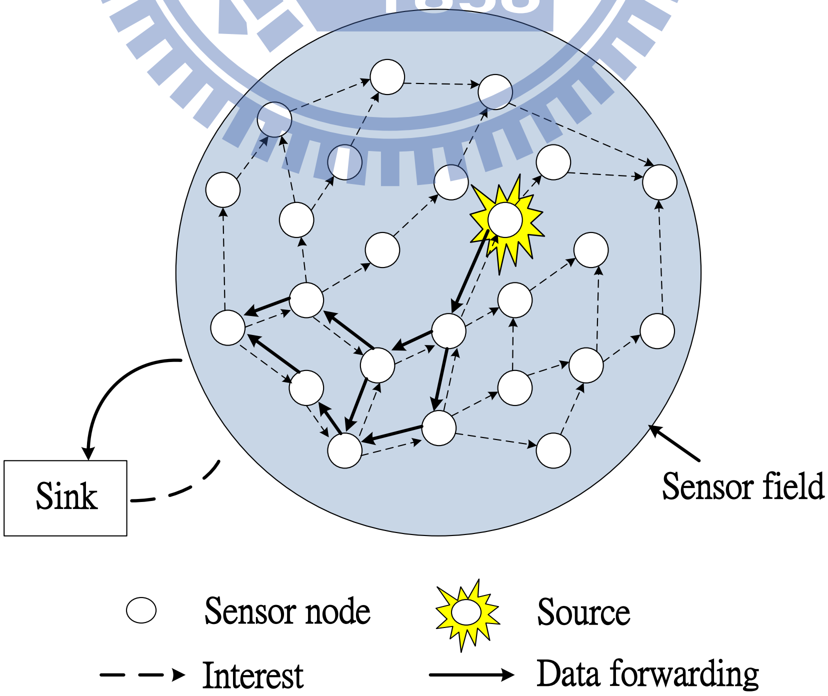



**Figure 1. Typical wireless sensor network model.**

such as wildlife habitats, disaster management, emergency response, ubiquitous computing environments, asset tracking, healthcare, and manufacturing process flow [2]. Nevertheless, the capabilities of a sensor node, such as battery power, computing, signal range and storage space, are limited. Thus, efficient resource management of a sensor node is essential, especially for energy management. The energy consumption of sensor nodes is greatly affected by the forwarding path selection mechanisms in data dissemination protocols. Therefore, designing an energy-efficient forwarding path selection mechanism becomes an important research issue in WSNs [1].

## 1.2 Motivation

Traditional data dissemination protocols use flooding to send interests and forward sensed data. One main disadvantage of flooding is high energy consumption. Directed Diffusion (DD) reduces energy consumption by selecting empirically good paths and by caching and processing data in-network [3, 4]. Although DD reduces energy consumption, it still periodically diffuses interests and sends *exploratory data messages* to find a forwarding path with low data dissemination delay. The exploratory data messages are used for data dissemination path setup and repair. Existing DD-based data dissemination protocols [5, 7, 8, 10, 11] use clustering techniques to reduce the amount of interests and exploratory data messages. According to the type of control messages dissemination, clustering mechanisms can be divided into *active clustering* and *passive clustering*. The active clustering [8] periodically sends control messages to compose and maintain clusters, while the passive clustering piggybacks control information into packets to build and maintain clusters in order to reduce control message overhead [8]. Although the clustering mechanism can resolve the interests flooding problem, sending exploratory data messages still causes large energy consumption.

## 1.3 Problem statement

As mentioned before, the WSN technology can be applied to various domains in our daily life. However, the behavior of data transmission, the characteristics of the wireless environment, even hardware capabilities of sensor nodes will affect construction and maintenance of data dissemination paths. A data dissemination path may change if some of the sensor nodes become invalid due to energy depletion. Therefore, designing an energy-efficient data dissemination protocol is a great challenge. In order to reduce the energy consumption of sensor nodes, finding an efficient forwarding path is important. Several existing approaches flood exploratory data messages to select a better path; however, flooding exploratory data messages causes significant energy waste and data dissemination delay. In order to save more energy and achieve lower data dissemination delay, we piggyback control information in the interest to set each node a hop count for assisting a node to select next forwarding node with the least hop count to the sink. In this way, a shortest path to forward sensed data back to the sink can be found.

## 1.4 Thesis organization

The remainder of this thesis is organized as follows. We review related work in Chapter 2. We describe the proposed Efficient Cluster-based Data Dissemination (ECDD) scheme in Chapter 3. Simulation results are presented in Chapter 4. We evaluate the proposed ECDD and compare it with DD and ELPC. Chapter 5 concludes the thesis and outline future work.

# Chapter 2

# Related Work

## 2.1 Directed diffusion protocol

Directed Diffusion (DD) [3, 4] is a classical data-centric data dissemination scheme in WSNs. The sink repeatedly diffuses interests to inform each sensor node what kinds of events the sink wants. When a sensor node senses events which match the interests and the forwarding path is not decided, the sensor node sends exploratory data messages along all possible forwarding paths toward the sink. After the sink received the exploratory data messages, the sink chooses a path with the lowest delay, which is called a reinforced forwarding path, according to the received exploratory data messages. The sink notifies the sensor node with this path using a reinforcement message, and the sensor node uses this reinforced forwarding path to forward sensed event data back to the sink instead of flooding. In DD, there are two flooding probable problems. First, the sink periodically floods interests in order to notify sensor nodes to send sensed event data that match the interests [5]. Second, in order to find and maintain reinforced forwarding paths, sensor nodes repeatedly send exploratory data messages by flooding [13]. Because of these two flooding problems, the energy of sensor nodes will be exhausted rapidly.

## 2.2 Improved directed diffusion protocol

Improved Directed Diffusion (IDD) [5] is a cluster-based directed diffusion protocol in WSNs. Before the sink propagates interests into sensor fields, the sink sends a cluster formation message into sensor fields to organize clusters. After the cluster formation finished, all nodes can be categorized into three kinds of nodes: cluster head, cluster member and border node [5]. If a cluster head receives an interest, the cluster head broadcasts the received interest to its cluster members and border nodes. When a cluster member senses events which match the interests, the cluster member forwards the sensed data via its cluster head to the sink. This approach restricts interests to be diffused among cluster heads and border nodes, which suppresses the flooding phenomenon to reduce energy consumption and delay [5]. The disadvantage of IDD is that it utilizes the active clustering mechanism which sends additional control messages to perform cluster formation, floods exploratory data messages, and therefore increases overheads.

## 2.3 Directed diffusion based on clustering protocol

Directed Diffusion Based on Clustering (DDBC) [7] adopts passive clustering to achieve cluster formation, which reduces extra overheads by piggybacking additional control information into the interests. After a sensor node receives an interest, the sensor node will be assigned to one of the three node statuses, which were defined in [7], according to the content of the interest. After the sensor node change its node status, the content of the interest will be updated and then sends to the sensor node’s neighbors. After cluster formation, a cluster member only communicates with its cluster head. All the sensed data that match the interests will be sent to its cluster head. The cluster head then sends the sensed data back to the sink. This approach not only reduces delay of cluster formation but also reduces additional control messages to improve energy consumption [7]. Although DDBC utilizes the passive clustering mechanism to solve the interest flooding problem, it still has the exploratory data message flooding problem while finding a data forwarding path for a sensor node to send back its sensed data back to the sink. Flooding exploratory data messages increase the energy consumption of cluster heads and boarder nodes, as well as the delay of forwarding paths selection.

