Hybrid Hierarchical Agglomerative Clustering Protocol for Wireless Sensor Networks

by

Jiang Zhu

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment to the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems & Computer Engineering
Carleton University
Ottawa, Ontario, Canada

September 2012

Your file Votre référence
ISBN: 978-0-494-93645-0

Our file Notre référence
ISBN: 978-0-494-93645-0

The undersigned hereby recommends to

the Faculty of Graduate Studies and Research

Acceptance of the thesis

Hybrid Hierarchical Agglomerative

Clustering Protocol for Wireless Sensor

Networks

Submitted by Jiang Zhu

In partial fulfillment of the requirements for the degree of

Master of Applied Science

Chair, Department of Systems and Computer Engineering

Thesis Supervisor, Professor Chung-Horng Lung

Carleton University

September 2012

Abstract

Clustering is one of the most energy-efficient ways to organize sensor nodes in Wireless Sensor Networks (WSNs). To perform clustering, location data are normally used for calculating the distance between sensor nodes. But location data may not always be available due to GPS failures or consideration of cost. Alternatively, Received Signal Strength (RSS) or RSS Indicator (RSSI) is used as the distance estimator, but it has been proved that RSS or RSSI is unreliable in many studies. In order to mitigate these problems, this thesis proposes a hybrid clustering protocol — Hybrid Distributed Hierarchical Agglomerative Clustering (H-DHAC) protocol which uses both quantitative location data and qualitative connectivity data in clustering for WSNs. Our simulation results show that H-DHAC has a lower percentage of compromise in performance in terms of network life time and total transmitted data compared to similar approach that uses complete location data. Further, it still outperforms the well known clustering protocols, e.g. LEACH and LEACH-C.

Acknowledgments

First and foremost, I offer my sincerest gratitude to my supervisor, Professor Chung-Horng Lung, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

Thanks for the support from Cistech/Natural Sciences and Engineering Research Council (NSERC), Canada.

Many friends have helped me stay optimistic through these years. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me.

Finally, I am forever indebted to my parents and grandparents for their support, understanding, endless patience and encouragement when it was most required.

Contents

List of Tables

List of Figures

List of Acronyms

μAMPS	micro-Adaptive Multi-domain Power-aware Sensors
ATPC	Adaptive Transmission Power Control
AHP	Analytical Hierarchy Process
BS	Base Station
BSDCP	Base Station-controlled Dynamic Clustering Protocol
CDMA	Code Division Multiple Access
CH	Cluster Head
CL	Confidence Level
CLINK	Complete Linkage method
CM	Cluster Member
CSMA	Carrier Sense Multiple Access
DHAC	Distributed Hierarchical Agglomerative Clustering
DHAC-CON	DHAC with CONnectivity qualitative data
DHAC-LOC	DHAC with LOCation quantitative data
DWEHC	Distributed Weight-based Energy-efficient Hierarchical Clustering
EECS	Energy Efficient Clustering Scheme
EEUC	Energy Efficient Unequal Clustering
FCM	Fuzzy C-Means
FDMA	Frequency Division Multiple Access
G-MAC	Gateway-MAC
GPS	Global Positioning System
HAC	Hierarchical Agglomerative Clustering

H-DHAC	Hybrid Distributed Hierarchical Agglomerative Clustering
HEED	Hybrid Energy-Efficient Distributed clustering
H-PEGASIS	Hierarchical PEGASIS
LACBER	Location Aided Cluster Based Energy-efficient Routing
LEACH	Low-Energy Adaptive Clustering Hierarchy
LEACH-C	LEACH-Centralized
LMAC	Lightweight MAC
MAC	Medium-Access Control
PEGASIS	Power-Efficient GAthering in Sensor Information Systems
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SLINK	Slingle LINKage method
S-MAC	Sensor MAC
SPIN	Sensor Protocols for Information via Negotiation
TDMA	Time-Division Multiple Access
TMH-DHAC	Tree-based Multiple-Hop DHAC
TPC	Transmission Power Control
UPGMA	Unweighted Pair-Group Method with arithmetic Averages
WINS	Wireless integrated network sensors
WPGMA	Weighted Pair-Group Method with arithmetic Averages
WSN	Wireless Sensor Network

Chapter 1

Introduction

Sensors have been around us for a long time; their existence greatly improves our living environments by offering convenience in many aspects. There are numerous types of sensors, for example, smoke detector, voice operated switch, velocity transducer, motion detector, etc. Normally, the traditional wired sensor networks need careful planning before deployment and attentive maintenance afterwards.

With the recent advancements in technologies and microelectronic fabrication techniques, tiny size, low-cost, low-power sensors with wireless communication capability are able to be put into mass production. The appearance of wireless sensors brings several new features to the sensor society. In applications using wireless sensors, the interested areas are covered by Wireless Sensor Networks (WSNs) which are deployed with large number of sensor nodes. However, in many outdoor applications, the working environments of wireless sensors are inaccessible or even dangerous to mankind [1], such as military detection and disaster relief operation. In these situations, it is unnecessary or impossible to predetermine the position of sensor nodes. Those

applications require random deployment of wireless sensors which also implies these sensor nodes should work collaboratively and have the ability of self-organization. These new features have broadened the range of the applications for WSNs, which have evolved from initial military [2] purposes to many civil applications, such as environmental observation [3], traffic control [4], health care [5], etc.

1.1 Challenges and Requirements in WSNs

The unique features of WSNs have brought many benefits, including feasible deployment, free of infrastructure, unattended operations etc. However, there are some challenges in WSNs due to the complicated wireless communication environments and hardware constraints of wireless sensors. To address these challenges, several factors should be considered in the design of WSNs:

- In the traditional wireless networks, such as cellular networks, the energy consumption is never the first issue. However, in WSNs, the power supply of wireless sensors is usually limited, hence unwanted energy waste needs to be minimized and the average energy consumption should be low.
- In many applications, wireless sensors are deployed randomly which means the location of each sensor cannot be planned or known beforehand. Thus, localization of wireless sensors is needed.
- It is not practical or even possible to have a central organizer like a base station in

cellular network when wireless sensors are deployed in an inaccessible area. *Wireless sensors should have the ability of self-organization and self-adaptation.*

- ❖ Due to the hardware constraints, the processor of wireless sensors has limited computational capabilities. And the data transmission rate of WSNs is low compared to the traditional wireless networks.
- ❖ Since wireless sensors are normally deployed in harsh environments plus wireless communication environments are complicated by nature, problems and failures can occur unexpectedly. A robust WSN should be able to deal with these problems or prevent malfunctioning caused by minor failures.
- ❖ Data gathering in WSNs should reach certain efficiency depending on different applications. Generally, out dated data will become meaningless.

1.2 Practical Issues of Clustering in WSNs

Protocols in WSNs can be classified into two main categories by network structure — flat and hierarchical structured protocols [6, 7]. Hierarchical structured protocols have better scalability, data transmission efficiency, and load balancing compared to flat structured protocols. One major type of hierarchical structured protocols is clustering of sensor nodes.

Clustering is a data analysis approach that explores the characteristic of data objects and classifies them into clusters if they have high relevance. When it comes to

WSNs, two sensor nodes tend to be clustered together when they are in close distance. Usually, location data are used to calculate the distance between sensor nodes. Many papers have adopted location data for clustering, such as LEACH-C [8], DWEHC [9] and BSDCP [10]. In these papers, it is assumed that every sensor node aware of their location or have Global Positioning System (GPS) equipped. Since sensor nodes are randomly deployed in many applications, they cannot know their location beforehand without localization. GPS is the most accurate and direct localization method; however, it faces a certain possibility of failure and the cost of a sensor node will be higher with GPS equipped. Alternatively, some studies adopted Received Signal Strength (RSS) or RSS Indicator (RSSI) to estimate the distance between sensor nodes, such as LEACH [8], EECS [11] and EEUC [12]. However, RSS and RSSI are not reliable distance estimators in practice which has been proven in many studies with experiments in real live environments. More details will be discussed in Chapter 4.

1.3 Motivation and Thesis Contributions

In WSNs, the major concern has always been energy efficiency. Many protocols including clustering protocols have been proposed aiming to organize sensor nodes efficiently. However, in practice, many unexpected or uncontrollable difficulties may occur due to the complicated network environments, such as poor channel quality, sudden sensor node failures, etc. Most of the existing clustering protocols are based on the ideal

case and fail to take some key problems and difficulties into consideration.

The motivation of this thesis is to provide a robust hybrid protocol in order to mitigate the problems of GPS unavailability and low reliability for distance estimation using RSS or RSSI without compromising much in energy efficiency. The hybrid approach makes use of the variation of DHAC [13, 14]. By exploiting the advantages of using two different kinds of data — quantitative location data and qualitative connectivity data in clustering — the assumption or requirement for each sensor node to have location data is relieved. Although connectivity data are not as accurate as the location data, the error level is expectable or controllable. Besides, connectivity data are always available to all sensor nodes as long as the radio communication component is functioning. The benefits of the hybrid approach are reduced cost for GPS (not all nodes need GPS) and increased reliability in the presence of unreliable GPS data and/or RSS (or RSSI) data.

The Hierarchical Agglomerative Clustering (HAC) [15] algorithm is an simple and effective centralized clustering approach which has been successfully applied in many areas. Distributed Hierarchical Clustering (DHAC) [13, 14] adapted HAC to WSNs as a distributed bottom-up hierarchical clustering protocol without having global knowledge. This thesis proposes a Hybrid DHAC (H-DHAC) protocol of which the fundamental mathematical model is adopted from HAC. This protocol further modifies the clustering approach of DHAC by extending it to support hybrid data. Furthermore, this protocol works with any percentage of location data availability (0% — 100%). As a

static clustering protocol, H-DHAC exploits the immobility of most WSNs. Therefore it not only provides flexibility for those who may have different specific demands on cost or performance, but also increases robustness by considering possible GPS failure.

The main contributions of this thesis are summarized as follows:

- Developed a hybrid protocol that uses two different kinds of data — quantitative location data and qualitative connectivity data for clustering. H-DHAC extended the original HAC algorithms and DHAC protocol. DHAC can only use either one of data types (location or connectivity) for the entire lifetime and has a strong assumption on reliability of the data.
- Designed two novel control parameters: confidence level and C_{MIN} (the minimum number of coefficients for coefficient estimation) to minimize the errors and randomness in clustering and ensure the robustness.
- Increased the flexibility in protocol design. H-DHAC is able to work with any percentage of location data availability (from 0% to 100%).
- Purposed coefficient estimation schemes for four commonly used HAC methods: SLINK, CLINK, UPGMA and WPGMA to deal with missing location data.
- Integrated transmission power control (TPC) scheme in the proposed protocol, the transmission power can be effectively reduced in practical environments by using TPC.

A paper based on the thesis is to be submitted to a conference:

J. Zhu and C. H. Lung, "H-DHAC: A Hybrid Clustering Protocol for Wireless Sensor Networks".

1.4 Thesis Organization

The remaining part of this thesis is organized as follows:

Chapter 2 introduces the background knowledge of WSNs with reviews on some related works.

Chapter 3 provides the basic concept of the HAC algorithms with an example for illustration.

Chapter 4 discusses the reliability of RSS or RSSI as distance estimator and the vulnerability of GPS. This chapter also presents H-DHAC.

Chapter 5 presents the simulation results and discussions. H-DHAC with four different percentages of location data availability is simulated in different conditions, and H-DHAC outperforms LEACH, LEACH-C and DHAC-CON (connectivity) in all scenarios and with minimum compromise compared to DHAC-LOC (location) which is the theoretical upper bound for H-DHAC as DHAC-LOC assumes 100% of the location information is available.

Chapter 6 summarizes this thesis and offers some suggestions for future work.

Chapter 2

Background of WSNs and Related Works

Basic background knowledge of wireless sensor network will be provided in this chapter, with details of the WSNs common architecture, hardware, protocols etc. Related works of WSNs, especially existing clustering techniques will also be illustrated.

2.1 General View of Wireless Sensor Networks

Wireless sensor network research is a field that attracts many attentions from researchers and industries. This field is now advancing faster under the push of emerging applications and the development of advanced technologies.

A WSN is a network that has hundreds or even thousands of sensor nodes deployed in the interested area. A sensor node normally has the functions of sensing, computing, signal processing and communicating [16]. In WSNs, sensor nodes are used to do various kinds of job, such as detecting a certain event, sensing the desired information in its surrounding area, or providing status update upon user's requests.

The architecture of a typical WSN is shown in Figure 2-1. The target sensing area which is called sending field, is normally deployed with a mass of sensor nodes. Since

the sensing field of many outdoor applications may be inaccessible or even dangerous to human (such as human detection for military purpose in a hostile area), it's difficult to control the precise location when the sensor nodes are deployed. Hence in many cases, sensor nodes of WSNs are deployed randomly in the sensing field which is different from the traditional wired sensor network. When sensor nodes start working, they firstly collect data of their own sensing area, then process the sensed data and send it to base station (BS) in the end. The sink/BS acts like a gateway not like in cellular network; it does not involve with many network management work. The BS will aggregate the collected data and forwards it to user or the administrator through a reliable network. The location of a BS could be inside the sensing field. However, it's more realistic to have BS located outside the sensing field because of the same reason: the sensing field could be inaccessible or dangerous to mankind.

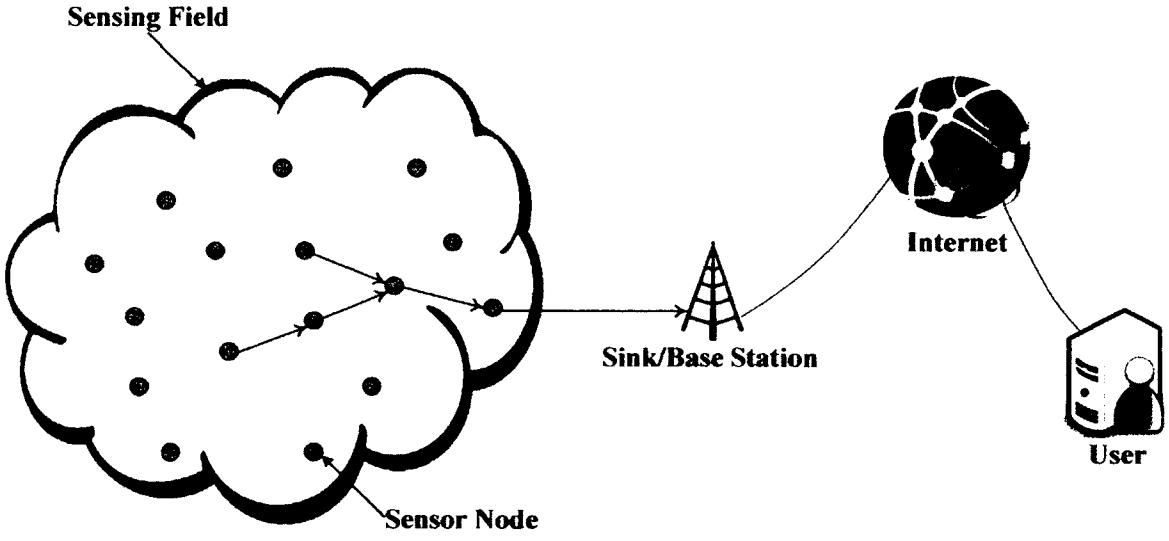


Figure 2-1 Architecture of a typical WSN

Although WSNs have different kinds of applications, they can be classified into three major types in terms of data delivery model: query-driven, event driven or continuous data generation applications. In the query-driven application, sensor nodes will sense and send the data back to BS upon receiving the query that was initiated by BS. For event-driven application, usually there is certain type of event that's interested, and sensor nodes need to perceive and detect it. Since there may be several sensor nodes deployed in a close area, the interested event may be reported by more than one sensor node. In the continuous data generation application, sensor nodes will repeat the sensing and data transmission task in period. Note that these three types of applications are not exclusive. In some cases, they can be combined as a hybrid application.

2.2 Sensor Architecture

As the essential element of WSNs, sensor nodes are built in compact size and integrated with several basic units. The low cost characteristic has made sensor nodes disposable, along with the self-organizing capability, the difficulty of large number outdoor deployment has been eased.

Usually a sensor node is embedded with power-efficient sensing unit, processor, radio communication unit and power supply unit [17]. The general system architecture of a wireless sensor node is shown in Figure 2-2. The radio communication unit which is usually a radio transceiver with antenna, will transmit and received the interested data

within or outside the network. The sensing unit is responsible for detecting changes in physical condition and transforms it to digital form. There are different kinds of sensor that deals with specific environments, such as light, pressure, temperature, humidity etc. The information detected by sensing unit will past to processor to process. After the processor finished its job, the processed data will pass to radio unit for data transmission. Normally, there is a memory unit to store the detected information inside of a processor. The power supply unit is usually a battery, or solar powered in some cases. These four components are only the basic unit of a wireless sensor node. It's possible to have some add-ons to provide more features. For example, Global Positioning System (GPS) chip can be embedded in sensor nodes to provide precise location.

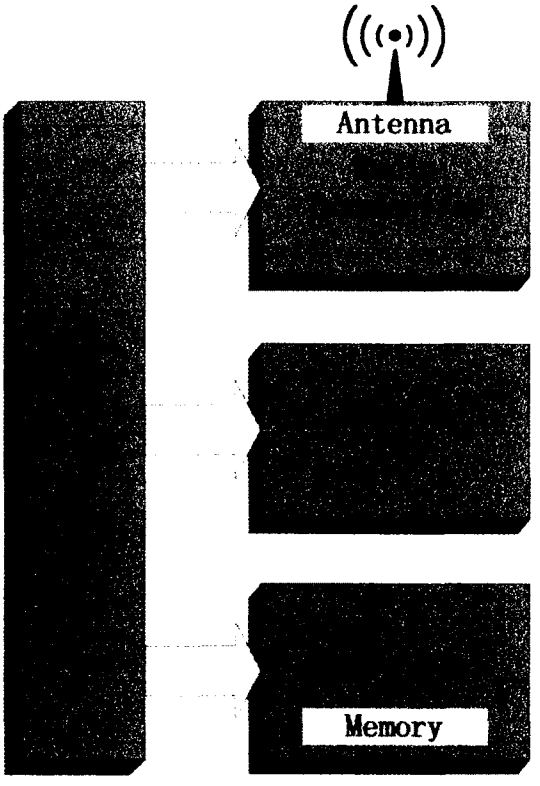


Figure 2-2 System architecture of a wireless sensor node

When it comes to integrating these four basic units into one sensor node, there are some challenges cause by limitations and constraints of each unit. The battery is expected to supply a sensor node for months or even years, which means large amount of energy stock is required. However, in many outdoor applications, since sensor nodes are deployed in harsh environments, battery recharging and replacing is impossible. For sensing unit, whether it detects the information in time and whether the data detected accurately reflect the status of surrounding area is the challenge. The embedded processor normally has limited processing ability due to the consideration of cost and energy consumption, so it will be difficult for it to run complicated algorithms. The memory could exceed and the processor may not be unable to process if there are too many data. The radio communication unit must run under an efficient protocol in order to achieve energy efficiency and lessen the burden of power supply unit.