## 2.4 Energy level-based passive clustering protocol

Energy Level-based Passive Clustering (ELPC) [10] modifies a passive clustering mechanism for building and maintaining the cluster formation. ELPC also combines DD so that the consumed energy balancing of the sensor nodes can be achieved [10]. It adds additional two parameters, the node energy level and the network energy level, in order to achieve balanced energy consumption in each node. The node energy level represents the total energy consumption of a sensor node and the network energy level represents an energy threshold. Cluster heads or border nodes change their node statuses when their node energy levels are higher than the predefined network energy level [10]. Each sensor node has chances to be a cluster head or a border node so that the energy consumption of each node will be balanced and the network lifetime will be prolonged. ELPC may cause the detour problem [14] which causes more energy consumption and delay because a newly selected cluster head or border node may increase the length of a data forwarding path. On the other hand, this protocol also consumes more energy consumption and has more delay due to flooding exploratory data messages for selecting a data forwarding path.

## 2.5 Qualitative comparison of existing data dissemination protocols

The qualitative comparisons of the protocols mentioned above and the proposed Efficient Cluster-based Data Dissemination (ECDD) protocol shown in Table 1. First, the *passive clustering* indicates that if a protocol uses passive clustering, it can reduce control message overhead for building and maintaining the clusters. IDD needs to send additional control messages to perform cluster formation. The *interests flooding problem* indicates that each node of the sensor field rebroadcasts interests when it receives interests, which will happen in DD while it propagates interests. The *exploratory data messages flooding problem*

[13] indicates a source sends exploratory data messages to sensor nodes in order to find a low delay path between the source and the sink. The proposed ECDD piggybacks hop count information in the interest for finding a shortest forwarding path instead of flooding exploratory data messages.

**Table 1. Comparison of existing data dissemination protocols, including the proposed ECDD.**

| Approach | Passive clustering | Interests flooding problem | Exploratory data messages flooding problem |
|---|---|---|---|
| **DD [3, 4]** | No<br>(no clustering) | Yes | Yes |
| **IDD [5]** | No | No | Yes |
| **DDBC [7]** | Yes | No | Yes |
| **ELPC [10]** | Yes | No | Yes |
| **ECDD (proposed)** | Yes | No | No |

# Chapter 3

# Proposed Efficient Cluster-based Data Dissemination Scheme

In this chapter, we propose an Efficient Cluster-based Data Dissemination (ECDD) scheme and describe details of our proposed approach. The proposed ECDD is based on a passive clustering mechanism to build and maintain clusters and uses a *first declaration wins* [8, 9] mechanism to select cluster heads. The passive clustering mechanism not only reduces extra overheads caused by the cluster-based mechanism but also reduces significant energy consumption and delay caused by flooding interests and exploratory data messages. With the *first declaration wins* mechanism, a node which first claims to be the cluster head will become the cluster head of its clustered area (in the radio coverage) and manages nodes in this clustered area [8, 9]. The proposed ECDD does not use waiting periods (to make sure all the neighbors have been checked) [8], which is used in all other weight-driven clustering mechanisms, to select the best node to be the cluster head [8]. Once the cluster formation is finished, all nodes can be categorized into three types: *cluster head*, *cluster member*, and *border node*. A cluster head has the responsibility to manage cluster members and border nodes in its clustered area. A border node is a node which is in the communication range of two or more different clusters. In order to avoid increasing overheads of cluster heads, we assume that cluster heads do not sense events and only forward sensed data which are sent by cluster members to the sink. When a cluster head receives an interest, the cluster head sends the interest to its cluster members and border nodes. A cluster head will send the sensed data to border nodes in order to forward the sensed data via another cluster heads back to the sink.

The main tasks of a cluster member are receiving interests from its cluster head and forwarding sensed data that match the received interests to its cluster head.

Besides piggybacking control information [11] into the interests, ECDD also piggyback additional fields, *HOP_COUNT* and *ENERGY_THRESHOLD*, into the interests in order to set the hop count and energy threshold to each node for data dissemination path selection. The definition of HOP_COUNT and ENERGY_THRESHOLD will be described in Section 3.1. Using hop count information, we can also find a shortest data forwarding path which can save more energy and achieve lower data dissemination delay. In order to guarantee that data dissemination is reliable, we use a route maintenance mechanism which will be described in Section 3.3. We make the following assumptions in the proposed ECDD:

- The sink has unlimited memory, processing capability and rechargeable battery. That is, the energy of the sink can be recharged anytime [14].
- All sensor nodes besides the sink have the same processing capability.
- Initially, the hop count of the sink is set to 0 and the hop count of each node other than the sink is set to the total number of sensor nodes in the sensor field.
- The cluster formation is complete before any data gathering and transmission begins [5]

## 3.1 Piggybacked control information in an interest

The piggybacked control information in the interests is used to set the hop count of each node for selecting an efficient data dissemination path and an energy threshold of each node for reconstructing the clusters. In the proposed ECDD, we extend the piggybacked control information proposed by [11] by adding two more fields: HOP_COUNT and ENERGY_THRESHOLD, as shown in Figure 2. The definitions of the fields contained in the piggybacked control information in an interest are shown in Table 2. The piggybacked control information will add additional 21 bytes to the original interest defined in the DD protocol. Although the modified interest increases 21 bytes compared with the original interest, the construction and maintenance of clusters does not need to send additional control messages due to the piggybacked control information. Furthermore, exploratory data messages can be eliminated because we use hop count information to select a data dissemination path. The cost of using the modified interest is lower than the cost of using additional control messages for clustering and exploratory data messages for finding a low delay path.

| 0 | 31 | 39 | 71 | 103 | 135 | 167 |
|---|---|---|---|---|---|---|
| NODE_ID | STATUS | CH1_ID | CH2_ID | HOP_COUNT | ENERGY_THRESHOLD | |

**Figure 2. Piggybacked control information in an interest.**

**Table 2. The definition of fields in piggybacked control information.**

| **Field** | **Definition** |
|---|---|
| NODE_ID | Identification of a sensor node |
| STATUS | Type of a sensor node: cluster head, cluster member, or border node |
| CH1_ID & CH2_ID | If a sensor node is selected as a border node, the border node notifies the NODE_IDs of the two cluster heads to which it connects |
| HOP_COUNT | Set or update the hop count of a sensor node; the hop count represents the total hops from the sensor node to the sink |
| ENERGY_THRESHOLD | An energy threshold of a cluster head or a border node for the route maintenance mechanism |

Once a sensor node receives an interest, it updates a *neighbor information table* according to the piggybacked control information in the interest. Figure 3 shows the structure of the neighbor information table. The neighbor information table contains the following information: *NEIGHBOR_ID*, *NEIGHBOR_STATUS*, and *NEIGHBOR_HOP_COUNT*. NEIGHBOR_ID is used to identify which neighbor is referred by this information record.

NEIGHBOR_STATUS keeps track of which type of this neighbor is. NEIGHBOR_HOP_COUNT represents the total hop counts from this neighbor to the sink. According to the neighbor information table, a sensor node selects the next forwarding node with the least hop count, and thus a shortest data dissemination path to forward sensed data back to the sink can be found.

0 31 39 71

| | | |
|---|---|---|
| NEIGHBOR_ID | NEIGHBOR_STATUS | NEIGHBOR_HOP_COUNT |

**Figure 3. Structure of a neighbor information table.**

## 3.2 Cluster formation and data dissemination path selection

If a cluster-based sensor network has not been initialized, the cluster formation mechanism will be performed. In the proposed ECDD algorithm, we design a novel cluster formation mechanism which utilizes hop count information in each sensor node. After the cluster formation mechanism is finished, the data dissemination path selection algorithm can make use of the updated hop count information in each sensor node to select a data dissemination path from source to sink with low delay.

### 3.2.1 Cluster formation

In this phase, we set the hop count of each node by using a hop count update procedure, as illustrated in Figure 4. Initially, the hop count of each node is set to the total number of sensor nodes in the sensor field. When an interest is propagated into the sensor field for the first time, the cluster formation mechanism will be performed, and the interest will be propagated to all sensor nodes in the sensor field. Before the interest is propagated into the sensor field, the HOP_COUNT field in a piggybacked interest will be set to the value of the hop count stored in the sink plus one. When a sensor node receives the interest, the sensor node will compare its hop count with the value stored in the HOP_COUNT field of the received interest. If the hop count value stored in the sensor node is larger than the value of the HOP_COUNT field in the received interest, the sensor node updates its hop count by the value stored in the HOP_COUNT field of the received interest. Otherwise, the interest will be discarded. Then, the sensor node checks whether there are more interests coming in or not. If the sensor node receives other interests, the sensor node must compare the hop count value with the value of the HOP_COUNT field in the recently received interests again to decide whether the hop count value needs to be updated or not. If the sensor node does not receive any more interests, it replaces the value of the HOP_COUNT field in the received interest with the value of the sensor node's hop count plus 1. Finally, the sensor node sends the updated interest to other sensor nodes. When an interest is diffused throughout the entire sensor field, each node will be set a new hop count.

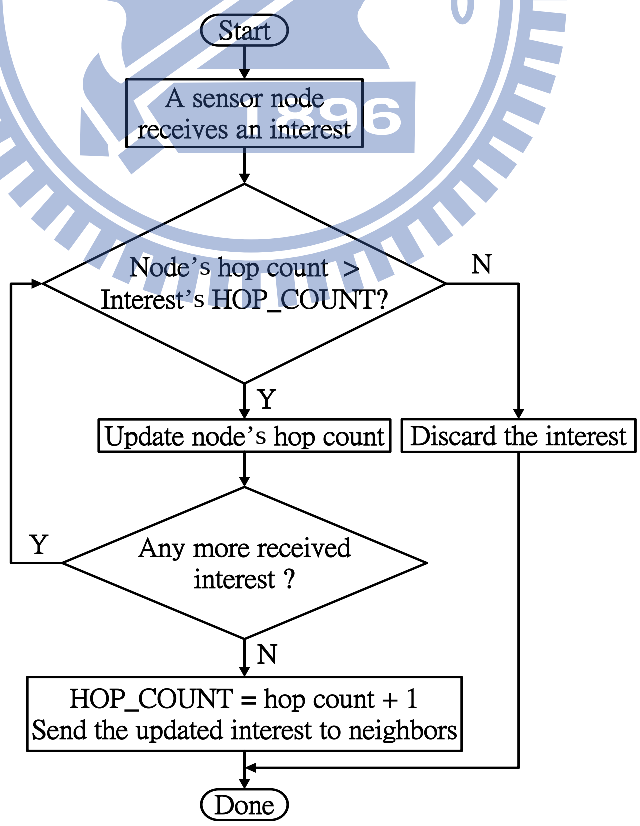



**Figure 4. Hop count update procedure for a sensor node.**

Figure 5 is an example schematic chart of the hop count update procedure shown in Figure 4. In Figure 5(a), each node is assigned an ID and the initial hop count in each sensor node is set to the value of total number of sensor nodes in the sensor field. After initialization, the interest with HOP_COUNT set by 1 is sent to the sensor field. Figure 5(b) shows that nodes, A, B, and C, that receives an interest and compares its hop count with the HOP_COUNT of the received interest. Nodes A, B, and C update their hop counts by the HOP_COUNT of the received interest because the HOP_COUNT of the received interest is smaller than their hop counts. Subsequently, nodes A, B, and C replace the HOP_COUNT of the received interests by the value of their hop counts plus 1. Finally, nodes A, B, and C send the updated interests to neighbor sensor nodes. In Figure 5(c), nodes D, E, F, and G receive the updated interests which are sent by nodes A, B, and C. Nodes D, E, F, and G also update their hop counts by the HOP_COUNT value in the updated interests according to the hop count update procedure. Each node will set a new hop count when an interest is diffused throughout the entire sensor field, which is shown in Figure 5(d).