There are many projects dedicated to overcome these challenges, researchers have paid efforts on integrating these four basic units together in an energy efficient way. Such as the Smart Dust project [18] of UC Berkeley in 1988, where they tried to built a very small sensor node, even as tiny as dust. Besides, there is another project at UC Berkeley called PicoRadio [19] which focus more on radio communication quality and aim to develop a low-cost low-power sensor node. iBadge [20] is a sensor nodes that developed by researchers of UCLA which embedded with two processors and Bluetooth chip. In addition, there are also other projects which have addressed different issues, such as

μAMPS (micro-Adaptive Multi-Domain Power-Aware Sensors) [21] project and WINS (Wireless Integrated Network Sensors) [22] project of Rockwell Science Center etc. All these different projects and different sensor nodes developers have produced show the tremendous variety in sensor market, which leads a bright future of wireless sensor applications.

2.3 Critical Design Factors of WSNs Protocols

Although there are many different sensor applications, the major concern of WSNs is the energy consumption issue. As discussed in previous section, it's almost impossible to replace or recharge the main power supply unit — battery in many outdoor applications. Thus, in order to let the sensor nodes keep running for a long period, the protocol designed for WSNs must be efficient enough. When an interested event occurs, there might be several sensor nodes around that area detect it at the same time. This redundancy in collected data should be reduced by using properly designed data to minimize it to improve the efficiency and channel utilization. For many applications, it's difficult and impractical to have a central coordinator. Hence sensor nodes themselves should have self-organization and self-adaptation ability to work in harsh or remote environments. WSNs is very different from traditional network because environmental and hardware constraints, therefore, the way to design WSNs protocols should be different. Here are some general factors that should be considered in designing WSNs

protocols:

- Energy efficiency

  The power supply unit of a sensor node is usually battery. The progress of developing small battery with high capacity is really falling behind compared to the fast advancement in nanometer processor area. And for many applications, batteries cannot be recharged or replaced due to the inaccessible environments. Since the constraints of battery can hardly break through, the energy efficiency of running protocols become vital. Among all energy consuming parts, radio communications expend the highest amount of energy. The balance of data collecting and energy saving should be achieved by careful consideration in protocol design. Moreover, the redundancy of unprocessed raw data could be huge. Proper data fusion technique can save a lot of data traffic without losing fidelity.

- Self-organization and self-adaptability

  Wireless communication environments are very complicated, the situation is even worse for WSNs which are set up in harsh and even hostile environments. In this kind of complicated network environment, having a central coordinator is impractical or even impossible. Thus sensor nodes must do the organization, transmission scheduling and maintenance jobs by themselves in a distributed way. Besides, since in most cases sensor nodes are not replaceable, there might be an existing sensor node leaves the network (when battery ran out) or new sensor node

joins the network (a newly deployed sensor node). So sensor nodes should be adaptable to these changes of WSNs.

- Flexibility and scalability

  In practice, there are many unexpected problems and difficulties due to the complicated network environment. The designed protocol should be flexible enough to deal with problems that encountered unexpectedly. For example, the protocol should continue to run without being disrupted if some sensor nodes' location data are missing. The density of sensor nodes and monitored field size could vary even for the same application depends on different requirements. Hence the protocol should be able to work in different network scale, in other word, it should have good scalability.

- Latency and accuracy

  Out dated data is meaningless for many real-time monitoring applications. An efficient WSN protocol ought to provide interested data in time with low latency. On the other hand, unprocessed data contains all details but relatively large, and it will take much more time to deliver compared to compressed data. Moreover, faster data update will cause higher energy consumption. After all, it's the designed protocol's job to balance among energy efficiency, accuracy and latency. A protocol with high accuracy, low latency and good energy efficiency is desired.

- Fault tolerance

For traditional wired network or ad-hoc network, errors barely occur or controllable. When it comes to WSNs, certain degree of error will occur and may not be fixed easily, because sensor nodes are randomly deployed, the network environments are intricate and there is no central organizer. The communication between some sensor nodes could be block by obstacles. Some sensor nodes might stop working earlier than others because batteries drained. However, small number of failures should not bring disaster to the whole network. The other part of the WSN should continue to work without being affected. Hence the designed protocol is expected to be robust enough to deal with small failures and assure performance quality at the same time.

2.4 MAC Layer Protocols

The medium access control (MAC) layer is the intermediate layer between physical layer and network layer in the layered network model. In general, the MAC layer has several responsibilities: provide reliable data transmission, control device to access the channel at any time, control data flow and error [23]. When it comes to WSNs, the priority concerns are energy-efficiency, scalability and adaptability. While latency, throughput, fairness and bandwidth utilization are important factors for MAC layer protocols of other wireless networks, they become secondary concern in WSNs [24]. In the following section, some common wireless MAC protocols as well as MAC protocols specifically for WSNs will

be introduced.

2.4.1 Common MAC Layer protocols of Wireless Networks

CSMA

Carrier Sense Multiple Access (CSMA) is a simple MAC protocol without having synchronization or slot allocation. There are two versions of CSMA: non-persistent CSMA and p-persistent CSMA. In non-persistent CSMA, if a device has data to send, it will detect whether the channel is idle first. The data will be sent out if nobody else is sending at that time. However, if the channel is occupied, the device will back-off for a while before it detects the channel availability again. In p-persistent CSMA, a device will keep detecting channel availability even the channel is busy. Then, when the channel is free, the device will decide whether it will send the data now by probability p or delay for a while by probability 1 – p.

FDMA

Frequency Division Multiple Access (FDMA) allocates users one or several channels by dividing one large frequency band into several small ones. The concept of FDMA is relatively simple, and the coordination among multiple users is not complicated as well. However, since the resource of frequency bands is very limited, FDMA is not efficient in terms of bandwidth utilization.

TDMA

Time Division Multiple Access (TDMA) allows multiple users share the same frequency band with no contention by dividing a time frame into different time slots and allocate them to users. There will be one or more time slots allocated to each user. Users can access the channel and transmit data when their specific allocated time slot comes, and will stop using the channel in other time. TDMA provides good efficiency and reduces collisions, but synchronization is necessary to correct possibly timing error.

CDMA

Code Division Multiple Access (CDMA) allows multiple users to access the same frequency band at the same time which is different from FDMA and TDMA. Instead of dividing frequency band or time frame, CDMA modulates the original signal with spreading code in the sender side, and the receiver side demodulates the processed signal by the same spreading code. There is a unique spreading code for each specific user. However, the CDMA modulated signal uses higher bandwidth compared to the original signal because the spread-spectrum effect.

2.4.2 MAC Layer protocols of WSNs

S-MAC

Sensor MAC (S-MAC) [25] is a one of the well know MAC layer protocols designed for WSNs in which it combines scheduling and contention scheme together. Each sensor node wake up or sleep according to its schedule, in this way, they can save energy during

the sleep period. To reduce the delay cause by periodic sleep, each sensor node will wake up for a short time in the end of transmission to see if there is any request from neighbors. S-MAC also adopts "message passing" method which is a message fragmentation technique that increases data transmission efficiency. It can break a larger message into smaller fragments and transmit it one at a time, so that only one fragment need to retransmit in case of corruption happens.

LMAC

Lightweight MAC (LMAC) [26] is a TDMA-based protocol. Sensor nodes can transmit data during their specifically assigned time slots. When the specific time slot comes, the sensor node will send out the data with a control message in front. The receiver's address is included in the control message, so when other sensor nodes receive the control message, they can turn off their radio component if the message are not intended for them. After receiving the control message, the receiver will continue to receive the remaining data while other sensor nodes save energy with radio switched off.

G-MAC

Gateway-MAC (G-MAC) [27] is a contention-based protocol that adopts the cluster head election idea of LEACH [8] and use it to elect gateway. The differences in the election process between LEACH and G-MAC is that G-MAC elects gateway base on residual energy, memory and other resources while LEACH's cluster head election is a probability-based approach. The gateway takes charge of scheduling and assigns time

slots for each sensor node. A sensor node only wakeup when its time slot comes and then goes back to sleep in the remaining data transmission period. In order to keep the energy and memory consumption level equal for every sensor node, gateway will re-elect periodically.

2.4.3 Taxonomy of Wireless MAC Layer Protocols

In general, wireless MAC layer protocols (including WSNs and other wireless network like ad-hoc network) can be classified into two types: fix-assignment and contention-based.

The control mechanism is relatively easy for contention-based protocols, however, the energy and data transmission efficiency of contention-based protocols are low since corruption and retransmission could occur frequently especially in high-density network. On the other hand, although fix-assignment approach might comes with control overhead, better performance with higher efficiency is provided because there will be nearly no collision. Because of the advantage of fix-assignment approach, many papers have adopted it for their MAC layer, and TDMA is the most popular one.

2.5 Network Layer Protocols in WSNs

Network layer protocols take in charge of structuring the network, organizing the data gathering and managing the network maintenance. There are two types of network layer

protocol in terms of network structure: flat structured protocol and hierarchical structured protocol [6, 7].

In flat structured protocol, the role of each sensor node is the same. Most flat structure protocols are data-centric based, sensor nodes response when BS sends out queries. The queried data are sent back to BS in multi-hop manner. Two typical flat structured protocols — SPIN (Sensor Protocols for Information via Negotiation) [28] and directed diffusion [29] laid the foundation of flat structure and inspired many other protocols which follow a similar idea.

Cluster-based protocol is the most common hierarchical structured protocol. Sensor nodes play different roles in hierarchical structured protocol, such as cluster head (CH) and cluster member (CM). As the high level sensor nodes, CHs are responsible for managing the CM, receiving and processing data from them. While low level sensor nodes — CM only need to sense data and send it to cluster head. Only CHs are allowed to communicate with BS directly, while CMs just simply communicate with their own cluster heads. Usually, CHs consumer more energy than their CMs since CHs have responsibility of network organization, data gathering and long distance data transmission with BS. Hence re-clustering or re-selecting cluster heads periodically is necessary to balance the energy consumption of each sensor node.

Hierarchical structured protocols have several advantages over flat structured protocols:

- Reduced data redundancy

  When an interested event occurs, several nearby sensor nodes may detect it simultaneously thus generate data with high relevance and redundancy. Cluster heads of hierarchical structured protocol gather data from cluster members which are close to each other and perform data fusion to reduce data size and redundancy.

- Better scalability

  For large scale network, hierarchical structure provides better organization and coordination. Better energy efficiency can be achieved by using CHs to assign sleep and wake schedule to CMs. Besides, with contention-free schemes implemented under hierarchical structure, high energy loss situations like collision and overhearing can be avoided.

- Balanced work load

  In cluster-based hierarchical structured protocols, complex work and long distance communication are performed by sensor nodes with higher energy while other sensor nodes with lower energy only perform basic sensing and short distance communication. Along with scheduled periodic sleep and re-clustering or cluster head re-selecting schemes, the flexible work load balance of hierarchical structured protocol prevents early death of sensor nodes in certain area and prolongs the network lifetime.

Although hierarchical structured protocol might not be optimal and need some

control overhead like synchronization, it shows promising advantages over flat structured protocol when it comes to WSNs. Hierarchical structured protocols offer a better organization of large scale network which results in better energy utilization of resource-constrained sensor nodes. With energy balancing and efficient communication schemes, hierarchical structured protocols can achieve better performance in WSNs.

Hierarchical structured protocols can be further classified into two categories: distributed hierarchical protocols and centralized hierarchical protocols. In the following section, related works of network layer protocols in these two categories are introduced.

2.5.1 Distributed Hierarchical Protocols

One important characteristic of distributed hierarchical protocols is the ability of self-organization without global knowledge and central coordinator. This characteristic provides feasibility and flexibility which are necessary for WSNs. To ensure the distributed approach functioning properly, a little control overhead is involved.

LEACH

Low Energy Adaptive Clustering Hierarchy (LEACH) [8] is a well-known energy efficient clustering protocol. In LEACH, CHs are elected randomly based on a probabilistic function. A sensor node which has not become CH for many rounds is more likely to be elected as CH than a sensor node has recently became a CH. The function for sensor nodes to determine whether they will become CHs is defined as follow:

$$T(N)\begin{cases} \dfrac{p}{\left(1-p\times(r \bmod \frac{1}{p})\right)}, & if \quad n\in G \\ 0 \qquad\qquad , & otherwise \end{cases} \tag{2-1}$$

Where p (for LEACH, p = 5% is the optimal percentage) is the predefined percentage of CHs in all sensor nodes; the current round number is r; G is the set of sensor nodes that have not become CH in the last 1/p rounds; if a sensor node has became CH in the last 1/p rounds, it won't be elected as CH in this round.

The operation of each round in LEACH consists of two phases: cluster formation phase and steady data collection phase. In the beginning of each round, sensor nodes generate random numbers between 0 and 1. If the generated number T(N) of node-N is less than the predefined threshold, this sensor node will become a CH and broadcasts an advertisement message. Other sensor nodes will hear the advertisement messages from several CHs and choose the closest one to join (depending on the measured received signal strength) and become CMs with the chosen CHs. After clusters have been formed, the CHs will create TDMA-based schedule which assign time-slot for each CM. Then the created data transmission schedules will be sent to CMs, and each CM transmits its data to CH according to the schedule. In the end of each round, CHs perform data aggregation on gathered data and send them to BS directly. After a certain period, a new round will start and re-clustering will be performed.

The main constraint of LEACH is the randomness of the CH election. Since the

CH election process is based on a probabilistic function, a sensor node with low energy compared to others can still be elected as CH which unbalanced the energy dissipation among all sensor nodes. Moreover, the randomness also causes uncontrollable number of CHs and uneven distribution of clusters in each round which have bad effects on the energy efficiency.

Still, LEACH is the one of the first protocols that introduces hierarchical clustering into WSNs, its concept and model has inspired many follow up researches and studies. Even in today, LEACH is still being used for discussion and performance comparison.

HEED

Hybrid Energy-Efficient Distributed (HEED) [30] clustering is an improvement from LEACH. HEED reduces the effect of randomness in LEACH by setting up well-distributed clusters, and improves the energy balance by considering energy status in clustering.

Different from LEACH, in which each sensor node is elected as CH determined by possibility without consideration of the energy status, HEED uses the residual energy of each sensor node as the main factor in CH selection. The probability of sensor nodes to become tentative CHs is calculated by:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{\max}} \tag{2-2}$$

In equation (2-2), the predefined initial percentage of CHs among all sensor nodes is given as C_{prob}; the current residual energy and maximum energy of the concerned sensor node is presented by $E_{residual}$ and E_{max}, respectively.

In HEED, a sensor node with high energy is more likely to become a CH. The CH selection process includes several iterations. At the beginning, if a sensor node becomes a tentative CH, it will inform its neighbors with advertisement message so that its neighbors will not try to become tentative CH themselves. Then, if two tentative CHs are too close, a second measurement which involves neighbor proximity or node degree will be used to select one of them. The CH_{prob} will be doubled in each iteration. Then, after several iterations, the CH selection process is finished when the CH_{prob} reaches 1 and the tentative CH becomes the final CH. The inter-cluster transmission of HEED is in multi-hop manner, a CH far from BS will send its data to BS through nearby CHs instead of sending it to BS directly.

HEED avoids the problems of uneven cluster distribution and unbalanced energy consumption in LEACH, which result in better network lifetime and energy efficiency over LEACH.

PEGASIS

Power-Efficient GAthering in Sensor Information Systems (PEGASIS) [31] is a chain-based protocol. Every sensor node is organized into one chain by the greedy algorithm, and only one sensor node (head node) in this chain is selected to communicate

with BS. The head node rotates periodically to balance the energy consumption. The chain is started from the sensor node closest to the BS, and then this sensor node connects with its nearest neighbor according to the measured received signal strength. Every newly joined sensor node follows the same manner and connects with its closest neighbor until all of sensor nodes are in the chain.

In PEGASIS, each sensor node only communicates with its two one-hop neighbors. When a sensor node receives the data from its neighbors, it will aggregate the data with its own and send them to the leader node. In this way, the data gathering is performed throughout the chain until the data reach at head node and be sent to the BS. By minimizing the average transmission distance through establish a logic chain; PEGASIS outperforms LEACH in many aspect. However, since the data need to traverse the chain, excessive delay is introduced by PEGASIS. This problem becomes worse with larger network size which indicates the major drawback of PEGASIS is low scalability.

H-PEGASIS

Hierarchical PEGASIS (H-PEGASIS) [32] protocol is proposed to address the delay issue in PEGASIS. Simultaneously data transmission scheme is adopted to reduce the delay. To avoid collision, sensor nodes that allow simultaneously data transmission can either be CDMA capable or spatially separated. A tree like hierarchy is formed on the chain-linked sensor nodes which separates sensor nodes in several levels. A sensor node aggregates data from its neighbor's and transmit them to upper level. This approach

allows parallel data transmission which results in significant delay reduction.

DHAC

Distributed Hierarchical Agglomerative Clustering (DHAC) [13, 14] is a novel distributed static clustering protocol that introduces bottom-up approach into WSNs clustering. DHAC is inspired by the centralized HAC algorithm [15] and adapts HAC to WSNs in a distributed manner. Unlike most of clustering approaches which use top-down structure, clusters are formed before CH election in DHAC's bottom-up approach. As a comprehensive clustering protocol, DHAC can use location data, RSS or connectivity data individually as the input data set for clustering. Only local information is required in the process of clustering, the necessity of global knowledge in original HAC algorithm is eased. The immobility characteristic of most WSNs is exploited by DHAC to avoid re-clustering, which results in higher energy efficiency. In the simulation, DHAC shows better performance than LEACH and LEACH-C in various aspects.

TMH-DHAC

Tree based Multi-Hop DHAC (TMH-DHAC) [33] protocol derives the idea from DHAC and extends DHAC by adopting a multi-hop transmission tree structure in clustering. In TMH-DHAC, distributed clustering is performed to generate a loop-free minimum spanning tree of links between each sensor node. The multi-hop tree structure of a single cluster provides energy-efficient routes for data transmissions. Residual energy of each sensor node is an important factor in the CH election. A tree structure repairing scheme is

applied when there are some changes in the network. To reduce the delay in a tree structure, an interference-avoidance mechanism is integrated with this protocol to enable simultaneously data transmissions. The simulation results show that TMH-DHAC has better performance over LEACH, LEACH-C and DHAC-RSS (Received Signal Strength).