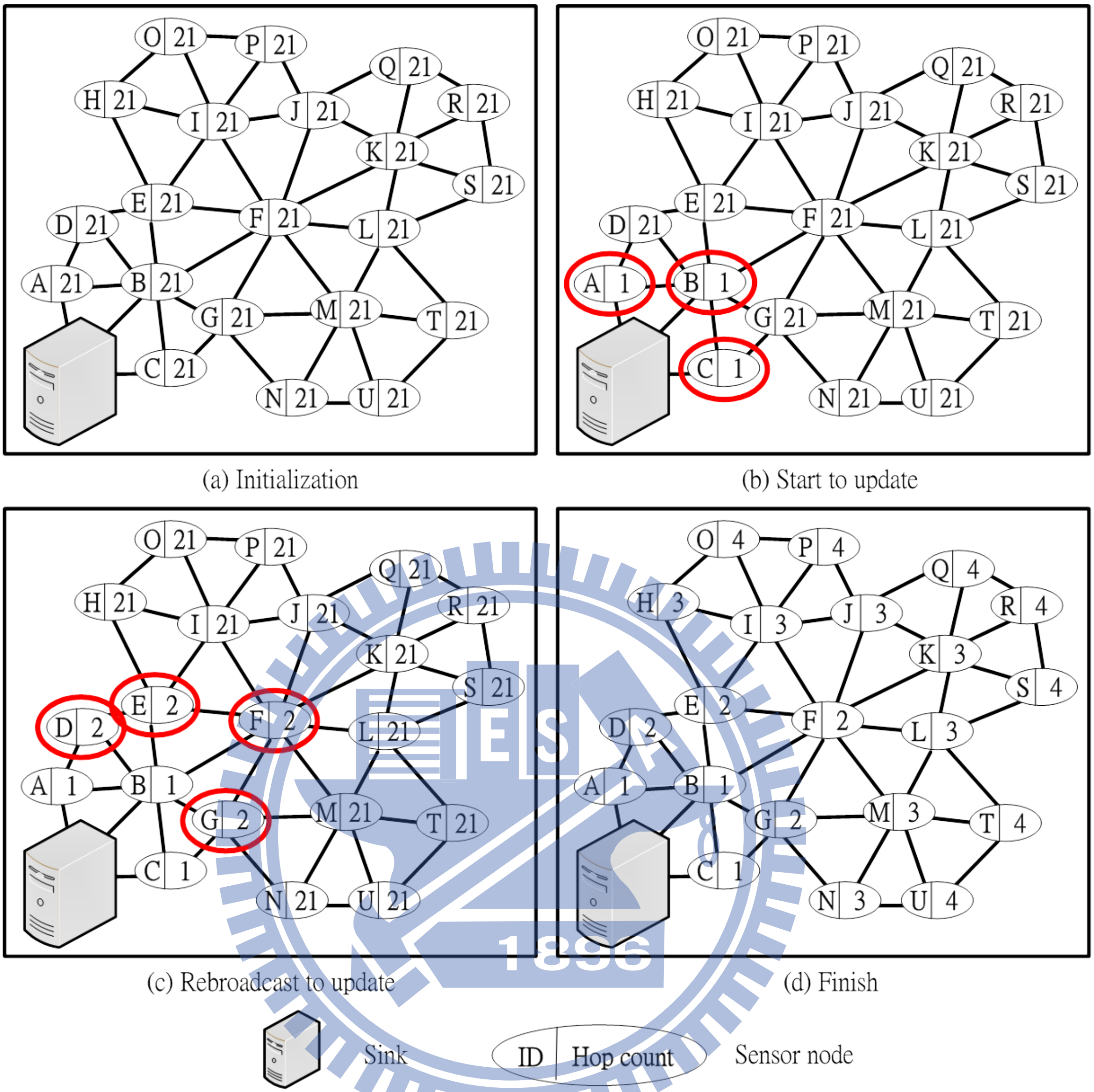
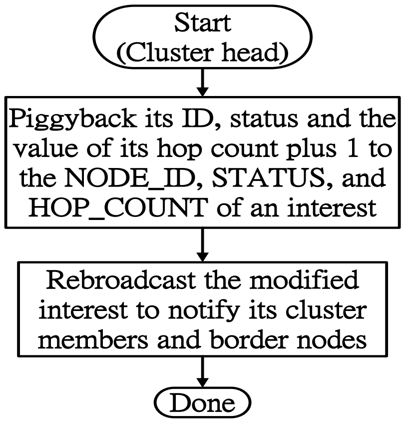
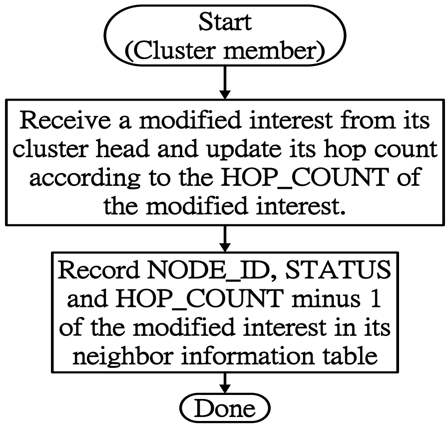



**Figure 5. An example schematic chart of the hop count update procedure (a) initialization, (b) start to update, (c) rebroadcast to update, (d) finish.**

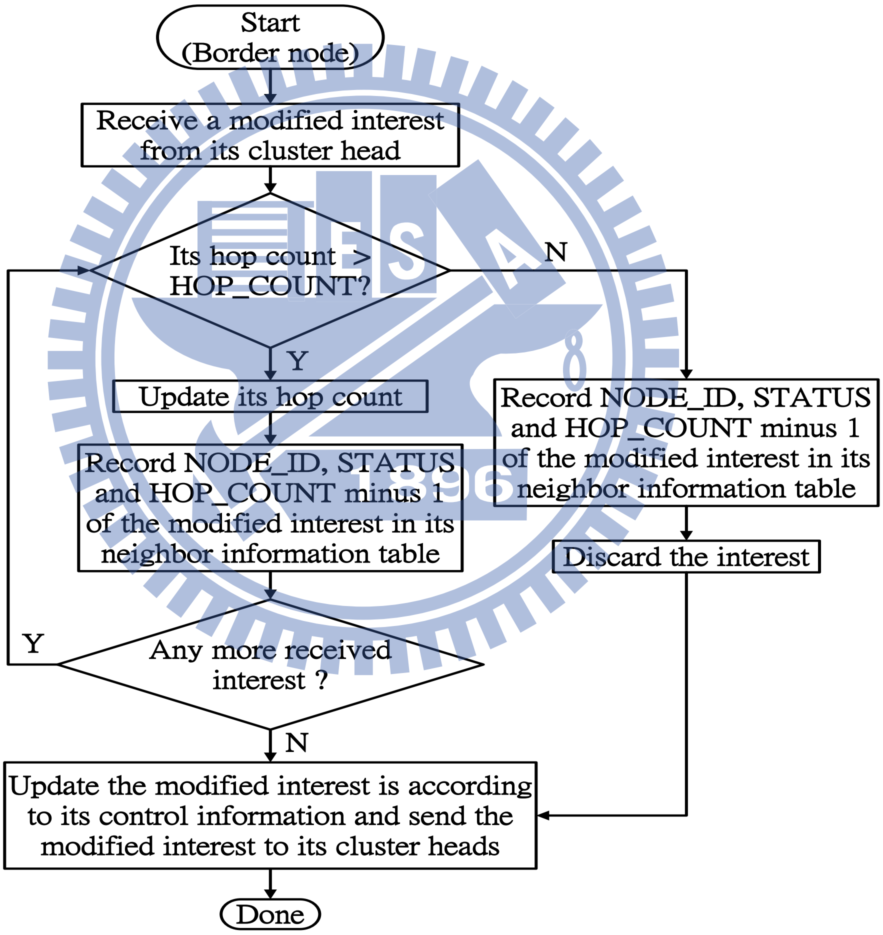
Figure 6 illustrates flowcharts of the actions taken by cluster heads, cluster members, and border nodes. Figure 6(a) shows the actions taken by a cluster head. After the cluster formation is finished, a cluster head will piggyback its ID, status, and the hop count value plus 1 into the NODE_ID, STATUS, and HOP_COUNT fields in the received interest. After finishing the update of the received interest, the cluster head then rebroadcasts the updated interest to its cluster members and border nodes. Figure 6(b) illustrates the actions taken by a cluster member. If a cluster member receives a modified interest, the cluster member will replace its hop count with the HOP_COUNT in the modified interest. The cluster member also stores NODE_ID, STATUS and HOP_COUNT minus 1 in the modified interest to the neighbor information table. Figure 6(c) depicts the actions taken by a border node. If a border node receives a modified interest, the border node will compare its hop count with the HOP_COUNT of the modified interest. If the HOP_COUNT in the modified interest is larger than the border node's hop count, the border node stores NODE_ID, STATUS, and HOP_COUNT minus 1 in the modified interest to the corresponding fields in the neighbor information table, then discards the interest. Otherwise, the border node will replace its hop count with the HOP_COUNT of the modified interest. Besides, the border node also stores NODE_ID, STATUS, and HOP_COUNT minus 1 in the modified interest to the corresponding fields in the neighbor information table. Afterwards, the border node checks whether there are more interests coming in. If the border node receives more interests, the hop count must be checked again. No matter the border node updates its hop count or not, it updates the received interest according to its NODE_ID, STATUS, CH1_ID, CH2_ID, and HOP_COUNT, and sends the updated interest to each cluster head that sends the modified interest to the border node. The border node uses this interest to notify the cluster heads that this border node belongs to them.




(a) Cluster head




(b) Cluster member




(c) Border node

**Figure 6. Flowcharts of the actions taken by (a) cluster heads, (b) cluster members, and (c) border nodes after the cluster formation is finished.**

Figure 7 shows an example of actions taken cluster heads, cluster members, and border nodes. After the cluster formation is finished, node B is the first cluster head formed. Node B piggybacks its ID, status, and its hop count plus 1 into the NODE_ID, STATUS, and HOP_COUNT of an interest. The interest then will be rebroadcast to its cluster member nodes A, C, D, E, and G. Nodes A, C, D, E, and G update their hop counts according to the HOP_COUNT in the modified interest and record the control information of node B in their neighbor information tables. When nodes I, K, and M become cluster heads subsequently, nodes I, K, and M also send the modified interests to their cluster members. Because nodes E, F, G, J, and L receive more than two modified interests, nodes E, F, G, J, and L will become border nodes. Node E, F, G, J, and L compare their hop counts with the HOP_COUNT of the modified interests which nodes I, K, and M sent, and update their hop counts. Nodes E, F, G, J, and L record the control information of nodes E, F, G, J, and L in their neighbor information tables. Finally, nodes E, F, G, J, and L update the modified interests according to their control information and send the modified interests to their cluster heads.

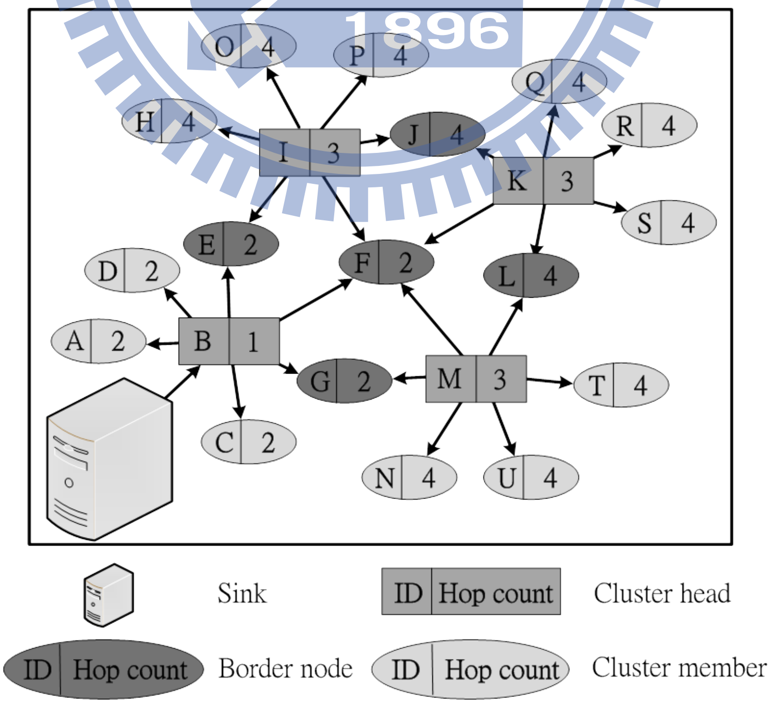



**Figure 7. An example of actions taken by cluster heads, cluster members, and border nodes.**

### 3.2.2 Data dissemination path selection

In this phase, we want to find a shortest data dissemination path to forward sensed data back to the sink after the cluster formation is finished. Figure 8 illustrates the data dissemination path selection process. If a sensor node senses an event that match an interest, the sensor node will become a source and periodically forwards the sensed data to its cluster head. After the cluster head receives sensed data, the cluster head checks whether the next hop is the sink or not. If the next hop is the sink, the sensed data will be forwarded back to the sink directly. Otherwise, the cluster head looks up its neighbor information table to select a border node with the least hop count. The cluster head then forwards the sensed data to the selected border node. After the selected border node receives the sensed data, it also looks up its neighbor information table to find a cluster head with the least hop count, which connects to the selected border node. The border node then forwards the sensed data to the selected cluster head. The sensed data will be forwarded hop by hop between cluster heads and border nodes until the sensed data arrive at the sink.

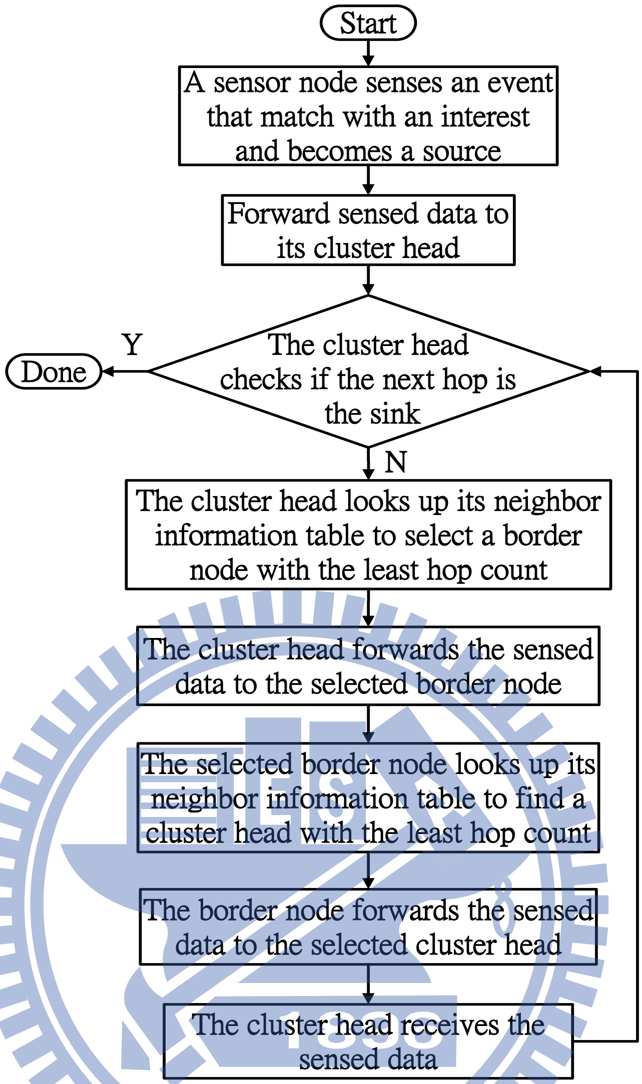



**Figure 8. The data dissemination path selection process.**

Figure 9 shows an example of how the sensed data being transferred toward the sink. Because node R senses an event that match an interest, node R becomes a source and forwards sensed data to it cluster head, which is node K. When node K receives the sensed data, node K checks whether the next hop is the sink or not. If the next hop is not the sink, node K looks up its neighbor information table to select a border node with the least hop count among nodes J, F, and L, which is node F. Node K then forwards the sensed data to node F. When node F receives the sensed data, node F also looks up its neighbor information table to select a cluster head with the least hop count among nodes B, I, M, and K, which is node B. Node F then forward the sensed data to node B. When node B receives the sensed data, node B forwards the sensed data to the sink because the next hop of node B is the sink.