2.5.2 Centralized Hierarchical Protocols

Normally, there is a central coordinator which organizes the network and makes all decisions in centralized approaches. In WSNs, the central coordinator is usually the BS. With global knowledge of every sensor node, centralized approach can provide efficient management of the network.

LEACH-C

LEACH-Centralized (LEACH-C) [8] is the centralized version of LEACH which improves LEACH by generating fix number of CHs and taking energy status into account in the CH selection process. The steady data transmission phase is the same as LEACH while the cluster formation phase is different. In the beginning of cluster formation phase, each sensor node sends its information which includes location and residual energy information, to BS directly. The location information is assumed to be available through localization methods such as Global Positioning System (GPS). After receiving the information of each node, the BS calculates the optimal cluster formation for this round

and generates the data gathering schedules for each cluster. The BS broadcasts the decisions to the network in a message, and then each sensor node knows its role as CH or CM and starts data gathering according to the schedule upon receiving the message.

As a centralized approach, LEACH-C performs a lot better than LEACH since LEACH-C reduces the randomness and generates optimal clusters. However, the performance of LEACH-C could have a huge drop when the cost for each sensor node to communicate with BS directly is large (usually it happens when the BS is far away from the network).

AHP

Analytical Hierarchy Process (AHP) [34] is a centralized clustering approach designed for improving the CH selection with support of mobile sensor nodes. There are three main factors considered in AHP: sensor node energy, mobility and the distance to the involved cluster centroid. Similar to LEACH-C, each sensor node sends their information (including ID, velocity and residual energy) to the BS in the beginning of clustering. Then, the BS selects CHs based on those three factors and broadcast the result to the network. The data gathering process of AHP is also similar to LEACH-C.

However, the CH re-selection is different from LEACH-C. The Ch re-selection is not performed periodically but triggered by events like a CH ran out of energy or moved to other location. AHP is a more complex approach compared to LEACH-C, since more information is needed for clustering, the energy consumption of communication between

sensor nodes and the BS can be higher.

BSDCP

Base station-controlled dynamic clustering protocol (BSDCP) [10] is a centralized clustering protocol that utilizes a CH-to-CH multi-hop routing scheme for inter-clustering communication to reduce energy cost. In BSDCP, the operation includes two phases: setup phase and data communication phase. The current energy status of each sensor node will be sent to the BS in the beginning of setup phase, and then the average energy level is calculated by the BS. Two sensor nodes which have higher energy level and at maximum separation distance will be chosen to split the network into two clusters, and other sensor nodes are assigned to the closest one. This cluster splitting process will continue to perform in the newly split clusters until the required cluster formation is satisfied.

The scheduling for data communication phase is TDMA-based. The inter-cluster CH-to-CH multi-hop routing paths are generated by the BS using the minimum spanning tree approach. This inter-cluster multi-hop routing scheme reduces the energy cost for data transmission significantly which results in better energy efficiency compared to LEACH.

FCM

Fuzzy C-Means (FCM) [35] clustering protocol is a centralized cluster based approach using fuzzy c-means algorithms. In FCM, instead of partition a sensor node to one cluster

only, sensor nodes are grouped into clusters with a degree of belonging to each cluster. Hence, sensor nodes close to the boundary of a cluster may belong to several clusters. The degree of belonging of each sensor node will be calculated by the BS continuously until the convergence is achieved (reach a threshold or maximum number of iterations). Sensor nodes will be assigned to the clusters with highest degree of belonging, and the nearest sensor nodes to the cluster centers will become CHs. Then the results of cluster formation are sent to the network by the BS. A TDMA-based scheduling is used for data transmission in FCM.

The transmission distance between sensor nodes and the distance from sensor node to cluster center are reduced by creating optimal clusters using fuzzy c-means algorithms. Hence power consumption is reduced and life time of the network is extended compared to LEACH. However, all fuzzy c-means approach suffers the same issue: the initial values which are set manually affect the result of clustering.

Chapter 3

Concept of the Hierarchical Agglomerative Clustering (HAC) Algorithm and its Application to WSNs

Before Hierarchical Agglomerative Clustering (HAC) has been applied to WSNs for the first time by Lung et al [13], HAC is already a well known clustering algorithm that has been used in many areas. In addition, HAC is also the fundamental algorithm used in this thesis for the improved clustering protocol. Hence it is necessary to introduce HAC first. In this chapter, the concept of HAC will be described and an example of applying HAC in WSNs will be illustrated.

3.1 The Basic Concepts of HAC Algorithm

In this information explosion age, people are overwhelmed by a huge amount of data. To analyze these data and make them more understandable, classification of data is necessary. Organizing a huge amount of data into groups can lead to discovering the patterns of these data which provide us a more comprehensive understanding on them. Clustering is one of the data classifying technique.

HAC is a family of algorithms that is a subdivision of hierarchical clustering

techniques based on agglomerative method. In HAC, data are not classified into certain number of groups at a single step. On the contrary, it consists of several steps, which can run from n clusters each contains one data object to one final cluster that contains all data objects [36]. In general, the original raw data will be processed first and then put into a resemblance matrix for comparison. The resemblance matrix will show the dissimilarity or similarity (depending on the way of processing data) relationship between data objects. Then the data objects are classified into clusters according to the resemblance matrix, and these clusters will become larger through the multistep operation. A dendrogram can be created along with this multistep operation which can provide a visualization of the relationship among these data objects, the visualized relationship will be more distinct if hierarchical relationships exist [37]. Since HAC is an uncomplicated and efficient data analysis algorithm, it has been applied to many fields.

3.2 Operation of HAC Algorithm

There are three essential procedures while operating HAC algorithm, which are: obtain input data set, calculate resemblance coefficients and execute HAC algorithm [14]. During the process, various types of data, equations and methods can be applied. The input data can be either quantitative or qualitative, and different types of data need to be processed by different kinds of equation to calculate resemblance coefficients. Further, there are several methods for clustering.

In the following passage, each procedure is explained in more details

3.2.1 Input Data Set

In HAC, the input data set is presented as a matrix consisting of data objects and their attributes. Data objects are the entities to be analyzed, normally they have certain similarities which can be classified into clusters. Attributes are the properties of data objects. In the area of WSNs, attributes can be the location of sensor nodes, the connectivity information and the residual energy of sensor nodes.

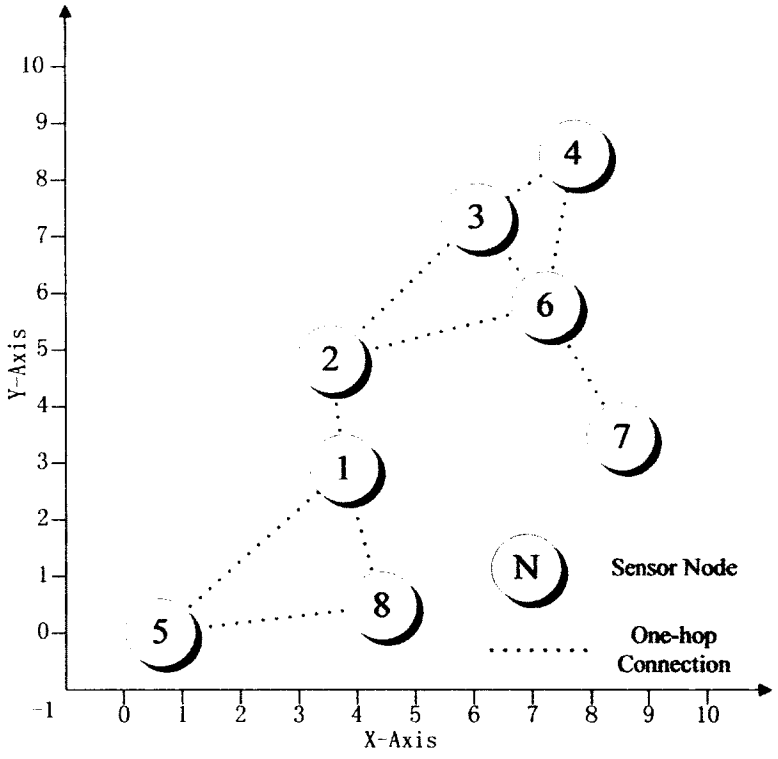


Figure 3-1 A 8-node network topology [14]

A network with 8 sensor nodes deployed in a 10×10 m^2 field is shown in Figure 3-1. This topology has been firstly used as an example in [14]. The input data set of this

network is presented at Table 3-1.

Table 3-1 *Input data set of 8-node network*

(a) Qualitative connectivity data matrix of 8-node network

Object (Node) \ Attribute (Connectivity)	①	②	③	④	⑤	⑥	⑦	⑧
①	1	1	0	0	1	0	0	1
②	1	1	1	0	0	1	0	0
③	0	1	1	1	0	1	0	0
④	0	0	1	1	0	1	0	0
⑤	1	0	0	0	1	0	0	1
⑥	0	1	1	1	0	1	1	0
⑦	0	0	0	0	1	1	1	0
⑧	1	0	0	0	1	0	0	1

(b) Quantitative location data matrix of 8-node network

Object (Node)	Attribute(Location)	
	x-axis	y-axis
①	3.78	2.9
②	3.56	4.83
③	6.06	7.34
④	7.71	8.46
⑤	0.63	0.01
⑥	7.23	5.78
⑦	8.52	3.46
⑧	4.43	0.48

As it shows in Figure 3-1, the one-hop connection between each pair of sensor nodes are represented by dot line. Table 3-1 (a) is the binary qualitative connectivity data

matrix of the 8-node network. Qualitative information is one option of input data to be used for the HAC algorithm. In the connectivity matrix, the value will be either 0 or 1. The value '1' means these two corresponding sensor nodes have one-hop connection and the value '0' means otherwise. Besides, the value '1' also represents the connection relationship between a sensor node and itself. In other words, a sensor node is considered as connected with itself.

Alternatively, quantitative information is another option of input data for the HAC algorithm. Table 3-1 (b) shows the quantitative location data of the 8-node network, in the form of x-y coordinate.

3.2.2 Calculate Resemblance Coefficients

HAC is an algorithm that classifies data objects according to dissimilarity or similarity between them. However, the original input data set do not show the resemblance characteristic directly. Hence these raw input data set need to be processed by some equation so they can turn into resemblance coefficients. For two give data objects, the resemblance coefficient is a measurement of the degree of dissimilarity or similarity.

For the quantitative location data from Table 3-1 (b), resemblance coefficient can be calculated by the following equation which is the Pythagorean Theorem:

$$D_{ab} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \qquad (3\text{-}1)$$

The initial resemblance matrix using input data set from Table 3-1 (b) and

calculated by equation (3-1) is shown in Table 3-2. The dissimilarity coefficients in this matrix are the Euclidean distance between each sensor nodes.

Table 3-2 Initial resemblance matrix with quantitative coefficients

Coefficient (Distance)	①	②	③	④	⑤	⑥	⑦	⑧
①	—	—	—	—	—	—	—	—
②	1.94	—	—	—	—	—	—	—
③	4.99	3.54	—	—	—	—	—	—
④	6.81	5.51	1.99	—	—	—	—	—
⑤	4.27	5.64	9.12	11	—	—	—	—
⑥	4.49	3.79	1.95	2.72	8.77	—	—	—
⑦	4.77	5.15	4.59	5.07	8.61	2.65	—	—
⑧	2.51	4.44	7.05	8.63	3.83	5.99	5.06	—

As for the qualitative connectivity data from Table 3-1 (a), there are many different methods to calculate the resemblance coefficients. In the survey of Choi et al [38], 76 methods have been mentioned. Three typical methods will be introduced in this thesis [15]:

- JACCARD Coefficient:

$$C_{ab} = \frac{N_{1-1}}{N_{1-1} + N_{1-0} + N_{0-1}} \tag{3-2}$$

- Simple Matching Coefficient:

$$C_{ab} = \frac{N_{1-1} + N_{0-0}}{N_{1-1} + N_{1-0} + N_{0-1} + N_{0-0}} \tag{3-3}$$

- SORENSON Coefficient:

$$C_{ab} = \frac{2N_{1-1}}{2N_{1-1} + N_{1-0} + N_{0-1}} \tag{3-4}$$

Which N_{1-1}, N_{1-0}, N_{0-1}, N_{0-0} represent the number of counts of 1-1, 1-0, 0-1, and 0-0 matches of attribute pair between two data objects a and b in a qualitative data matrix like Table 3-1 (a).

Table 3-3 Initial resemblance matrix with qualitative coefficients using SORENSON method

Coefficient (SORENSON)	①	②	③	④	⑤	⑥	⑦	⑧
①	—	—	—	—	—	—	—	—
②	0.5	—	—	—	—	—	—	—
③	0.75	0.25	—	—	—	—	—	—
④	1	0.429	0.143	—	—	—	—	—
⑤	0.143	0.714	1	1	—	—	—	—
⑥	0.778	0.333	0.111	0.25	1	—	—	—
⑦	1	0.667	0.667	0.6	1	0.429	—	—
⑧	0.143	0.714	1	1	0	1	1	—

Table 3-3 shows the result of resemblance coefficient calculation which the input data set is from Table 3-1 (a) and calculated by SORENSON method (equation (3-4)). Note that the qualitative coefficients in Table 3-3 are dissimilarity coefficients while the original SORENSON method in equation (3-4) produces similarity coefficients. In order to remain consistent with dissimilarity quantitative coefficients in Table 3-2, a "1 minus original SORENSON coefficient" process is performed to turn similarity coefficients into dissimilarity coefficients. For example, in Table 3-1 (a), the counts of 1-1, 1-0, 0-1, and

0-0 matches between node-1 and node-3 are $N_{1\text{-}1} = 1$, $N_{1\text{-}0} = 3$, $N_{0\text{-}1} = 3$, $N_{0\text{-}0} = 1$. The similarity coefficient between 1 and 3 calculated by SORENSON method is 0.25, therefore, the dissimilarity coefficient between 1 and 3 equals 0.75 (1 – 0.25).

3.2.3 Hierarchical Agglomerative Clustering Method

After necessary information is gathered by obtaining the initial resemblance matrix, the interested data objects can be started to be classified into clusters by HAC algorithm. The first cluster is formed by the pair of data objects with smallest dissimilarity or highest similarity in the resemblance matrix. Then, since a new cluster appears, resemblance matrix update is needed. There are four standard agglomerative methods for calculating new resemblance coefficient when updating a resemblance matrix [15]:

- Single LINKage (SLINK) method.

  When using SLINK, the similarity measurement between two clusters is defined as the minimum resemblance coefficient among all pair entities of the two clusters.

$$C_{SLINK} = Min(C_1, C_2, \cdots, C_n) \tag{3-5}$$

- Complete LINKage (CLINK) method.

  As opposite to SLINK, CLINK defines the similarity measurement between two clusters as the maximum resemblance coefficient among all pair entities of the two clusters.

$$C_{CLINK} = Max(C_1, C_2, \cdots, C_n) \tag{3-6}$$

❖ Un-weighted Pair-Group Method using the arithmetic Averages (UPGMA).

In UPGMA, the similarity measurement between two clusters is defined as the arithmetic average of resemblance coefficient among all pair entities of the two clusters.

$$C_{UPGMA} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{3-7}$$

❖ Weighted Pair-Group Method using the arithmetic Averages (WPGMA).

As its name, WPGMA is the weighted version of UPGMA.

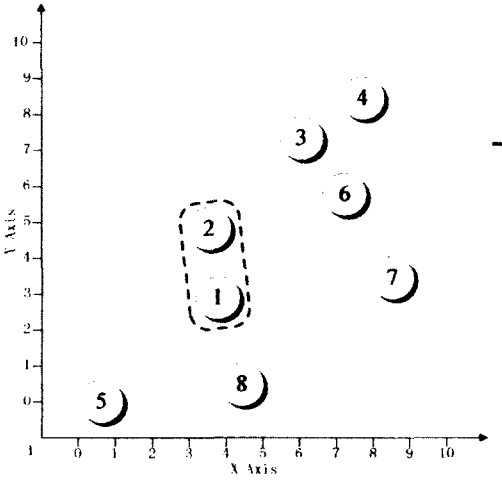
$$C_{WPGMA} = \frac{1}{n}\sum_{i=1}^{n} W_i C_i \tag{3-8}$$

3.2.4 Execute HAC Algorithm: An Example

To have a more intuition elaboration of how HAC algorithm works, an example of HAC execution will be presented as follows. The 8 sensor nodes network topology in Figure 3-1 is the topology that used in this example. The quantitative location data is the input data set of this example, thus in the beginning of executing HAC algorithm, the initial resemblance matrix with quantitative coefficient in Table 3-2 will be checked first. To make this example less complicated and more understandable, SLINK is chosen as the HAC method.

In the initial resemblance matrix in Table 3-2, the smallest dissimilarity coefficient is $D_{12} = 1.94$ which means node-1 and node-2 are the closest pair. Hence they will form a cluster and the resemblance matrix will be updated using SLINK. Figure 3-2

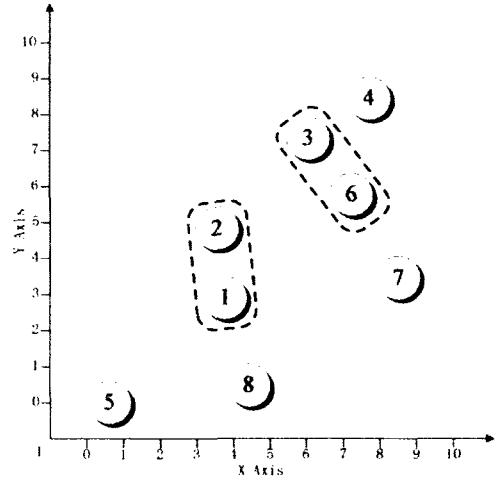
shows the newly formed cluster and the updated resemblance matrix. At this point, there are one cluster which contains 2 sensor nodes and 6 other sensor nodes in the network.



Coefficient (Distance)	③	④	⑤	⑥	⑦	⑧	{①,②}
③	—	—	—	—	—	—	—
④	1.99	—	—	—	—	—	—
⑤	9.12	11	—	—	—	—	—
⑥	***1.95***	2.72	8.77	—	—	—	—
⑦	4.59	5.07	8.61	2.65	—	—	—
⑧	7.05	8.63	3.83	5.99	5.06	—	—
{①,②}	3.54	5.51	4.27	3.79	4.77	2.51	—

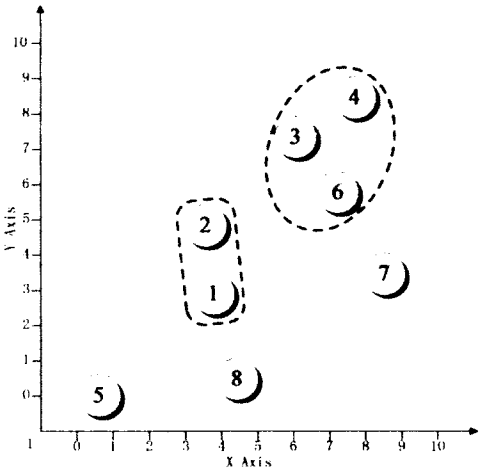
Figure 3-2 Result of clustering and updated resemblance matrix in first step

The same operation will be iterated several times in the following clustering steps. The clustering result and the updated resemblance matrix of each remaining step are shown from Figure 3-3 to Figure 3-6 as following.



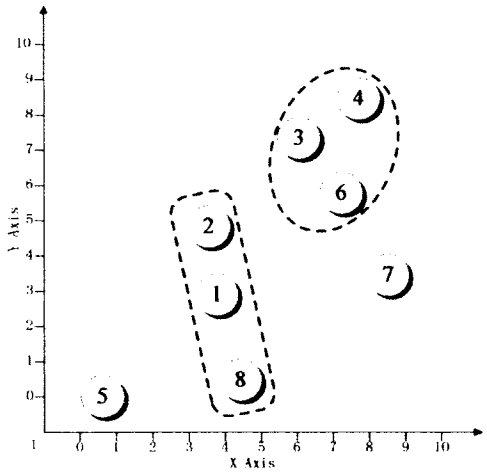
Coefficient (Distance)	④	⑤	⑦	⑧	{①,②}	{③,⑥}
④	—	—	—	—	—	—
⑤	11	—	—	—	—	—
⑦	5.07	8.61	—	—	—	—
⑧	8.63	3.83	5.06	—	—	—
{①,②}	5.51	4.27	4.77	2.51	—	—
{③,⑥}	***1.99***	8.77	2.65	5.99	3.54	—

Figure 3-3 Result of clustering and updated resemblance matrix in second step



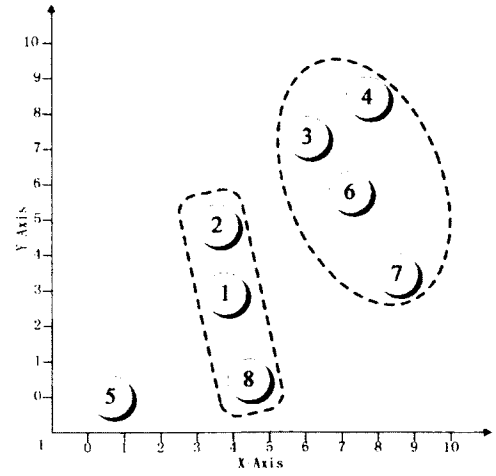
Coefficient (Distance)	⑤	⑦	⑧	{①,②}	{③,④,⑥}
⑤	—	—	—	—	—
⑦	8.61	—	—	—	—
⑧	3.83	5.06	—	—	—
{①,②}	4.27	4.77	*2.51*	—	—
{③,④,⑥}	8.77	2.65	5.99	3.54	—

Figure 3-4 Result of clustering and updated resemblance matrix in third step



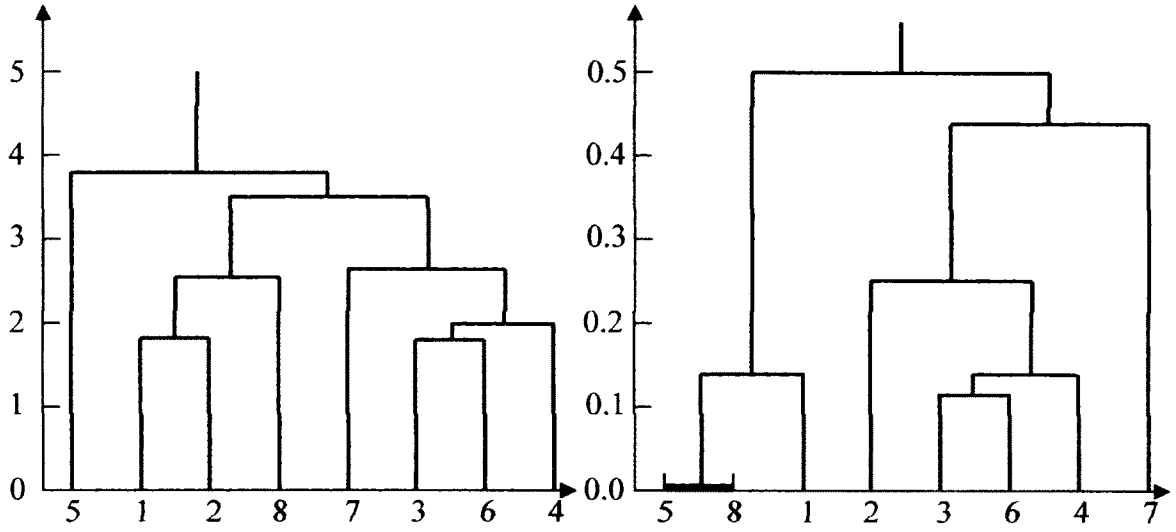
Coefficient (Distance)	⑤	⑦	{①,②,⑧}	{③,④,⑥}
⑤	—	—	—	—
⑦	8.61	—	—	—
{①,②,⑧}	3.83	4.77	—	—
{③,④,⑥}	8.77	*2.65*	3.54	—

Figure 3-5 Result of clustering and updated resemblance matrix in fourth step



Coefficient (Distance)	⑤	{①,②,⑧}	{③,④,⑥,⑦}
⑤	—	—	—
{①,②,⑧}	3.83	—	—
{③,④,⑥,⑦}	8.61	3.54	—

Figure 3-6 Result of clustering and updated resemblance matrix in fifth step



(a) Dendrogram with quantitative location data **(b) Dendrogram with qualitative connectivity data**

Figure 3-7 Dendrogram of HAC algorithm with different kinds of data

A dendrogram which is the visual presentation of the clustering process can be generated in the end. The cluster structure and the hierarchical relationship between clusters can be discovered clearly through dendrogram. Figure 3-7 (a) shows the dendrogram of the example in this section.

The process of clustering of HAC with qualitative data is similar to HAC with quantitative data which is shown in the previous example. However, since the input data set is different, the result of clustering might be different too. Figure 3-7 (b) is the dendrogram with qualitative data which is from Table 3-1 (a), and the HAC method used is also SLINK. As it shows in Figure 3-7 (b), when it compared to Figure 3-7 (a), the clustering process is different and the result is different too. Besides, due to different data type, the value of tree height (the y-axis of Figure 3-7) is different as well.

3.3 Application of HAC in WSNs domain

HAC algorithm is an uncomplicated flexible clustering approach that can be used as a powerful data analyze tool in many area. When it comes to application in WSN, HAC algorithm is totally applicable. The example in previous section shows how HAC algorithm works in WSN domain: by considering sensor nodes as data objects and the location data or connectivity data as attributes. However, this example only shows the mathematical part of applying HAC in WSNs. In order to design a clustering protocol based on HAC, there are many more factors to consider. For example, it is necessary to have control messages; information should be exchanged between sensor nodes; and the minimum size of each cluster need to be considered; etc. In [13], Lung et al successfully developed a WSN clustering protocol that applies HAC algorithm, and the simulation result out performed two famous cluster protocols — LEACH and LEACH-C.

Chapter 4

Hybrid Distributed Hierarchical Agglomerative Clustering (H-DHAC) Protocol

As introduced in Chapter 3, the HAC algorithm is a feasible method which has been successfully applied in many areas. When it comes to WSNs, the HAC algorithm is found to be applicable as well. However, in order for the HAC algorithm to fit the unique conditions of WSNs, some adaptations are needed.

The HAC algorithm works in a centralized way. In other words, if HAC is applied directly to WSNs, each sensor node needs to have global knowledge. But due to the special environmental characteristic of WSNs, it is not practical for each sensor node to have such global knowledge. Location data are one of the most popular to be used in clustering, which is normally obtained by GPS devices. However, embedding GPS chips into sensor nodes increase the cost and GPS devices are not immune to failure. Some studies try to use RSS or RSSI as distance estimator but RSS and RSSI are unreliable as other studies have indicated (see section 4.2). Connectivity data is an alternative choice of input data for clustering, however, binary connectivity data is not as accurate as

location data.

In this chapter, a Hybrid Distributed HAC (H-DHAC) protocol is proposed in order to address the previously indicated issues of high cost and low reliability. H-DHAC is a distributed protocol; each sensor node only needs to have local knowledge instead of global knowledge. More importantly, with H-DHAC, it is not necessary for every sensor node to have location data. H-DHAC exploits the advantage of using two different kinds of data — location and connectivity data in clustering to reduce the cost of using GPS and improve reliability during eventualities such as GPS failure or using only connectivity data.

4.1 Qualitative and Quantitative Coefficients

Clustering in WSN is a task that assigns sensor nodes to different groups. In this process, clustering coefficients are used to measure similarities or dissimilarities between sensor nodes. Sensor nodes with higher similarities, e.g., being close in proximity, tend to be grouped together.

Clustering coefficients can be divided into two categories: quantitative and qualitative. For example, the location of each sensor node is one kind of quantitative data. The distance between two sensor nodes can be obtained on the basis of the well known Pythagorean Theorem. In this case, the distance is the dissimilarity quantitative coefficient. Normally, closer nodes are more likely to form a cluster. In WSNs, when two

nodes are communicating with each other, less radio communication energy will be consumed if they are closer. Hence it is usually desirable to use distance as the representation of dissimilarity.

Received Signal Strength (RSS) or Received Signal Strength Indicator (RSSI) is another kind of quantitative data. RSS is the radio signal power that is measured at the receiver's end. In an ideal case, if one takes the value of RSS as x axis and the value of real distance as y axis to draw a plan, the outcome is expected to show that they are linearly related. In other words, the larger signal power being detected indicates a closer distance. Some papers, such as LEACH [8] and LACBER [39], have adopted RSS as the distance estimator either fully or partially in their clustering scheme. However, many research efforts have been put in the study of RSS and RSSI and the results show that in practice RSS and RSSI are not reliable distance estimators [40, 41]. The unreliability of RSS and RSSI as distance estimator will be discussed in detail later in section 4.2.

Contrary to quantitative data, wherein data is in the form of numeric values, qualitative data use binary values to describe particular information. In our case, qualitative data represent the connectivity information, which is a binary representation of whether two nodes can communicate with each other. The connectivity between nodes is represented by 1 or 0. 1 means that these two nodes are connected, 0 means otherwise. By having the complete connectivity information of each node in the WSN, one can create an N×N (N is the number of nodes in the network) binary matrix to represent the

connectivity relationship of the network. Afterwards, this binary matrix will be further converted to a similarity matrix with qualitative coefficients that can then be used for clustering. This kind of conversion is done through coefficient calculation as described in section 3.2.2, e.g., equation (3-4).

A similarity qualitative coefficient (which can be easily converted to dissimilarity coefficient and is calculated from binary qualitative data) is a good measurement of difference between two sensor nodes. However, qualitative coefficients are not as accurate as quantitative coefficients in terms of clustering in WSNs. In [14], Zhou showed that DHAC (Distributed HAC) using location quantitative data outperforms DHAC using binary qualitative data. Although qualitative data may not be the most accurate, the experimental results do not reveal a huge difference. Therefore, qualitative connectivity data is an adequate substitute in the absence of quantitative location data.

4.2 Reliability of RSS and RSSI as Distance Estimator

The difference between RSS and RSSI is that RSS is the real value of signal strength measured at the receiver's end, while RSSI is an indicator of RSS and hence relative value. RSS has a unit and the value could be negative, but RSSI's value is always positive and unitless. In other words, although RSS and RSSI are two different representations of signal strength, they are essentially saying the same thing. From now on, RSS(I) will be used to denote both of them to avoid duplication except for those

which studied one of them specifically.

RSS(I) is a metric that has been largely used in distance measurement [42], node localization [43] and cluster head selection in WSNs [44]. There are a number of research studies that focus on exploring the use of RSS(I) in various applications, since measuring signal strength is exploring the nature of wireless network. However, many of these studies of RSS(I) are based on theoretical models and under the assumption that RSS(I) reflects the true distance relationship between two wireless sensor nodes, i.e., there is minimum or no error in RSS(I). In fact, this assumption is not always true in practice. Wireless network environments are unstable and changes over time, which means that RSS(I) is not a reliable distance estimator.

Lymberopoulos et al [45] conducted a study on the characterization of signal strength properties and link asymmetries for the CC2420 IEEE 802.15.4 compliant radio using a monopole antenna, with network being deployed in 3D environments. Naturally, signal strength measurements vary, which is caused by multipath, fading and shadowing of the RF channel, transmitter variability, receiver variability, and antenna orientation. In one of their experiments, receivers are placed at 6.5*ft*, 3.5*ft* and 1.25*ft*. They found that the raw RSSI data can not indicate any distance information when the receiver is at a height of 1.25*ft* and 3.5*ft*, because the RSSI value is sometimes the same even if their heights have significant distance. After several experiments, they came to the conclusion that it is not possible to do a direct distance prediction from raw RSSI data in a 3D indoor

environment. Signal strength localization only works in special deployed environments, but this becomes really difficult in other environments and 3D deployments.

Apart from RSS(I) which cannot be directly used for distance estimation, there are studies that try to find the error pattern of RSS(I). Rong-Hou et al [46] analyze the variance of RSSI signals on time domain and frequency domain with a survey on space and time. Their goal is to determine how RSSI varies. Although they try to maintain a stable and unchanging environment, the result of their analysis gives the following conclusions: sampling time has no relationship with RSSI values; RSSI values still vary drastically when the environment is an open area; there is no pattern to be found under the time domain and frequency domain; signal strength and how it varies do not affect each other but both are affected individually by the environment's complexity.

In order to find whether RSSI is a reliable parameter when it comes to sensor localization, Parameswaran et al [41] conducted some practical experiments to prove or disprove the usefulness of using RSSI to estimate the distance between sensor nodes. In the experiment, a nearly ideal environment is carefully maintained: flat surface, no obstacle between sensor nodes, sensor nodes working in perfect condition, etc. The authors of this paper came to the conclusion that RSSI cannot be used as a reliable metric in localization after performing several repeat experiments in a nearly ideal environment. Because RSSI behaves inconsistently even in an ideal environment, it could only be worse when sensor nodes are deployed in more realistic unstable environments. And the

error of measured RSSI values increase with distance.

Several comprehensive experiments using three different platforms with different topologies and material have been conducted by Heurtefeux et al [40]. It is reported in their survey that three main factors (path-loss, fading, and shadowing) affect signal strength [47], which in turn cause the intrinsic limits of using RSSI for distance estimation. The Spring-Relaxation algorithm, a force-based localization algorithm, is investigated on whether it can smooth the distance-estimation errors of RSSI. Their simulation results show that the performance of localization estimation is poor even in very good network conditions. RSSI is thus concluded to be an ineffective candidate for estimating distance in WSNs according to their study.

After seeing above the various conclusions regarding RSS(I) as a distance estimator, it is evident that RSS(I) is not a reliable parameter in distance estimation. It has to be optimized through a very complicated mathematical model in order to minimize the error into an acceptable range. Even if the error of RSS(I) become acceptable after all these complications, the complicated calculation is not desirable for a sensor node with limited processing ability and power.

4.3 GPS Availability and Vulnerability

The location of sensor nodes is a parameter that has been commonly used in studies on clustering and routing for WSN. For instance, in LEACH-C [8], the BS is assumed to

have the location information of every sensor node. In many cases, these studies assume that the sensor nodes are aware of their locations, but how to obtain this information has not been considered.

The location of a sensor may be acquired by using anchor nodes [48] (some nodes know their location, while other nodes will get their approximate position by complex calculations with several iterations) or RSS(I) [49] for estimation (which has just been proved unreliable). Both methods are normally involved with complex mathematical models, and neglect power consumption as well as the computational limitation of sensors during the localization process. It is not a reasonable scenario if a sensor node consumes a lot of energy in localization before it actually starts to sense something. Or the location information has been configured beforehand such that sensors are manually deployed (which only applies to limited scenarios). There are many cases where sensors are dropped by an airplane, making it impossible to know its position before any localization procedure.

There is a low complexity and an accurate way for localization: Global Positioning System (GPS). After decades of evolving and improving, GPS technology has grown mature and easy to get. Today a common cell phone is most likely integrated with GPS technology. The cost of a GPS chip has gone down drastically from $100 to $15,000 ten years ago [50] to $5 [51] to a few thousands [52] now. However, compared to wireless sensors, the price of a sensor ranges from less than $1 [1] to a few thousands

[53]. The cost of a sensor could be doubled or higher if it is equipped with a GPS chip.

Although having a GPS chip integrated with sensors will result in a higher cost, a WSN cannot gather useful data without knowing any location information. In other words, at least a fraction of sensor nodes must be equipped with GPS; otherwise, the gathered data will not be helpful even those sensor nodes can work and communicate with each other. The reasons may be described as follows: if the sensors are deployed randomly by a helicopter in a forest to detect forest fire [54], the location of the sensors are essential since a potential fire cannot be located without location information. For military sensors [55], in the case where human activity in the monitoring area is the focus, using location-aware sensors is necessary. There are some applications that may not be sensitive to very accurate location information, such as detecting animal activity [56] or the degree of soil moisture [57]. In terms of WSN clustering, some sensors with GPS being equipped inside of a cluster will be adequate for determining the approximate location if one sensor (without GPS) of that cluster sense something interesting. In case a more accurate location of that specific location-unaware sensor is desired, those sensors that are location-aware can act as anchor node and let the BS do the complicated calculations. Therefore, a distributed energy consuming localization procedure can be avoided and be replaced by centralized localization until some useful information presents.

As for sensors that are integrated with GPS chips, questions remain. Are they

guaranteed to get the location information correctly? Is there any chance the GPS may fail to provide the correct location? The answer is that GPS always faces the possibility of failure, in addition to the possibility of reporting an incorrect location because of interference [58]. In both cases, the cause may be that the satellite signal at the GPS receivers end is very weak. It could even be 10^{18} times weaker than a 100 Watt light bulb [59]. Therefore, the satellite signal can be easily interfered with at the receiver's end, and the receiver could possibly miss the satellite signal or be misled by other signals.