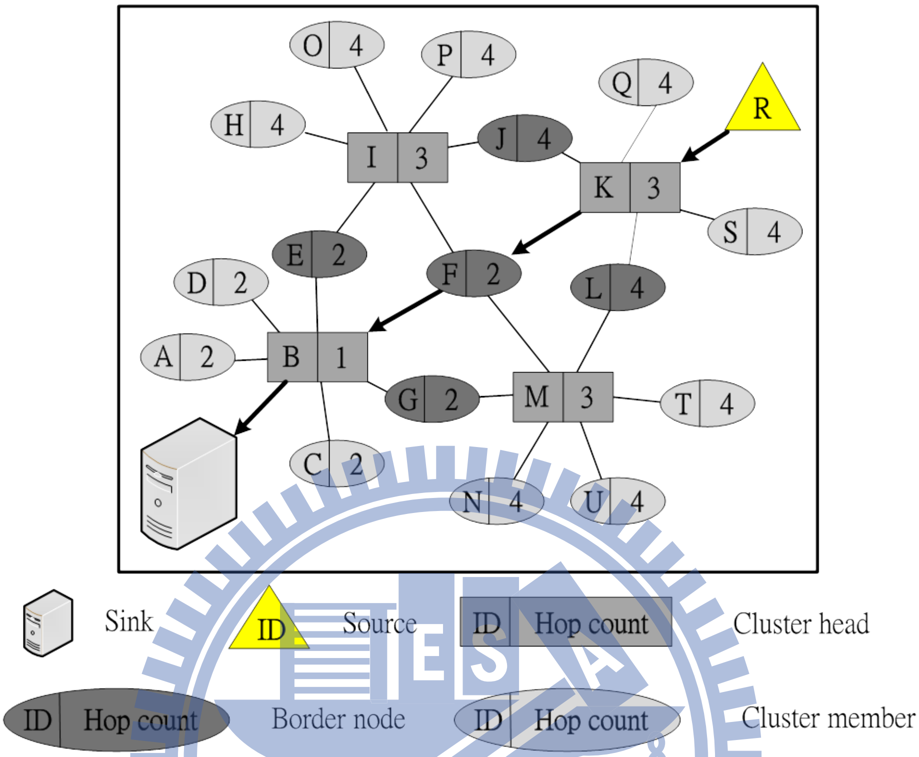



**Figure 9. An example of how sensed data being transferred toward the sink.**

Because the selection of the next forwarding node is based on the least hop count, the selected data dissemination path will be the shortest path. Using the data dissemination path selection process, sensed data can be forwarded back to the sink along a shortest data dissemination path. The hop count information not only helps construct a data forwarding path with low delay but also helps reduce energy consumption and delay for data dissemination.

## 3.3 Route maintenance mechanism

In the proposed ECDD, besides finding an efficient data dissemination path to reduce the energy consumption of sensor nodes, we also need to consider a situation that an data dissemination path may change if some of the sensor nodes become invalid due to energy depletion. Because our approach uses a cluster-based scheme for data dissemination, cluster heads and border nodes will cons7ume more energy than cluster members. In order to deal with the invalidation of a data dissemination path caused by energy depletion, we piggyback ENERGY_THRESHOLD, which is the energy threshold of a sensor node, into an interest. The value of ENERGY_THRESHOLD is controlled by the sink. When the sink wants to send an interest to a sensor field, the sink sets an energy threshold value to the ENERGY_THRESHOLD field of an interest. Therefore, when the residual energy of a sensor node is below the ENERGY_THRESHOLD, the sensor node sends a *reconstruction message* to notify the sink that the cluster must be reconstructed. After the sink receives the reconstruction message, the reconstruction of the cluster formation will be initiated by the sink. By using the data route maintenance mechanism, the energy consumption of each node can be balanced to prolong network lifetime.

# Chapter 4

# Performance Evaluation

In this chapter, we evaluate the performance of ECDD and compare it with directed diffusion (DD) [3, 4] and Energy Level-based Passive Clustering (ELPC) [10].

## 4.1 Simulation environment and parameters

The simulation programs were written in C++. Sensor nodes are randomly placed in a 160 x 160 $m^2$. The transmission range of each sensor node is 40 *m*. In the simulations, we set a single sink and five sources [7]. The sink is located in the bottom left corner of the sensor field and the five sources are randomly selected from the nodes in the sensor field. The sink periodically sends interests every 20 seconds, and the sources send data messages every 2 seconds after they receive an interest [11]. The data messages are 64-byte long for DD and ECDD and 72-byte long for ELPC. The interests are 57-byte, 49-byte and 36-byte long for ECDD, ELPC, and DD, respectively. The initial energy of a sensor node is 10 *J*. The transmit power is 0.66 *W* and the receive power is 0.395 *W*. The simulation time is 1000 seconds. We evaluate each scheme with the following parameters: *average dissipated energy*, *average delay*, *throughput*, and *the time till the first node death*, which are defined as follows.

- *Average dissipated energy*

The average dissipated energy ($E_{avg}$) measures the ratio of total dissipated energy of all nodes in the network to the number of distinct events seen by the sink [3, 4]. As defined in equation (1).

$$E_{avg} = \frac{\sum_{k=1}^{m}(Ei_k - Ef_k)}{n} \quad (1)$$

where $Ei_k$ is the initial energy of the $k^{th}$ node, $Ef_k$ is the final energy of the $k^{th}$ node, $m$ is the total number of sensor nodes, and $n$ is the total number of events.

- *Average delay*

The average delay ($D_{avg}$) measures the average transmission time of events, which indicates the average latency between a source that transmits an event and the sink that receives the event [14]. The definition of $D_{avg}$ is as follows:

$$D_{avg} = \frac{\sum_{i=1}^{n}(Ts_i - Te_i)}{n} \quad (2)$$

where $Ts_i$ is the timestamp when the interest of the $i^{th}$ event is transmitted from the sink, $Te_i$ is the timestamp when the $i^{th}$ event is received by the sink, and $n$ is the total number of events.

- *Throughput*

The throughput measures the ratio between the number of successfully received data packets at the sink and the total elapse time of the sensor network. This metric indicates the performance of data dissemination by the sensor network.

$$Throughput = \frac{Successfully\ received\ data\ packets}{Total\ elapse\ time} \quad (3)$$

- *The time till the first node death (FND)* [6]

The FND represents the timestamp that the first sensor node dies due to the depletion of energy [6]. It indicates whether the energy consumption of nodes in the sensor field is balanced or not. On the other hand, for a cluster-based scheme, this parameter also indicates the frequency of the re-construction of the cluster formation because when a sensor node (usually a cluster head) failed, the clusters must be reconstructed.

## 4.2 Comparison of the proposed ECDD with DD and ELPC

Figure 10 shows the average dissipated energy of DD, ELPC and ECDD. The DD protocol consumes more energy than the others because the interests sent from the sink are flooded all over the sensor field and so do as the exploratory data messages in order to establish a reinforced forwarding path. Because the ELPC protocol only floods interests among cluster heads and border nodes, it consumes less energy than DD. However, the ELPC protocol also floods the exploratory data messages to find a forwarding path, which causes additional energy consumption. The energy consumption of ECDD is the lowest because ECDD uses a cluster-based scheme to avoid the interests flooding problem and uses hop count information to avoid the exploratory data message flooding problem. Simulation results show that ECDD is 61.30% and 22.33% better than DD and ELPC, respectively, in terms of average dissipated energy.