For the purpose of classifying modes and models of GPS failure, Bhatti et al [60] identify potential GPS failure modes in their paper. There is a chart of 23 potential reasons for GPS failure. In spite of those that are related to the environments and the availability of the GPS satellite and solar system, some other causes are closely related to WSNs. A jamming signal is a powerful radio signal that is generated by some radio device that intentionally interferes with the reception by a GPS receiver; it could cause a serious degrading of localization accuracy. Besides, jammers could send a spoofing signal that pretends to be the authentic satellite signal so as to mislead GPS devices, leading them to report wrong locations. The purpose of such intentional interference is to disable GPS devices completely so that the jammer could hide from being tracked or keep some field undiscovered. One possible scenario of jammer usage is auto thieves trying to hide themselves from police in case the in-car GPS expose their positions. Besides intentional interference, some interference could be unintentional. This happens

when a GPS receiver is near some device or infrastructure that generates radio signal which has a similar frequency range to the satellite. VHF, mobile satellite service, ultra wideband radar, and television are some of the possible source of unintentional interference. Another possible failure mode is problems with the GPS receiver. There are some chances that the manufacturer of GPS device did not follow its specific documentation when making them, i.e., certain GPS receivers may not be properly installed or modulated. It is more likely to happen when manufacturers try to save up on cost by producing an inexpensive low-end GPS device; hence, the reliability of GPS is in proportion to the price. Cheaper GPS chips reduce cost but fail to provide reliability; high reliable GPS chips give us a better result in all applications in WSNs (including clustering) at a much higher cost. Thus, there is a tradeoff here between reliability and cost.

4.4 H-DHAC Network Environments and Assumptions

Given that sensor nodes can sometimes be deployed in wild or harsh environments, the network environment of a WSN could be complicated and vary through time. However, researchers and scholars have been working on this field and have proposed many schemes and models to enable simulations of clustering in WSNs.

Nowadays, many applications in WSNs were developed to cater for different needs. Although settings and requirements of different applications are not all the same,

the main constraints and limitations they suffer are similar, e.g., limited energy, low bandwidth and inadequate computation ability. Thus, energy efficiency, load balance, and self-adaptation become vital in the design of WSNs.

LEACH [8] and DHAC [13, 14] share similar settings and assumptions, most of which are commonly used by various WSN clustering methods. These assumptions and settings also apply to H-DHAC, as it is an improvement of DHAC. These are:

- Sensor nodes are stationary after they have been deployed in the sensing area. The sink for collecting data is static.
- Radio transmission channel is symmetric.
- Channel propagation model is distance-related.
- The transmission power of sensor nodes can be adjusted to fit a shorter or longer transmission distance. Every sensor node can communicate with the sink directly.
- The minimum transmission power between each pair of sensor nodes can be calculated or estimated by Transmission Power Control.
- All sensor nodes have homogeneous capabilities. They have the same initial power, data processing and transmission capabilities, computational and transmission cost.
- The radio communication component of each sensor node can be turned on/off.
- Sensor nodes know their GPS availability status, i.e., they are aware of whether a GPS is equipped or functioning properly.
- Sensor nodes only have local information of their neighbors.

4.5 H-DHAC Clustering Protocol

This thesis proposes a clustering protocol that can make use of two different kinds of data: quantitative location data and qualitative connectivity data. Qualitative connectivity data are always available to all sensor nodes as long as the radio communication component of a wireless sensor node is operational (a wireless sensor node is of no use if it cannot communicate with other sensor nodes). However, this statement is not always true for quantitative location data, due to cost saving, obstacles, GPS failures and some other reasons or needs.

H-DHAC clustering protocol is a solution to the problem when the location data are occasionally unavailable for whatever reasons. When location data are not available for some sensor nodes, the H-DHAC protocol can utilize the existing connectivity data for clustering. H-DHAC can function for every percentage of GPS availability (0% — 100%), i.e., it is designed to be adaptive, to work even in extreme cases where none of the sensors has GPS or GPS is available for every sensor, and to work in-between these two extreme cases. Therefore, H-DHAC not only provides a flexible design for those who have any specific demand on cost or performance but also supports a robust capability by considering possible GPS failures.

4.5.1 H-DHAC Procedures

H-DHAC is a static clustering protocol which means that the structure of each cluster will

not change after it is formed. When there is any sensor node that runs out of energy, the other living sensor nodes will continue to work without being affected. For a clustering protocol, clusters are created for organizing a WSN so that the management tasks can be simplified (such as communication) between sensor nodes. There is a Cluster Head (CH) for each cluster, which is the leader in that cluster. The CH's job is to manage the activities of that cluster, including scheduling, data gathering and transmissions to BS. After sending the data, which have been collected from Cluster members (CM), to the BS, a CH's job is completed for this round. Each CM has the potential to become the CH in the future, because CH rotation will be performed in the next round for balancing energy consumption. After cluster formation, a sensor node will keep repeating the data gathering and transmission process until its energy is exhausted.

H-DHAC is a distributed clustering protocol; sensor nodes only have local information of their neighbors. And sensor nodes only communicate with neighbors in the initial stage and with other sensor nodes within the same cluster after the cluster formation process, except when the CH sends gathered data to BS. Specifically, the H-DHAC protocol consists of several stages: (1) Clustering start-up stage; (2) Cluster formation stage; (3) Transmission power control (TPC) stage; (4) Scheduling and the data transmission stage; and (5) Cluster maintenance stage.

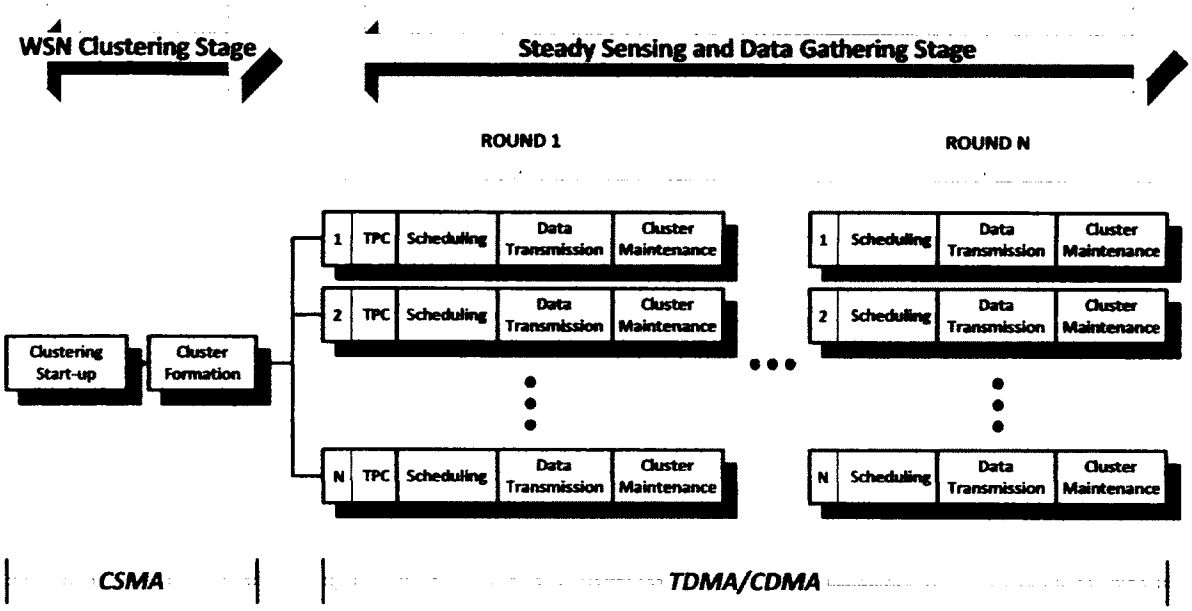


Figure 4-1 H-DHAC procedures

The H-DHAC procedures through time are shown in Figure 4-1. Sensor nodes will firstly use CSMA to send a beacon signal to acknowledge its neighbors in the clustering start-up stage. Then clustering will be performed using local information obtained from neighbors. Before sensor nodes proceed to the sensing and data gathering stage, Transmission Power Control (TPC) is needed for any pair of sensor nodes, both do not have location information to estimate the minimum transmit power required (it is assumed that in simulations, minimum transmit power can be calculated between the pairs that both have location information [61, pages 85-86]). And TPC is performed for the first round only, since H-DHAC is a static clustering protocol. A TDMA schedule for each cluster is created by the CH and sent to CM. The intra-cluster data transmission is based on "TDMA + CDMA". The data collected by CHs is finally transmitted to BS, and CDMA is adopted during this process to avoid inter-cluster interference. The steady

sensing and data gathering stage will continue to operate until all sensor nodes run out of energy.

The following subsections describe each stage in detail.

4.5.2 H-DHAC Clustering Start-up and Cluster Formation Stages

Before sensor nodes start to sense and gather sensed data, they need to be clustered in order to work in an energy-efficient way. Clustering start-up and cluster formation stages are involved in this process.

4.5.2.1 Clustering Start-up Stage

At the beginning of clustering, a sensor node has no knowledge of other nodes because they are deployed randomly. Thus each sensor node will broadcast a HELLO message to inform its neighbors. This HELLO message includes its ID, GPS availability status (Yes or No), and location (if available). Meanwhile, if a sensor node receives HELLO messages from other sensor nodes, it will store the information for further use. Sensor nodes will keep listening until they are done (determined by a pre-configured timeout) exchanging HELLO messages with neighbors.

After exchanging HELLO messages, sensor nodes are aware of their neighbors and able to establish their own neighbor lists. Then a sensor node will exchange its own neighbor list with its neighbors in the same manner as exchanging HELLO messages. It

is necessary to know the neighbors' neighbor list (i.e., neighbors' connectivity information) for building a local binary connectivity matrix. The clustering start-up stage is done when exchanging neighbor lists is complete.

Build Up Local Resemblance Matrix

Both quantitative and qualitative coefficients can be calculated after local location and connectivity information has been gathered during the cluster start-up stage. Table 4-1 shows both the qualitative connectivity data and quantitative location data of node-1 (which is a node as shown in Figure 3-1) and its neighbors, which are stored in node-1's memory after exchanging messages. In Table 4-1 (a), a '1' denotes one-hope connection between two sensor nodes and a '0' means otherwise. As mentioned previously, a sensor node's neighbor list equals its connectivity information with other sensor nodes. Hence, a local connectivity matrix can be easily built by filling in 1s between neighbors and 0s when they are not neighbors. (Besides, sensor nodes are always considered to be connected with each other; hence, the '1' values are for identical nodes.) Table 4-1 (b) shows the local location information (values of x and y axes) of node-1 and its neighbors. The raw data represent the relationship between node-1 and its neighbors. However, further processing for resemblance coefficients is needed to show the similarity or dissimilarity relationship between sensor nodes.

Table 4-1 Local qualitative and quantitative data of Node-1

(a) Local qualitative connectivity data matrix of Node-1

Node \ (Connectivity) Node	①	②	③	④	⑤	⑥	⑦	⑧
①	1	1	0	0	1	0	0	1
②	1	1	1	0	0	1	0	0
⑤	1	0	0	0	1	0	0	1
⑧	1	0	0	0	1	0	0	1

(b) Local quantitative location data matrix of Node-1

Neighbor Node	Location	
	x-axis	y-axis
①	3.78	2.9
②	3.56	4.83
⑤	0.63	0.01
⑧	4.43	0.48

Node-1's initial resemblance matrix for the first step of cluster formation is presented in Table 4-2. In this table, quantitative coefficients are the Euclidean distance calculated with the Pythagorean Theorem, while qualitative coefficients are computed using the SORENSON method (see equation (3-4)), both of which are dissimilarity coefficients. Note that the original SORENSON coefficient represents similarity. In order to be consistent with the quantitative coefficient, the qualitative coefficient is handled by 1 minus the original similarity coefficient.

The matrix in Table 4-2 is a complete resemblance matrix: there is no missing information for node-1 in the quantitative coefficient section (as discussed previously, qualitative information is always available). This is because neither node-1 itself nor any of its neighbors are experiencing GPS unavailability problem. Hence, node-1 can use quantitative coefficients in the first step for cluster formation. In other cases where the quantitative coefficient of a node or a cluster is unavailable or incomplete, qualitative coefficient will be used for clustering.

Table 4-2 Initial resemblance matrix of Node-1 with quantitative and qualitative coefficients

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
②	1.94	0.5
⑤	4.27	0.143
⑧	2.51	0.143

4.5.2.2 Cluster Formation Stage

When initial resemblance matrices of each sensor node in the network have been built, the process of forming clusters will begin. The detailed H-DHAC cluster formation steps are presented in Figure 4-2. Cluster formation in H-DHAC includes several iterations, and this process will be completed when the desired number of clusters is reached and the

cluster size is satisfied. It has been examined in LEACH that the optimal number of CH to be recommended is 5% of the total number of sensor nodes [61, pages 90-92]. Since our approach is static clustering, the number of clusters can be controlled. Also each cluster will have only one CH at a time, so that the optimal number of clusters will also be 5% of the total number of sensor nodes.

As shown in Figure 4-2, sensor nodes will gather the necessary information with neighbors and build the local resemblance matrices with two kinds of coefficients at the beginning of clustering before the cluster formation stage starts. First, each sensor node will form a singleton cluster (e.g., only the sensor node itself in the cluster) before becoming the CH of its cluster. After that, they will find their closest neighbor using the best coefficient available; and here quantitative coefficient is desired, but qualitative coefficient will also be used if the previous is not complete. H-DHAC is designed so that in a normal scenario a CH with a smaller ID (compared to its closest neighbor) will be the one that initiate the cluster formation process and sends the INVITATION message. However, when a cluster's closest neighbor has complete quantitative coefficients but it does not, then the CH with a larger ID will be the one that sends the INVITATION message. In this way, a potential error that could be caused by using two different types of coefficients can be avoided by opting in favor of quantitative coefficients.

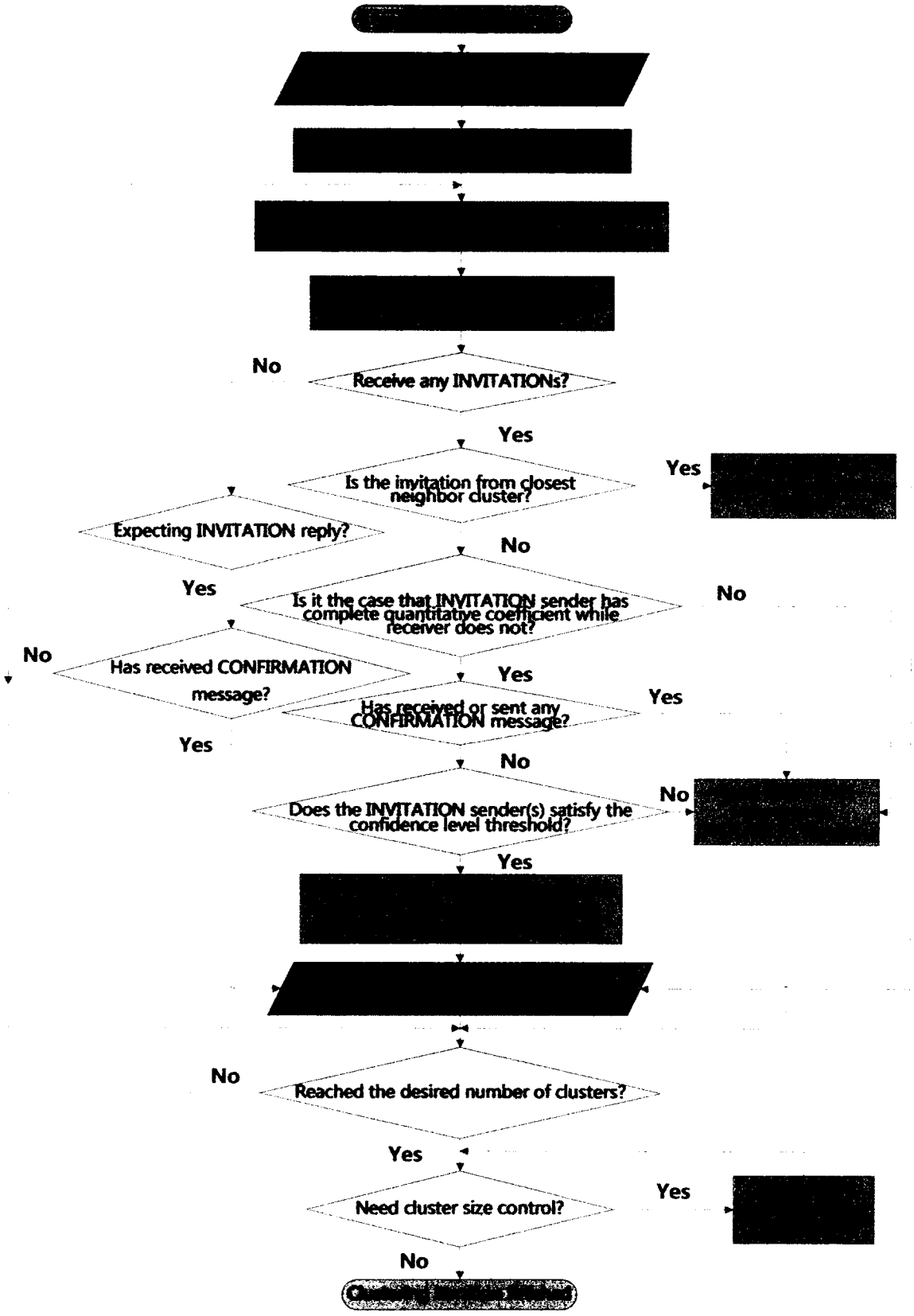


Figure 4-2 Flow-chart of H-DHAC cluster formation steps

If a CH receives an INVITATION message, it will check whether the invitation comes from its closest neighbor cluster. In general, invitations will be rejected if they are not from the closest neighbor cluster. REJECTION messages will be sent by the invitation receiver, and if a CH receives a REJECTION message, no more invitation attempt will be made in this round. However, a different approach is used when dealing with special cases. In the case where the invitation sender has complete quantitative coefficients, while the invitation receiver does not (this situation will henceforth be referred to as "CONFIDENCE INVITATION"), the receiver will keep sender's information (including the sender's confidence level) for now. Confidence level (CL) is calculated by:

$$CL = \frac{N_G}{N_C + N_N} \times 100\% \tag{4-1}$$

In the equation(4-1), N_C is the total number of sensor nodes of its own cluster; N_N is the total number of sensor nodes of all other neighbor clusters; N_G is the number of sensor nodes which have GPS (location data) available in N_C and N_N. Confidence level is used for determining the percentage of sensor nodes in a certain cluster, with GPS (location data) available. With this information, we can choose a cluster which has more potential to fill in the blanks of quantitative coefficients in a resemblance matrix. These blanks come about due to a lack of GPS information in some sensor nodes. How to fill in the blanks in resemblance matrix will be discussed later.