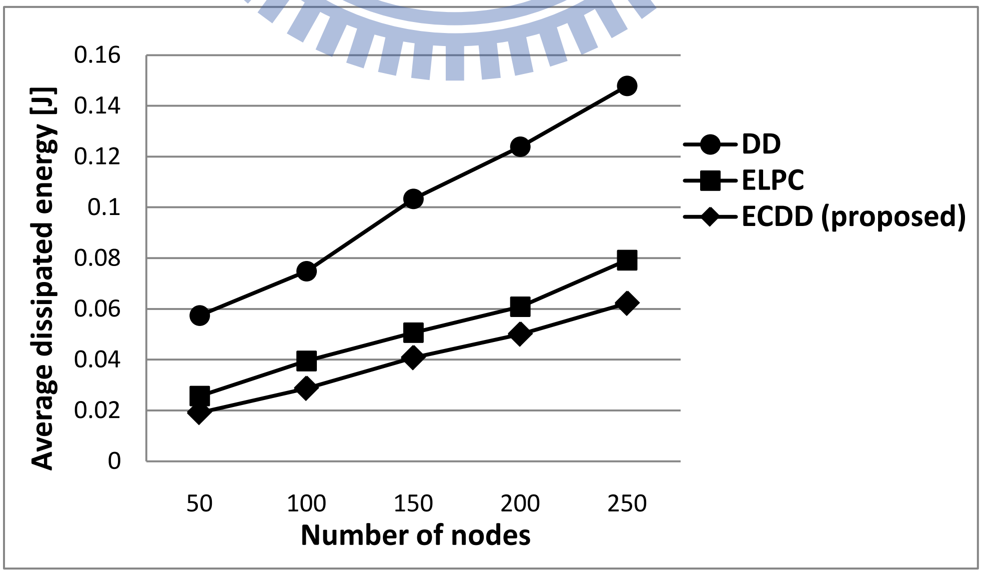



**Figure 10. Average dissipated energy.**

Figure 11 shows the average delay of each approach. The average delay of DD is the worst because DD floods interests and exploratory data messages to establish forwarding paths. In ELPC, the average delay is better than DD because ELPC uses clusters to reduce the number of flooded exploratory data messages. Although the number of flooded exploratory data messages in ELPC is decreased, the flooding problem still exists and causes additional overheads. The proposed ECDD has the lowest average delay because we use hop count information to further eliminate flooding of exploratory data messages. Simulation results in Figure 11 show that the average delay of the proposed ECDD is 57.45% and 23.49% better than DD and ELPC, respectively. Because ECDD uses hop count information to select the shortest forwarding path instead of flooding exploratory data messages, ECDD achieves the lowest average delay.

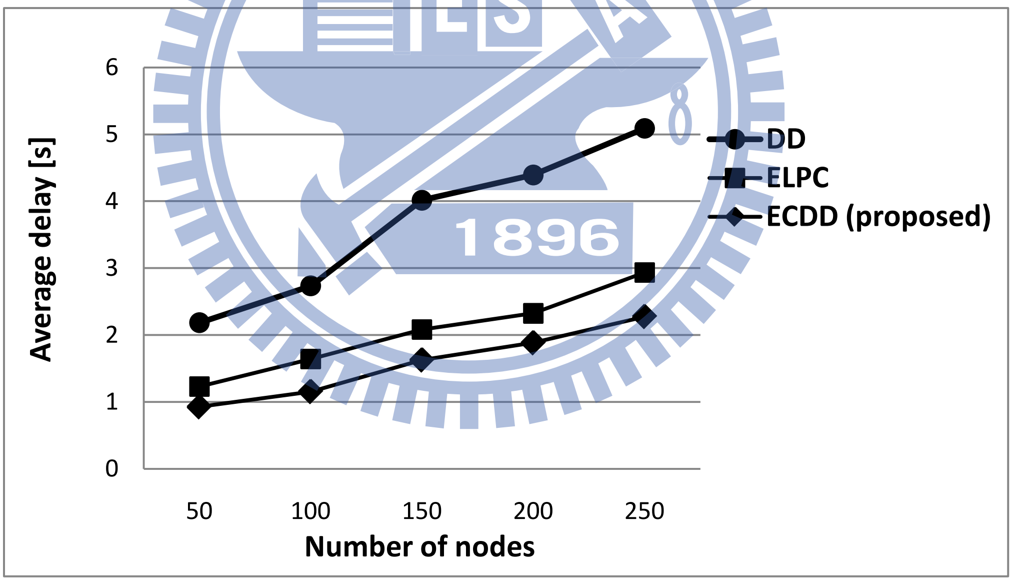



**Figure 11. Average delay.**

Figure 12 shows the comparison of throughput. In DD, sources forward data to the sink after the sink floods interests to the sensor field and the sources send exploratory data messages back to the sink. Therefore, DD has the lowest throughput. Although ELPC reduces the degree of the interest flooding problem by using clusters, which makes ELPC better than DD, ELPC still must flood exploratory data messages among cluster heads and border nodes to find a forwarding path. By piggybacking additional control information into interests, ECDD can establish clusters by sending interests from the sink. Thus, it has the highest throughput. Simulation results show that the proposed ECDD is 55.54% and 15.85% better than DD and ELPC, in terms of throughput.

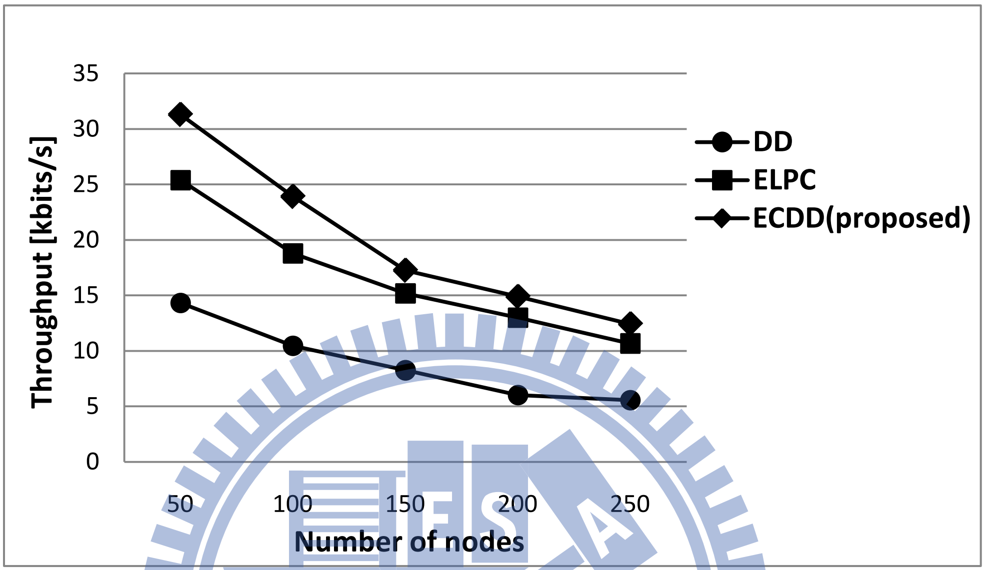



**Figure 12. Throughput.**

Figure 13 shows the comparison of the time till that the first node death (FND) among the proposed ECDD, DD, and ELPC. Because both ELPC and ECDD use a cluster-based mechanism, cluster heads and border nodes will deplete their energy faster than cluster members. In spite of the energy consumption of each node is balanced in ELPC, the amount of exploratory data messages will increase when the network scale increases. Thus, it causes cluster heads and border nodes to deplete their energy faster than the proposed ECDD. Simulation results show that the proposed ECDD is 71.62% and 24.40% better than DD and ELPC, respectively, in terms of FND.