During the process of sending invitations, an invitation receiver will reply with a CONFIRMATION message upon receiving its closest neighbor's invitation. Then these two clusters will form a new cluster, and the CH of this new cluster will broadcast an INFORM message to its neighbors. Subsequently, neighbors' resemblance matrices are updated. After all invitation messages have been sent in this round, special handling will be made for those CHs which received "CONFIDENCE INVITATION". As has been explained in the last paragraph, these CHs will temporarily store the information from senders of "CONFIDENCE INVITATION". If they have made any confirmation (i.e., they have formed a new cluster) with other CHs already in this round, REJECTION messages are sent to these "CONFIDENCE INVITATION" senders. However, if somehow these CHs did not receive the INVITATION or receive a REJECTION message from its closest neighbor, they will send confirmation messages to "CONFIDENCE INVITATION" senders if and when these "CONFIDENCE INVITATION" senders satisfy the confidence level threshold, e.g., 70%, otherwise REJECTION messages will be sent. In the case where there is more than one "CONFIDENCE INVITATION" sender for one CH (all of them pass the confidence level threshold), the sender with the highest confidence level is chosen. Similar to the normal situation, a new cluster is formed and an INFORM message is broadcasted by CH of the new cluster to update the neighbors' resemblance matrices. At last, if there are some CHs that did not receive any invitation and/or did not receive confirmation from

invitation receiver, they will stop performing any action in this round and wait for the next.

In addition to the flow chart, the steps for cluster formation of one round are listed in following in order to give a comprehensive description:

1. Each CH finds its closest neighbor using the best coefficient available (quantitative coefficients is preferred, but qualitative coefficients will be used if quantitative coefficients is not complete).
2. The CH with a smaller ID sends an INVITATION message to its closest neighbor.
3. In the case when a CH does not have complete quantitative coefficients but its closest neighbor does, the CH with larger ID will be the one that sends the INVITATION message.
4. Invitations will be rejected if it is not from the closest neighbor cluster. If a CH receives REJECTION message, it will stop making invitation attempt for this round.
5. In the case when invitation sender has complete quantitative coefficients while the invitation receiver does not ("CONFIDENCE INVITATION"), receiver will keep sender's information (including sender's (CL)) for now.
6. An invitation receiver will reply with CONFIRMATION message if the INVITATION message is from its closest neighbor. These two clusters will be merged into a new cluster, and an INFORM message will be broadcasted to the new cluster's neighbors to update their resemblance matrices.

7. If a CH receives "CONFIDENCE INVITATION" and did not receive INVITATION or receive REJECTION message from its closest neighbor, the CH will confirm with the "CONFIDENCE INVITATION" sender if the "CONFIDENCE INVITATION" sender satisfies the CL threshold (e.g. 70%); otherwise a REJECTION message will be sent.
8. If there is more than one eligible "CONFIDENCE INVITATION" sender for one CH, the sender with the highest CL is chosen.
9. In step 7 and 8, if there is any confirmation has been made, two clusters which are the CONFIRMATION message sender and receiver will be merged into a new cluster, and an INFORM message will be broadcasted to the new cluster's neighbors to update their resemblance matrices similar to step 6.

If at the end of this round of clustering, the desired number of clusters (5% of the total number of sensor nodes) has not been reached yet, a new round will start. It will start from CHs to find its closest neighbor from their resemblance matrices. Otherwise, CHs will proceed to check whether the cluster size control is needed for its own cluster. The size of a cluster could affect the performance of WSNs [62], hence it is desirable to have a certain cluster size. For H-DHAC, the cluster size control should have it bigger than a certain threshold; for example, each cluster should have at least 10% of the total number of sensor nodes. If a cluster has a smaller size than the threshold, it will be merged with other cluster(s). In the process of cluster size control, a cluster will merge

with its closest neighbor cluster. As usual, the CH of the new cluster will send an INFORM message to its neighbors in order to update their resemblance matrices. At last, when the desired number of clusters is reached and no more cluster size control is needed, cluster formation is thus completed. Since H-DHAC is a stationary clustering protocol, no more clustering will be performed beyond this point.

Update Resemblance Matrix in H-DHAC

As has been described at the beginning of this chapter, H-DHAC uses hybrid coefficient techniques which include both quantitative and qualitative coefficients. When there are certain changes in the neighbors' information (e.g., a new cluster is formed), the resemblance matrix of that cluster should be updated. There are four typical methods: SLINK, CLINK, UPGMA and WPGMA (see section 3.2.3). Since qualitative coefficient is always available, updating resemblance matrices in this part will be the same as HAC. On the contrary, because GPS might not be available to some sensor nodes, there might be some blanks for quantitative coefficients. To deal with this problem, special update methods are designed in H-DHAC. The following passage explains the methods for SLINK/CLINK and UPGMA/WPGMA.

SLINK and CLINK

SLINK and CLINK follow similar procedures; the only difference between them is that

SLINK chooses the minimum coefficient while CLINK opts for the maximum coefficient among all pairs of entities for updates. Hence one example for both of them will suffice. This thesis provides an example of using SLINK to update both quantitative and qualitative parts of a resemblance matrix (see Table 4-3). Cluster {4} is a singleton cluster (Figure 3-1), while clusters {3} and {6} are neighbor clusters of cluster {4} as shown in Table 4-3 (a), before any changes happen. Afterwards, cluster {3} and {6} form a new cluster, so that the resemblance matrix of cluster {4} needs to be updated. For the qualitative coefficient, 0.143 is chosen as the new coefficient according to SLINK because it is the smallest value. But since node-3's location information is missing, there is a blank in the quantitative coefficient part of cluster {3}. To prevent this blank from appearing in the updated matrix, cluster {6}'s quantitative coefficient 2.92 is chosen as the new coefficient. Table 4-3 (b) lays out the resemblance matrix of cluster {4} after having been updated. After that, the resemblance matrix can subsequently be updated in a normal fashion since it is complete. Note here that the blank cannot be filled if C_{MIN} equals 2, because there is only one coefficient available. C_{MIN} is the minimum number of coefficients required in two merging clusters for coefficient estimation.

UPGMA and WPGMA

UPGMA and WPGMA methods are conceptually similar. The difference between them is that UPGMA is an un-weighted arithmetic average method, while WPGMA is a weighted

arithmetic average method. To avoid repetition, only the special processing for the UPGMA method will be illustrated in this section. In Table 4-4, node-3 and node-6 are sensor nodes from Figure 3-1, and they form cluster {3, 6}. In this example, node-3 has no location information, which is why there are only blanks for all the quantitative coefficients in the resemblance matrix of cluster {3, 6}.

Table 4-3 Updating the resemblance matrix of cluster {4} using SLINK

(a) Resemblance matrix of cluster {4} before coefficient update

Cluster {④}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{③}	-	*0.143*
{⑥}	*2.92*	0.25

(b) Resemblance matrix of cluster {4} after coefficient update

Cluster {④}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{③,⑥}	2.92 -*	0.143

*When C_{MIN}=2

Table 4-4 Updating the resemblance matrix of cluster {3, 6} using UPGMA

(a) Resemblance matrix of cluster {3, 6} before coefficient update

Cluster {③,⑥}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{②}	-	0.2915
{④}	-	0.1965
{⑦}	-	0.548
Member ⑥	-	0.111

(b) Resemblance matrix of cluster {3, 6} after coefficient update, C_{MIN}=2

Cluster {③,⑥}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{④}	-	0.1965
{⑦}	-	0.548
{①,②}	4.14 -*	0.52775
Member ⑥	-	0.111

*When C_{MIN}=3

There is a basic rule for using UPGMA or WPGMA in H-DHAC to estimate

missing coefficients and to fill in the blanks in a matrix: there must be at least three sensor nodes that have GPS location information available in two merging clusters, and at least one in each cluster. This thesis uses D_{AB} to denote the quantitative distance coefficient of cluster (node) A and cluster (node) B. Thus generally, if all information is complete, the quantitative coefficient of cluster {1} and cluster {3, 6} should be calculated using UPGMA as follows:

$$D_{\{3,6\}\{1\}} = \frac{1}{2}\left(D_{13} + D_{16}\right) \tag{4-2}$$

But due to the absence of node-3's location data, coefficient D_{13} does not exist. Therefore, the coefficient $D_{\{3,6\}\{1\}}$ is a blank in the matrix in Table 4-4 (a), since it does not satisfy the basic rule of "at least three sensor nodes needing to have GPS location information". The same reason applies to blanks for coefficients $D_{\{3,6\}\{2\}}$, $D_{\{3,6\}\{4\}}$ and $D_{\{3,6\}\{7\}}$.

In addition to the basic rule, the control parameter C_{MIN} is used to control the accuracy of coefficient estimation. And this parameter can be applied to all four methods: SLINK, CLINK, UPGMA, and WPGMA. As introduced in the previous subsection, C_{MIN} is the threshold for coefficient estimation. In Table 4-4 (b), cluster {2} forms a new cluster with cluster {1}; therefore, the resemblance coefficients update of cluster {3, 6} is performed. As shown in Table 4-4 (b), C_{MIN} is set to 2. Under this condition, the coefficient $D_{\{3,6\}\{1,2\}}$ can be estimated with two coefficients using UPGMA:

$$D_{\{3,6\}\{1,2\}} \cong \frac{1}{2}\left(D_{16} + D_{26}\right) \tag{4-3}$$

Consequently, the estimated value 4.14 fills the blank of quantitative coefficient $D_{\{3,6\}\{1,2\}}$

in the resemblance matrix of Table 4-4 (b). However, if we set C_{MIN} to 3, the cell for coefficient $D_{\{3, 6\}\{1, 2\}}$ will still be blank (the blank may be updated in the upcoming round), as there are less than three coefficients available. For the qualitative coefficient in Table 4-4 (b), they are updated following the normal way of UPGMA as in HAC.

Cluster Formation Stage of H-DHAC: An Illustration

The main idea of H-DHAC cluster formation is shown in Figure 4-2, while the detail of cluster formation has been explained in the previous section. Now since the special process of updating resemblance coefficient in H-DHAC has been described, this section provides an example. The network topology for this example is as depicted in Figure 3-1, which is a simple network with 8 sensor nodes. In this example, assume that node-3 is the node without location data, while all other seven nodes have complete quantitative and qualitative data, and SLINK is chosen as the matrix update method.

As illustrated in Table 4-5, sensor nodes have exchanged necessary local information, built resemblance matrix and started as singleton clusters themselves, e.g., clusters with only one member. The quantitative coefficient is calculated using the Pythagorean Theorem, while qualitative coefficient is calculated with the SORENSON dissimilarity method. In these resemblance matrices, CHs are highlighted with bold and underline (e.g., **①**). Also the coefficients of a cluster's closest neighbors are highlighted in bold and italics (e.g., ***1.94***). Since H-DHAC is a distributed clustering protocol,

clusters in Table 4-5 only have local information of their neighbors. Some matrices have blanks in their quantitative coefficients caused by the missing location data of node-3.

Table 4-5 Initial local resemblance matrices with quantitative and qualitative coefficients

Cluster {①}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{②}	***1.94***	0.5
{⑤}	4.27	0.143
{⑧}	2.51	0.143

Cluster {②}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{①}	1.94	0.5
{③}	-	0.25
{⑥}	3.79	***0.143***

Cluster {③}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{②}	-	0.25
{④}	-	0.143
{⑥}	-	***0.111***

Cluster {④}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{③}	-	***0.143***
{⑥}	2.72	0.25

Cluster {⑤}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{①}	4.27	0.143
{⑧}	***3.83***	0

Cluster {⑥}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{②}	3.79	0.333
{③}	-	***0.111***
{④}	2.72	0.25
{⑦}	2.65	0.429

Cluster {⑦}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{⑥}	***2.65***	0.429

Cluster {⑧}

Neighbor Cluster	Coefficient	
	Quantitative	Qualitative
{①}	***2.51***	0.143
{⑤}	3.83	0

At the beginning, each cluster finds its closest neighbor using the best coefficient available. Clusters {1}, {5}, {7} and {8} will choose a quantitative coefficient since their quantitative coefficient are complete, while clusters {2}, {3}, {4} and {6} will have to use qualitative coefficients as a result of incomplete quantitative coefficients. Then, clusters {1}, {2}, {3} and {5} will send an INVITATION message to clusters {2}, {6},

{6} and {8}, respectively, because they have a smaller ID compared to their closest neighbors. However, cluster {7}, which has a larger ID compared to its closest neighbor {6}, will be the one to initiate the invitation and send an INVITATION message to {6}. Cluster {7} is an eligible invitation sender because it has complete quantitative coefficients, while its closest neighbor {6} does not.

When CHs receive invitations, they will normally reply depending on different scenarios. Cluster {8} will send a REJECTION message to cluster {5}, as clusters {5} and {8} are not the closest neighbors according to cluster {8}'s matrix. The same rule applies to cluster {3} which will send a REJECTION message to cluster {2}. Although cluster {1} is not the closest neighbor according to cluster {2}'s matrix, cluster {2} will temporarily stories cluster {1}'s invitation information (including cluster {1}'s confidence level which is 100%). Cluster {2} will respond later since cluster {1}'s invitation is a "CONFIDENCE INVITATION". Following the same rule as cluster {2}, cluster {6} will do the same thing to cluster {7}. But then cluster {6} will discover that the invitation from cluster {3} is an invitation from the closest neighbor, so that cluster {6} sends a CONFIRMATION message to cluster {3} and a REJECTION message to cluster {7}. Meanwhile, cluster {2} receives a REJECTION message from cluster {3}, and thus responds to cluster {1} with a CONFIRMATION message. In this step, clusters {1} and {2} as well as clusters {3} and {6} form two new clusters separately. And the newly formed clusters will inform their neighbors to update their resemblance matrices.

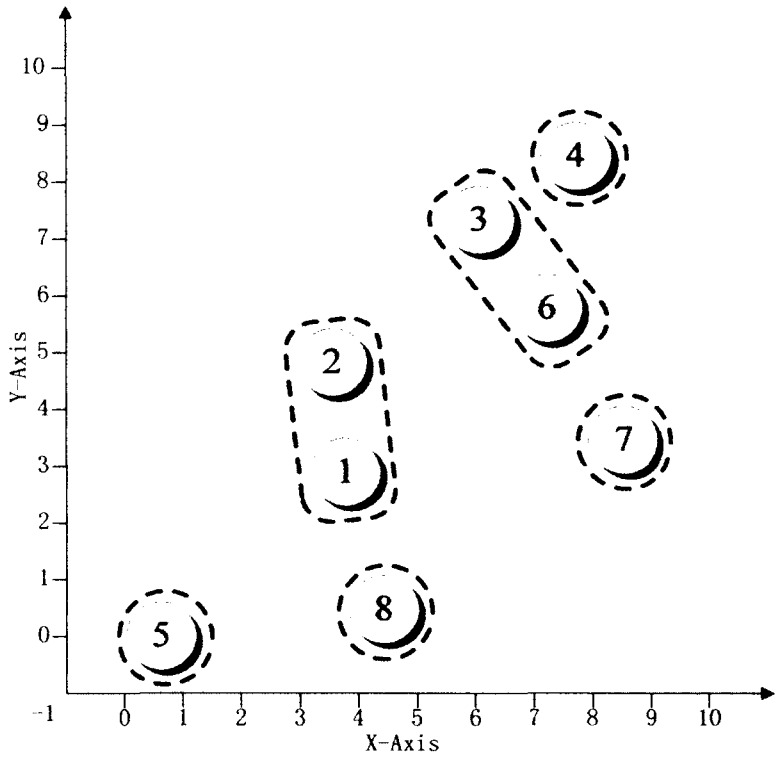


Figure 4-3 H-DHAC: Results of cluster formation in first step

Table 4-6 H-DHAC: *Updated resemblance matrices using* SLINK *with quantitative and qualitative coefficients in the first step*

Cluster {①,②}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{⑤}	4.27	0.143
{⑧}	***2.51***	0.143
{③,⑥}	3.79	0.25
Member ②	1.94	0.5

Cluster {③,⑥}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{④}	2.72	0.143
{⑦}	***2.65***	0.429
{①,②}	3.79	0.25
Member ⑥	-	0.111

Cluster {④}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{③,⑥}	2.72	0.143

Cluster {⑤}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{⑧}	***3.83***	0
{①,②}	4.27	0.143

Cluster {⑦}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{③,⑥}	***2.65***	0.429

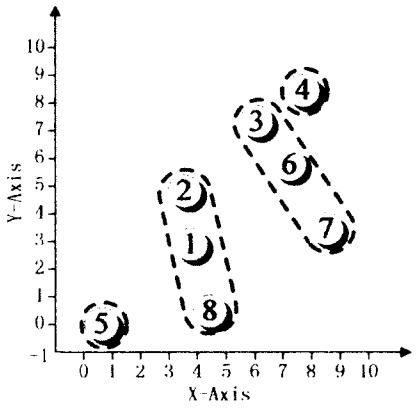
Cluster {⑧}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{⑤}	3.83	0
{①,②}	***2.51***	0.143

The result of H-DHAC cluster formation in the first step is shown in Figure 4-3.

Two new clusters have been formed in this step, so that there are now 6 clusters in the network. With the formation of new clusters, the resemblance matrix of each cluster (including those new clusters) needs to be updated. This action takes place when CHs receive the INFORM message from the newly formed cluster. Table 4-6 shows the updated resemblance matrices in the first step, while the SLINK method is used for the normal coefficient update as well as for special processing on coefficient blanks. As we can see, the blanks of initial resemblance matrices are mostly filled by the special processing method using SLINK in H-DHAC, which has been described previously. Those quantitative coefficients that are highlighted with a double underline are the specially processed ones. From now on, every cluster of this network can use their quantitative coefficient for clustering since the quantitative coefficients are complete.

And then, a new round of cluster formation begins. Following the same procedure, each cluster finds its closest neighbor in its matrix. Thereafter, clusters {1, 2}, {3, 6} and {5} send INVITATION messages to clusters {8}, {7} and {8}, respectively. CONFIRMATION messages are sent by clusters {8} and {7} to clusters {1, 2} and {3, 6}, respectively, since they are closest to each other. Also a REJECTION message from cluster {8} to cluster {5} is sent because they are not the closest according to the matrix of cluster {8}. At the end of this round, two new clusters, {1, 2, 8} and {3, 6, 7} have been formed which inform their neighbors. These outcome and updated resemblance matrices in the second step are shown in Figure 4-4 (a) and (b).



(a) Results of cluster formation

Cluster {①,②,⑧}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{⑤}	3.83	0
{③,⑥,⑦}	***3.79***	0.25
Member ②	1.94	0.5
Member ⑧	2.51	0.143

Cluster {③,⑥,⑦}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{④}	***2.72***	0.143
{①,②,⑧}	3.79	0.25
Member ⑥	-	0.111
Member ⑦	2.65	0.429

Cluster {④}

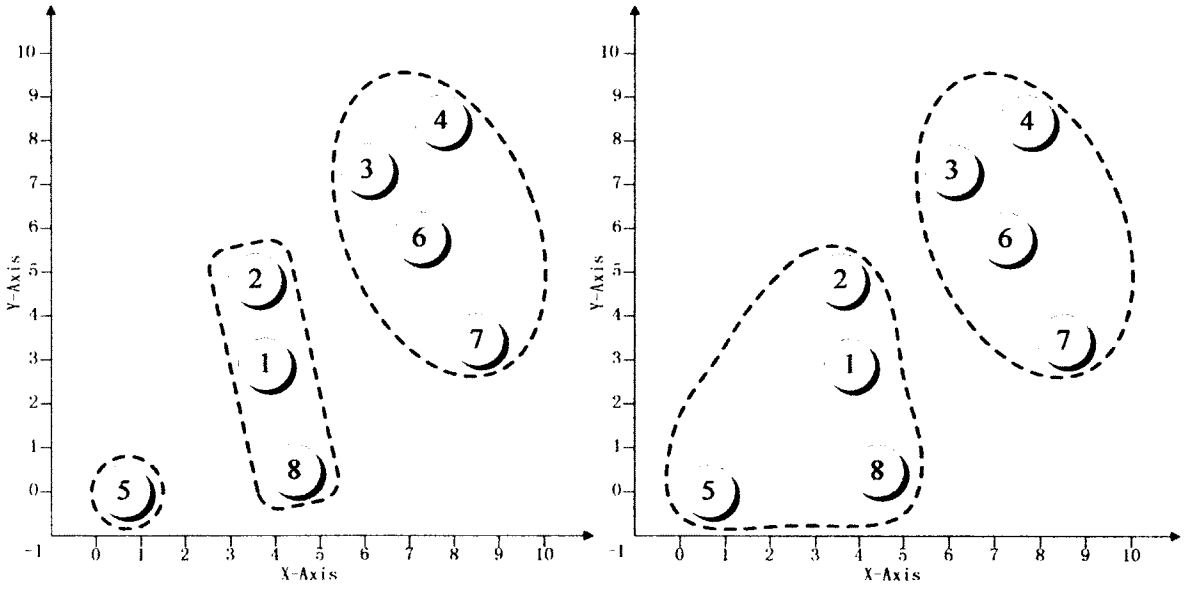
Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{③,⑥,⑦}	*2.72*	0.143

Cluster {⑤}

Neighbor Cluster	Coefficient	
	Quanti tative	Quali tative
{①,②,⑧}	***3.83***	0

(b) Updated resemblance matrices using SLINK with quantitative and qualitative coefficients

Figure 4-4 H-DHAC: Cluster formation results and updated resemblance matrices in the second step



(a) Result of cluster formation in the third step (b) Result of cluster formation in the final step

Figure 4-5 Results of cluster formation in the third and final step

In the next round, these clusters will keep repeating the previous procedures.

According to resemblance matrices in Figure 4-4 (b), cluster {4} will form a new cluster with cluster {3, 6, 7} following the same rule previously described. The result of this step is shown in Figure 4-5 (a). After this round, there is only one singleton cluster left in this network which is cluster {5}. And the requirement of minimum cluster size in this example is at least two sensor nodes in one cluster. Therefore, cluster size control is performed on cluster {5}, which will send a merge request to its closest neighbor cluster which is cluster {1, 2, 8}. Then cluster {1, 2, 8} is going to respond to this request with a CONFIRMATION message. Finally, as can be seen in Figure 4-5 (b), at the new cluster {1, 2, 5, 8} is formed and the cluster formation is thus completed.

Overheads in Cluster Formation of H-DHAC

As a distributed clustering protocol, overheads are necessary for H-DHAC in order to provide better management of the network. Exchanging extra information (GPS status, CL) between neighbors is needed to support hybrid data. Some control messages, e.g., INVITATION, CONFIRMATION, REJECTION, and INFORM are used to control the formation of clusters. Sensor nodes need to calculate and update local resemblance matrix with two kinds of coefficients.

The effect of overheads on energy consumption is considered in the simulations by using the energy model which is defined in section 5.1.2.

4.5.3 Transmission Power Control

The clustering process in H-DHAC is completed at the end of the cluster formation stage. The next step is to begin data sensing and gathering. Since static clustering has been accomplished, sensor nodes will only communicate within their own cluster from now on (except when sending gathered data to BS). Hence it is desirable to only use minimum transmission power to communicate so as in order to achieve high energy efficiency. It is assumed that in simulations, the minimum transmission power can be calculated with regard to the communication between two sensor nodes when both sensor nodes have GPS location information available [61, pages 85-86]. However, for those pairs where only one or both sensor nodes have no GPS location information, the minimum transmission power cannot be calculated. In this case, the minimum transmission power between those pairs will be estimated using TPC.

The Adaptive Transmission Power Control (ATPC) mechanism [45] has been adopted as the TPC scheme for H-DHAC. According to the experiment's results, ATPC manages to keep the end-to-end communication quality above 98%. In terms of energy saving, ATPC has saved 56.4% of energy compared to energy consumption using the maximum transmission power. These are the main reasons that this thesis integrates ATPC to H-DHAC.

The main operation of ATPC is demonstrated as follows: a sensor node that

belongs to a certain cluster will first send several beacons using different transmission power levels. Other sensor nodes of that cluster are the receivers of this beacon message. Each beacon receiver measures the RSSI/LQI (Link Quality Indicator) values of these beacons, and then sends the measured values back to the sender as a feedback packet. The minimum transmission power required will be estimated by the sender using ATPC algorithm for each sensor node according to the collected information in the feedback packet.

Since sensor nodes are stationary after deployment, TPC is a one-time only operation in H-DHAC. Knowing the minimum transmission power between each pair of sensor nodes that belong to a cluster will result in large energy saving in the upcoming data transmissions. After ATPC has been operated, sensor nodes which need TPC can achieve the same level of energy saving as those sensor nodes that have location information. In other words, all sensor nodes can reach their theoretical minimum transmission power after ATPC.

4.5.4 Scheduling and the Data Gathering Stage

In H-DHAC, data gathering is operated in a hierarchical manner. There are two phases in each round of data gathering: the intra-cluster transmission phase and CH to BS transmission phase. In the intra-cluster transmission phase, each CM sends its own data to the CH. Once the CH has collected all data from CMs, it will perform data fusion to

remove redundancy and compress all data (including its data) into one packet. Then each CH will send the gathered data of its own cluster to BS. This two-phase data gathering process is repeated several times in one round; and each time it is called a frame, see Figure 4-6.

To avoid intra-cluster interference, scheduling is needed for the intra-cluster transmission phase. Similar to LEACH [61], the TDMA approach is adopted for intra-cluster data transmissions. The CH will first create a TMDA schedule and assign a distributed time-slot to every CM. A CM can only send data to CH once in one frame. After a CM has received the TDMA schedule, and in order to avoid unnecessary energy consumption, it will switch off its radio component until the next specific time-slot assigned to it is reached. The CM will send its data to CH within this time-slot, and then again switches its radio component off till the next time-slot comes in the next frame. Since the transmission power of sensor nodes has been adjusted to the lowest, it reduces the possibility of intra-cluster interference. However, intra-cluster interference could still happen due to the broadcast nature of radio medium. To further reduce the possibility of interfering with neighbor clusters in the intra-cluster data transmission phase, CDMA [61] is used to combine with TDMA when CMs transmit data to CH. Different spreading codes are used at different clusters, in this way, CH can filter out the desired data using the specific spreading code.

At the end of each frame, when CH has received data from all CMs, it will

perform the data fusion operation in order to prepare for transmitting the gathered data to BS. All CHs will use one specific BS spreading code in the CH to BS transmission phase. To avoid inter-cluster interference, a CH will sense whether there is another CH using this BS spreading code by CSMA. If yes, then the CH will wait in order to send its data. If no other data transmission activity is using the BS spreading code, the CH will send the gathered data to BS. Afterwards, the next frame's operation will begin.

The relationship between the round, frame and time-slot as well as the one round data transmission operation is illustrated in Figure 4-6.

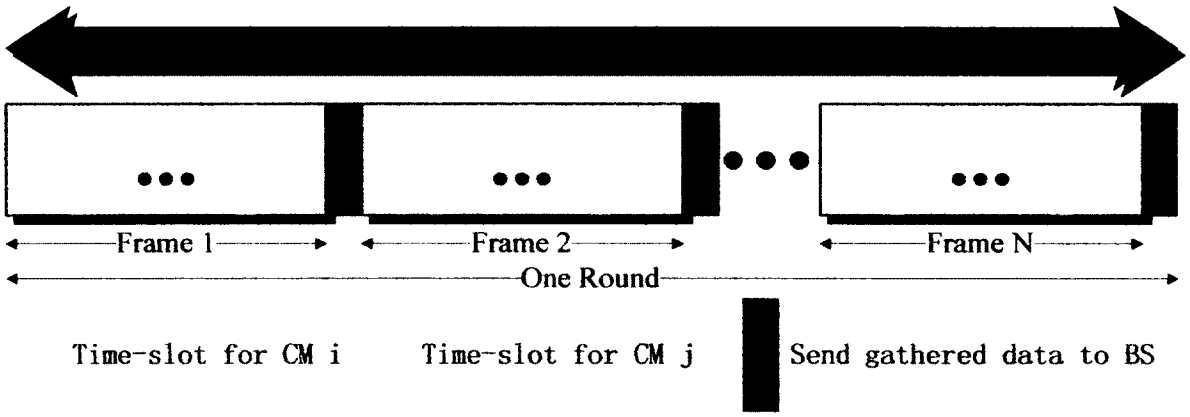


Figure 4-6 One-round data transmission operation in H-DHAC

4.5.5 Cluster Maintenance Stage

A CH plays an important role in a cluster; not only does it manage the process of forming clusters, it also takes care of data gathering. Once the network starts data transmissions, CH will be the only sensor node that keeps working without turning its radio component off. CH takes great responsibility, and consumes a lot of energy while managing data

transmissions with CMs and BS. Therefore, for cluster maintenance, CH rotation is needed in order to achieve the energy balance between CM and CH.

The time for CH rotation (i.e. time for one round) is defined by the following equation [61, page 94]:

$$T_{rotation}(s) = 0.08s \times \frac{E_{initial}(J)}{0.009J} \tag{4-4}$$

where $E_{initial}$ denotes average initial energy of sensor nodes; the unit of $T_{rotation}$ is in second. Thus, if we set $E_{initial}$ to be 1, the approximate time for CH rotation will be 8.89 seconds.

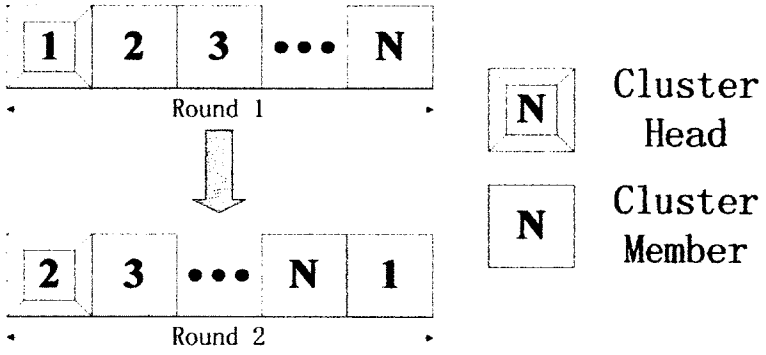


Figure 4-7 Automatic cluster head rotation

In a normal case, the CH rotation will be performed automatically when $T_{rotation}$ is reached. As shown in Figure 4-7, a cluster has N sensor nodes with node IDs from 1 to N. In the first round of data transmission, node-1 is the CH and node-2 is the CM which has been assigned as the first one in the TDMA schedule and as the next CH. Then when the first round ends, CH rotates automatically and node-2 becomes the CH. At the same time, node-1 has been moved to the end of the TDMA schedule. In this way, every sensor

nodes of the cluster will have the same opportunity to become CH.

However, when one of the following situations occurs, rescheduling of CH rotation is necessary: (i) if the energy level of the current CH is lower than a certain threshold E_{th}; or (ii) if there is a change in the cluster, such as when a sensor node runs out of energy and is removed from the cluster.

The energy threshold E_{th} is defined by this equation [14]:

$$E_{th} = P\% \times E_{average} \tag{4-5}$$

where $E_{average}$ is the average residual energy of the sensor nodes of that cluster; $P\%$ is the predefined percentage. According to the analysis in DHAC [14, pages 71-72], the optimum percentage is 60% in terms of energy balancing. Hence if the CH is running at an energy level lower than 60% of the cluster average, the rescheduling for CH rotation and TDMA schedule will be started. The sensor nodes of the rescheduling cluster will be arranged by residual energy in a descending manner; the one with the highest residual energy will become the CH and the first in the new CH rotation schedule, while the one with the lowest residual energy (normally it will be the former CH) will be put at the end of the new schedule. A new TDMA schedule for the next round is also created in this process, which follows the same order as the CH rotation schedule without CH in it.

An example of CH rotation rescheduling is presented in Figure 4-8. There are five sensor nodes with node IDs from 1 to 5 in this cluster. E_{th}, which is 60% of the average residual energy of that cluster, is calculated and shown as 0.2316. As a result, the CH

node-1 at round N has a lower energy than the threshold E_{th}. A new CH rotation schedule must be made for the next round. Following the rules of rescheduling, node-2 has the highest residual energy in this cluster, so that node-2 becomes the CH for the next round. Other sensor nodes will become or remain as CM and are arranged by their residual energy in descending manner. In the end, a new CH rotation schedule, 2-3-5-4-1 is created. CH will rotate automatically until special previously described situations will happen again. The new TDMA schedule of this cluster for round N+1 will be: 3-5-4-1.

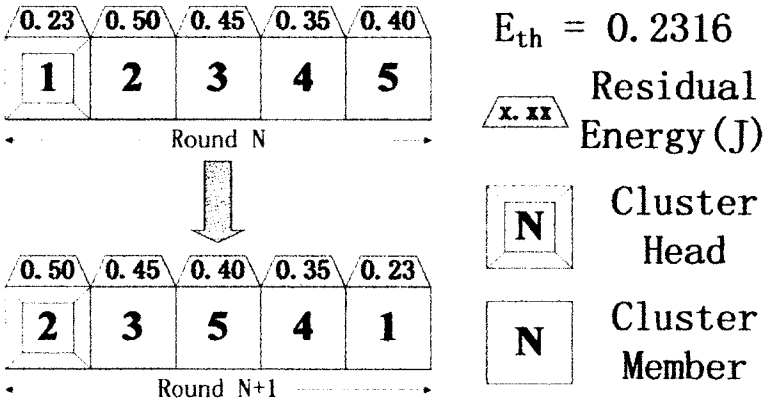


Figure 4-8 Cluster head rotation rescheduling

Chapter 5

Simulation Results and Performance Evaluation

In this chapter, the simulation performance of the H-DHAC protocol will be presented and discussed. In order to have a thorough evaluation, LEACH, LEACH-C and the original DHAC using location data (DHAC-LOC) or connectivity data (DHAC-CON) separately are also simulated under the same environment for comparison. The simulation is run on simulation software NS-2 (version 2.29) with an extension of LEACH which has been developed by the researchers of the LEACH protocol. NS-2 is a discrete event network simulator, which is suitable for wired and wireless network simulations. It is a popular simulation tool for sensor networks because its extensibility is high [63].

5.1 Simulation Environments

A number of parameters have been considered for the experiments. The main parameters include various network field sizes, the location of BS, different clustering protocols, and the percentage of the nodes that have GPS location information. What follows will explain each of the parameters related to the experimental environment.

The randomly created network topologies have been tested on contain 100

homogeneous nodes, in which every node is randomly deployed. These topologies have three different field sizes: 100×100 m^2, 200×200 m^2 and 300×300 m^2, as they have been used in many studies in WSNs. The locations of BS are set to be in the middle of the field edge and 200 meter away from the middle of the field edge. The following sums up the 6 different combinations of field size and BS location:

- 100 × 100 m^2, BS at (50 m, 100m); 200 × 200 m^2, BS at (100 m, 200m); 300 ×300 m^2, BS at (150 m, 300m)
- 100 × 100 m^2, BS at (50 m, 300m); 200 × 200 m^2, BS at (100 m, 400m); 300 ×300 m^2, BS at (150 m, 500m)

The simulation of LEACH, LEACH-C, DHAC-LOC and DHAC-CON will be performed on 10 different random topologies for each combination of field size and BS location. The network field size, where the difference in performance between DHAC-LOC and DHAC-CON is most significant is chosen to be compared with H-DHAC. H-DHAC is the clustering protocol that works in any percentage of GPS availability (0 - 100%). It works in a similar fashion to DHAC-LOC in its best case scenario (100%) and to DHAC-CON when 0% of sensor nodes have GPS location data available. Hence, a straight forward comparison between H-DHAC and the original DHAC can be shown by choosing the network field size in which the difference in performance between DHAC-LOC and DHAC-CON is most significant. In addition, the performance of H-DHAC with four different percentages (90%, 70%, 50%, and 30%) of

GPS location information availability will be shown in order to provide a comprehensive study and show the effect of the percentage of sensor nodes that have GPS location information. The UPGMA is used as the HAC method for DHAC-LOC, DHAC-CON and H-DHAC.

More practically, the sensor nodes without GPS are randomly selected in each run of simulation for H-DHAC. Therefore, there will be 10 simulation runs on each topology to minimize the influence of such randomness. So, for each combination of network field size and the BS location [300×300 m^2 with (150 m, 300m) and (150 m, 500m) when the simulation involves H-DHAC], there will be 100 simulation runs for H-DHAC with the specific percentage of GPS availability. Since this thesis will study four different percentages of GPS availability for each topology, there will be 800 simulation runs for H-DHAC in total which represents a broad coverage of experiments for H-DHAC.

5.1.1 Radio model characteristics and simulation parameters

The radio model selected for simulations plays a crucial role in the simulation results. This thesis mostly uses the one depicted in LEACH [61] which has also been adopted by many clustering protocols in WSNs. Table 5-1 shows the formulas and parameters used in the simulation, including the radio model characteristics.