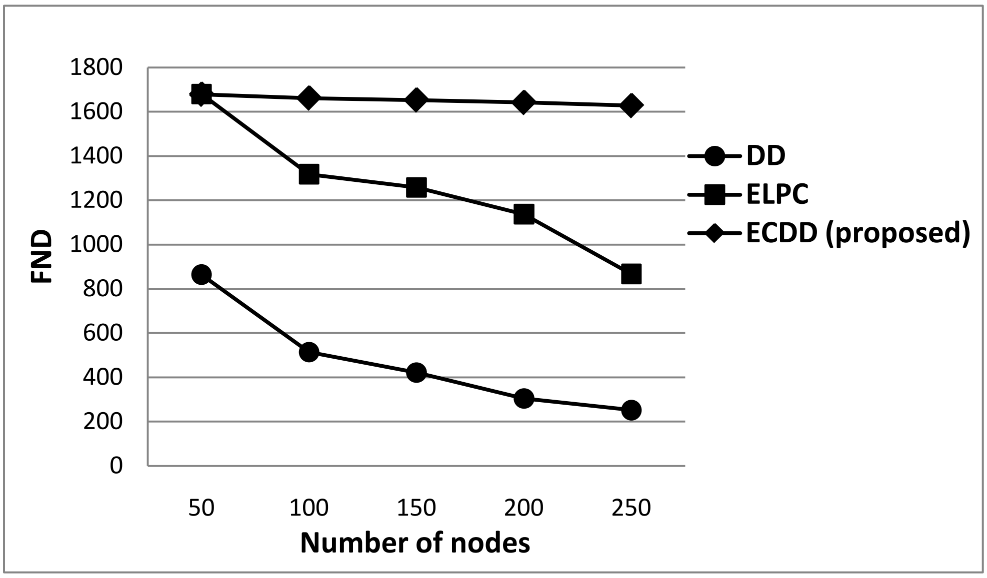



**Figure 13. The time till the first node death (FND).**

Because the proposed ECDD uses a neighbor information table to store the control information of sensor nodes, its overheads are the update and the maintenance of the neighbor information table, which is required when the cluster formation must be reconstructed. Although the overheads of the reconstruction of cluster formation are high, the frequency of performing the reconstruction of cluster formation is small. Thus, the cost of performing the reconstruction of cluster formation is much lower than the cost of repeatedly flooding exploratory data messages.

# Chapter 5

# Conclusion

## 5.1 Concluding remarks

We have presented an efficient cluster-based data dissemination (ECDD) scheme that piggybacks hop count information in an interest to each node. The hop count information can assist a node to select the next forwarding node with the least hop count. In this way, without flooding exploratory data messages, a shortest data dissemination path to forward sensed data back to the sink can be found, while saving energy and reducing delay. Simulation results have shown ECDD is 61.30% and 22.33% better than DD and ELPC in terms of average dissipated energy, respectively. Furthermore, ECDD is 57.45% and 23.49% better than DD and ELPC in terms of average delay, respectively. Besides, ECDD is 55.54% and 15.85% better than DD and ELPC in terms of throughput, respectively. In addition, we also piggyback an energy threshold into an interest to balance the energy consumption of sensor nodes and prolong network lifetime. Simulation results have also shown that, in terms of the time till the first node death (FND), the proposed ECDD is 71.62% and 24.40% better than DD and ELPC, respectively.

## 5.2 Future work

The proposed ECDD can be applied to real environments, such as a community health care system, rescue operation in a disaster area, and so on, where it is essential for wireless sensor networks to transmit long-term monitoring data and real-time sensed data back to the sink effectively.

# Bibliography

[1] I.F. Akyildiz, S. Weilian Su, Y. Sankarasubramaniam, E. Cayirci., "A Survey on Sensor Networks," *IEEE Commun. Mag.*, Vol. 40, Issue 8, pp. 102-114, Aug. 2002.

[2] D. Culler, D. Estrin, M. Srivastava, “Guest Editors’ Introduction: Overview of Sensor Networks,” *IEEE Computer*, Vol. 37, Issue 8, pp. 41- 49, Aug. 2004.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, pp. 56-67, Aug. 2000.

[4] IIntanagonwiwat, R. Govindan, D. Estrin , J. Heidemann, F. C. Silva, “Directed Diffusion for Wireless Sensor Networking,” *IEEE/ACM Trans. Netw.*, Vol. 11 , Issue 1, pp. 2-16, Nov. 2003.

[5] Y. Cui , J. Cao, “An Improved Directed Diffusion for Wireless Sensor Networks,” in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2380-2383, Sept. 2007.

[6] A. Ozgovde, C. Ersoy, “WCOT: A Realistic Lifetime Metric for the Performance Evaluation of Wireless Sensor Networks,” in *Proceedings of the IEEE 18$^{th}$ International Symposium on Personal, Indoor and Mobile Radio Communication*, pp.1-5, Sept. 2007.

[7] X. Liu, F. Li, H. Kuang, X.Wu, “The Study of Directed Diffusion Routing Protocol Based on Clustering for Wireless Sensor Network,” in *Proceedings of the IEEE 6$^{th}$ World Congress on Intelligent Control and Automation*, Vol. 1, pp.5120-5124, Oct. 2006.

[8] J. K. Taek, M. Gerla, V. K. Varma, M. Barton, T. R. Hsing, “Efficient Flooding with Passive Clustering-An Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Network,” *IEEE/ACM MobiHoc*, Vol. 91, Issue 8, pp.1210-1220, Aug. 2003.

[9] M. Gerla, T. J. KOWN, G. Pei, “On Demand Routing in Large Ad Hoc Wireless

Networks with Passive Clustering," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 1, pp.100-105, Aug. 2002.

[10] H. Zeghilet, N. Badache, M. Maimour, "Energy Efficient Cluster-based Routing in Wireless Sensor Networks," in *Proceedings of the IEEE Symposium on Computers and Communications*, pp.701-704, Aug. 2009.

[11] V. Handziski, A. Kopke, H. Karl, C. Frank, W.Drytkiewicz, "Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering," in *Proceedings of 1st European Workshop in Wireless Sensor Networks*, Vol. 2920, pp.172-187, 2004.

[12] Y. Zhang, L. Wang, "A Comparative Performance Analysis of Data Dissemination Protocols in Wireless Sensor Networks," in *Proceedings of the IEEE 7th World Congress on Intelligent Control and Automation*, pp.6669-6674, June 2008.

[13] A. Booranawong, W. Teerapabkajomdet, "Reduction of Exploratory Data Messages on Directed Diffusion in Mobile Wireless Sensor Networks," in *Proceedings of the IEEE 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Vol. 2, pp.996-999, May 2009.

[14] E. Lee, S. Park, F. Yu, Y. Choi, M. S. Jin, S. H. Kim, "A Predictable Mobility-Based Data Dissemination Protocol for Wireless Sensor Networks," in *Proceedings of the IEEE 22nd International Conference on Advanced Information Networking and Applications*, pp.741-747, March 2008.