Table 5-1 Simulation parameters and values

Parameter	Symbol	Value
Sensor Node Number	N_{node}	100
Field Size	S_{field}	$100\times100\ m^2$, $200\times200\ m^2$, $300\times300\ m^2$
Channel Bandwidth	R_b	2 Mbps
Radio Electronic Energy Dissipation Rate	E_{elec}	50 nJ/bit
Minimum Receiver Power	$P_{r\text{-}th}$	6.3 nW
Radio Amplifier Energy Dissipation Rate	$\epsilon_{friss\text{-}amp}$	$\frac{P_{r-th}(4\pi^2)}{R_b G_t G_r \lambda^2} = 5\ pJ/bit/m^2$
	$\epsilon_{two\text{-}ray\text{-}amp}$	$\frac{P_{r-th}}{R_b G_t G_r h_t^2 h_r^2} = 0.00065\ pJ/bit/m^4$
Cross-over distance for switching Friss and Two-ray Ground Attenuation Models	$d_{crossover}$	$\frac{4\pi h_t h_r}{\lambda} = 87\ m$
Initial Energy	$E_{initial}$	1 J/node
Data Fusion Energy Dissipation Rate	E_{fusion}	5 nJ/bit
Data Gathering Rate	G_{rate}	5 TDMA frames per 10 sec
Antenna gain factor	G_t, G_r	1
Antenna height	h_t, h_r	1.5 m
Radio wavelength	λ	0.325 m
Data Packet Size	S_{data}	500 Bytes
CDMA Spreading Factor	$C_{spreading}$	5

The radio characteristic model is simulated by two models: Friss and Two-ray ground attenuation models [61, pages 81-83]. Cross-over distance $d_{crossover}$ is the marginal distance between these two models. The Friss model is used when transmission distance is below $d_{crossover}$, otherwise the Two-ray ground model is chosen. $\epsilon_{friss\text{-}amp}$ and $\epsilon_{two\text{-}ray\text{-}amp}$ are the radio amplifier energy dissipation rates for these two radio models. Most of the parameters are adopted from LEACH [61, page 88], with some adjustments. Note that the

Channel Bandwidth R_b is 2 Mbps which is different from 1 Mbps in the paper of LEACH. Consequently, $\epsilon_{friss\text{-}amp}$ and $\epsilon_{two\text{-}ray\text{-}amp}$ which are related to R_b, are changed from 10 pJ/bit/m^2 and 0.0013pJ/bit/m^4 to 5 pJ/bit/m^2 and 0.00065pJ/bit/m^4, respectively. The spreading factor in the simulation is set to 5 for H-DHAC, DHAC-CON and DHAC-LOC (because the number of clusters is 5 for these protocols). For LEACH and LEACH-C, the values of the spreading factor were 8 and 5, respectively.

5.1.2 Energy Model

Energy consumption is a key factor to consider within the context of WSNs as adopted in this thesis. Hence, it is important to understand the energy model used in the simulation. The energy model is also based on the one used in LEACH [61, pages 83-86] and many clustering protocols. The energy model consists of three energy dissipation components: transmitting, receiving, and computation; each of which is fleshed out in the following.

- Transmitting Energy Dissipation

  The energy consumption model for a sensor node to transmit any type of data is defined by the following equation:

$$E_{Tx}(l,d) = \begin{cases} l \times \left(E_{elec} + \epsilon_{friss-amp} \times d^2, d < d_{crossover}\right) \\ l \times \left(E_{elec} + \epsilon_{two-ray-amp} \times d^4, d \geq d_{crossover}\right) \end{cases} \quad (5\text{-}1)$$

  where l is the total amount of data transmitted in bits; the distance between the transmitter and receiver is d; and E_{elec} is the energy consumption for transmitting one

bit of data. Whether the Friss space model ($\epsilon_{friss\text{-}amp}$ and d^2) or the Two-ray ground propagation model ($\epsilon_{two\text{-}ray\text{-}amp}$ and d^4) is chosen, it will depend on the cross over distance $d_{crossover}$.

- Receiving Energy Dissipation

  To receive any data from a sender, the energy is consumed as follows:

$$E_{Rx} = l \times E_{elec} \tag{5-2}$$

  In equation (5-2), l is the total amount of data received in bits; E_{elec} is the energy consumption for receiving one bit of data, which is the same as the E_{elec} in equation (5-1). The energy consumed in receiving data is not affected by the distance between transmitter and receiver.

- Data Fusion and Computational Energy Dissipation

  The energy consumption of data fusion and computation of the resemblance matrix is calculated by the following:

$$E_{comp} = E_{fusion} \times l \tag{5-3}$$

  where l represents the total amount of data processed in bits; and E_{fusion} is the energy consumption for processing one bit of data.

5.2 Simulation Results and Evaluation

5.2.1 Performance Metrics

There are three common metrics for evaluating the performance of clustering protocols:

network lifetime, energy efficiency and the total amount of transmitted data.

- *Network Lifetime*

  When most of the sensor nodes run out of energy, the sensor network may not function properly since the data that those remaining sensor nodes collect may be scarce and may not be valuable. But there is no generally accepted definition of a sensor network's lifetime since different applications have different requirements. In this thesis, network lifetime is defined as the period of time from the start of the network until 90% of the sensor nodes die ($\mathbf{T_{90}}$). The assumption here is that with less than 10% of sensor nodes remaining, the network may not function as planned. In addition, the time of n% of sensor nodes die ($\mathbf{T_n}$) is also used for comparison.

- Energy Efficiency

  Among the various tasks of a wireless sensor, data transmissions use up most of the energy. For clustering protocols, higher energy efficiency means that a more substantial amount of data has been collected. Hence, energy efficiency is a critical criterion for WSNs, and is defined as the total amount of transmitted data/energy dissipation at T_{90}.

- Total Amount of Transmitted Data

  Since data is the most fundamental interest of WSNs, the total amount of transmitted data is a critical metric for comparing the performance of different protocols. As discussed above, the data collected after most of the sensor nodes had died out may

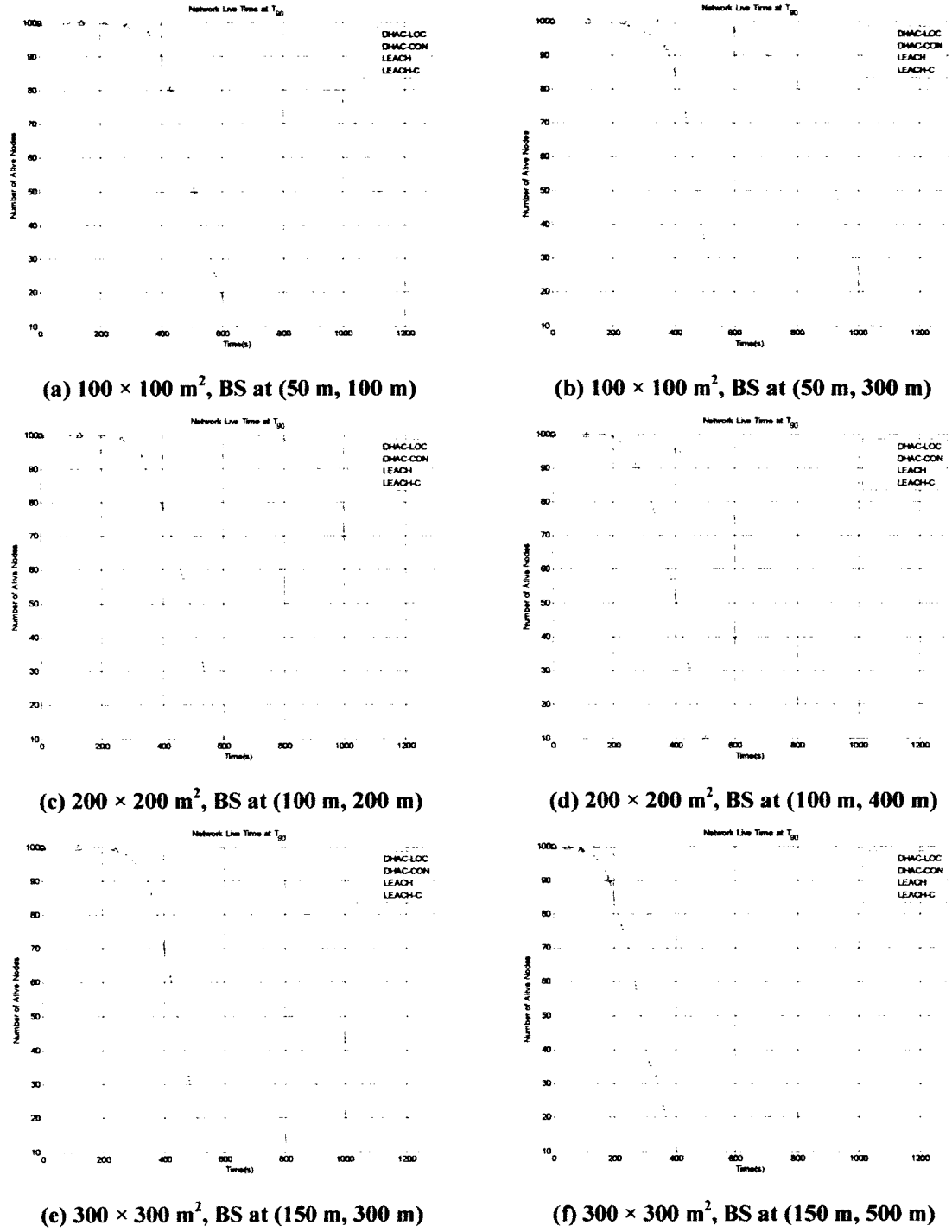
not be useful. Hence, in this thesis, the total amount of transmitted data at $T_{90\%}$ will be used for comparison.

5.2.2 Different Network Sizes

In this part, the simulation is conducted on three different network sizes, which are 100 × 100 m^2, 200 × 200 m^2, and 300 × 300 m^2. The location of BS is set to be on the middle of the edge ((50 m, 100 m), (100 m, 200 m), and (150 m, 300 m), respectively) and outside of the network ((50 m, 300 m), (100 m, 400 m), and (150 m, 500 m), respectively). The performances on different network sizes are compared among LEACH, LEACH-C, DHAC-CON, and DHAC-LOC without H-DHAC. The difference of performance and trend of variation between different network sizes will be studied. One network size will be chosen to be used for comparing H-DHAC with other protocols.

The network lifetime at T_{90} of 6 different combinations of network size and BS location are presented in Figure 5-1 (a) – (f), respectively. The simulation results of network lifetime share a common similarity: DHAC-LOC has the longest lifetime, DHAC-CON comes second, LEACH-C ranks third and LEACH performs worse. It can be discovered that when compared with the same network size (i.e., horizontal comparison), the network lifetime of each protocol in general is longer when the BS is closer to the network. This is because the communication between CHs and the BS consumes a lot of energy. Also, since LEACH-C is a centralized approach, every sensor

node which is alive has to send their information to the BS at beginning of each round, the lifetime of LEACH-C decrease faster when the BS is located outside of the network.



(a) 100 × 100 m^2, BS at (50 m, 100 m)

(b) 100 × 100 m^2, BS at (50 m, 300 m)

(c) 200 × 200 m^2, BS at (100 m, 200 m)

(d) 200 × 200 m^2, BS at (100 m, 400 m)

(e) 300 × 300 m^2, BS at (150 m, 300 m)

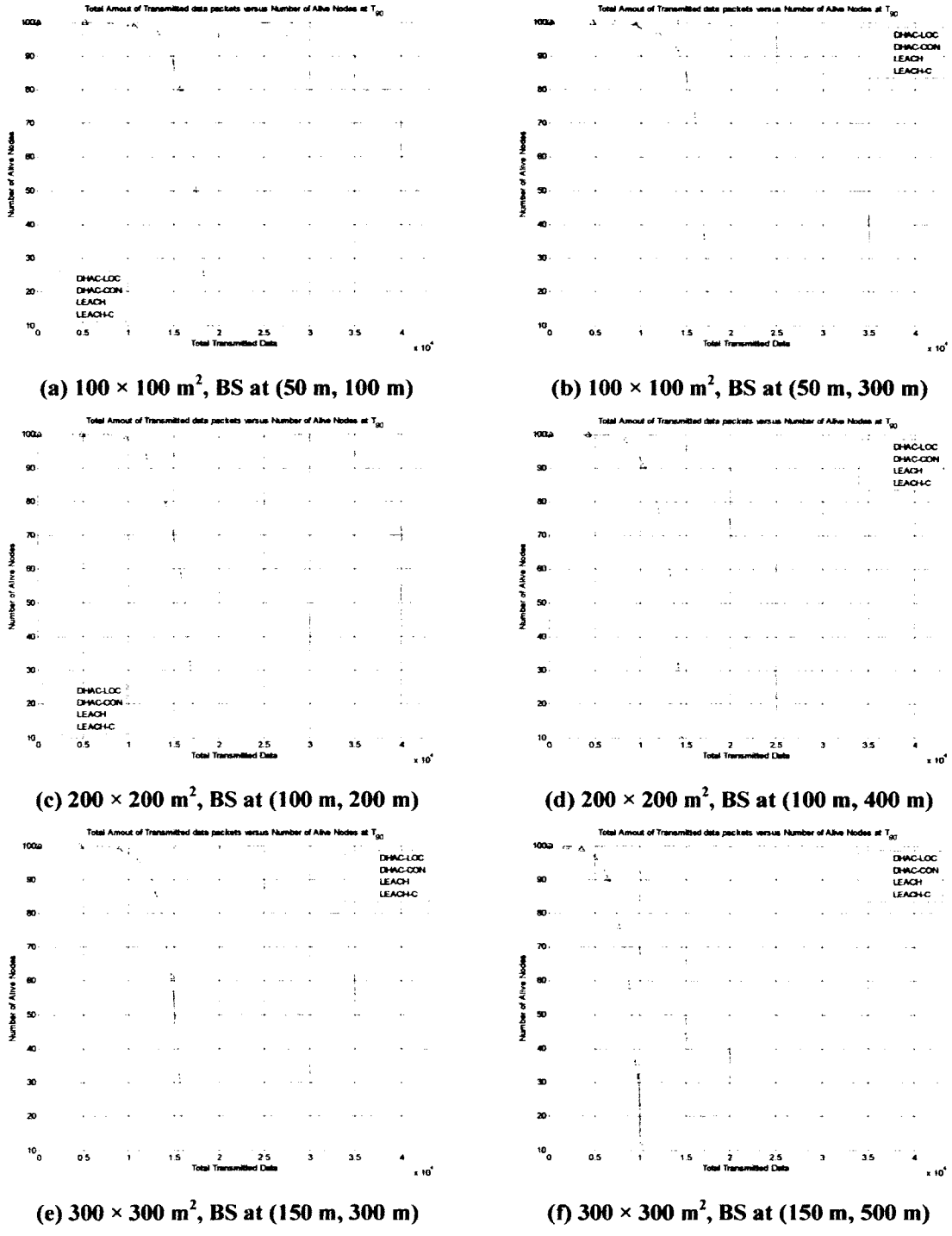
(f) 300 × 300 m^2, BS at (150 m, 500 m)

Figure 5-1 Network lifetime at T_{90}

The difference of network lifetime between DHAC-LOC and DHAC-CON become more evident along with the network size, i.e., compared to DHAC-LOC, the performance of DHAC-CON degrades more quickly in a larger network size. Because in a larger size network, the cost of intra-cluster communication increases (the average transmission distance is longer in a less dense network). Therefore, the communication cost will be higher in a larger network if the cluster formation is not optimum, which results in quicker sensor node death and shorter lifetime. Besides, DHAC-LOC outperforms LEACH and LEACH-C by higher number of percentage when the network size increases. For example, DHAC-LOC outperforms LEACH and LEACH-C by 91.15% and 36.71% at T_{90} in Figure 5-1 (b). When the network size becomes 300×300 m^2 as in Figure 5-1 (f), the differences increase to 113.45% and 52.17%, respectively.

The total amount of transmitted data at T_{90} for 6 different combinations of network size and BS location are shown in Figure 5-2 (a) – (f), respectively. Similar to the result in Figure 5-1, DHAC-LOC performs the best in terms of total amount of transmitted data; DHAC-CON, LEACH-C and LEACH rank in 2^{nd} – 4^{th} place accordingly. Since total amount of transmitted data is related to network lifetime (e.g., longer lifetime suggests more time to transmit data; data transmission consumes a lot of energy which affects the lifetime), they share a similar pattern: the location of the BS affects the performance, the total amount will be reduced if the BS is located farther; the difference of total amount of transmitted between DHAC-LOC and DHAC-CON is more

obvious when the network size increases; the number of percentage that DHAC-LOC outperforms LEACH and LEACH-C will be higher in larger networks.



(a) 100 × 100 m^2, BS at (50 m, 100 m)

(b) 100 × 100 m^2, BS at (50 m, 300 m)

(c) 200 × 200 m^2, BS at (100 m, 200 m)

(d) 200 × 200 m^2, BS at (100 m, 400 m)

(e) 300 × 300 m^2, BS at (150 m, 300 m)

(f) 300 × 300 m^2, BS at (150 m, 500 m)

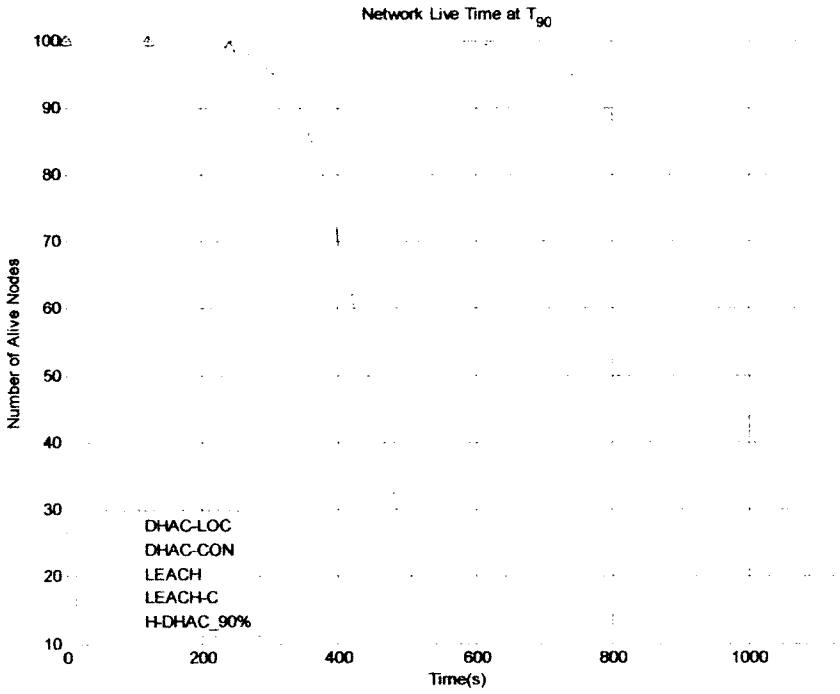
Figure 5-2 Total amount of transmitted data at T_{90}

Therefore, according to the characteristics which those four different protocols have shown, the network size of 300 × 300 m^2 is chosen to include H-DHAC for comparison in the next section. It is because the difference of performance between DHAC-CON and DHAC-LOC is the largest in the network size of 300 × 300 m^2, which can provide a straightforward comparison between H-DHAC and the original DHAC. Since H-DHAC uses both qualitative connectivity data and quantitative location data for clustering, hence in theory, the performance of H-DHAC should be in between DHAC-CON and DHAC-LOC. H-DHAC should not outperform DHAC-LOC and should be better than DHAC-CON since DHAC-LOC and DHAC-CON are similar to two extreme cases of H-DHAC (i.e. H-DHAC with100% and 0% of GPS location data available).

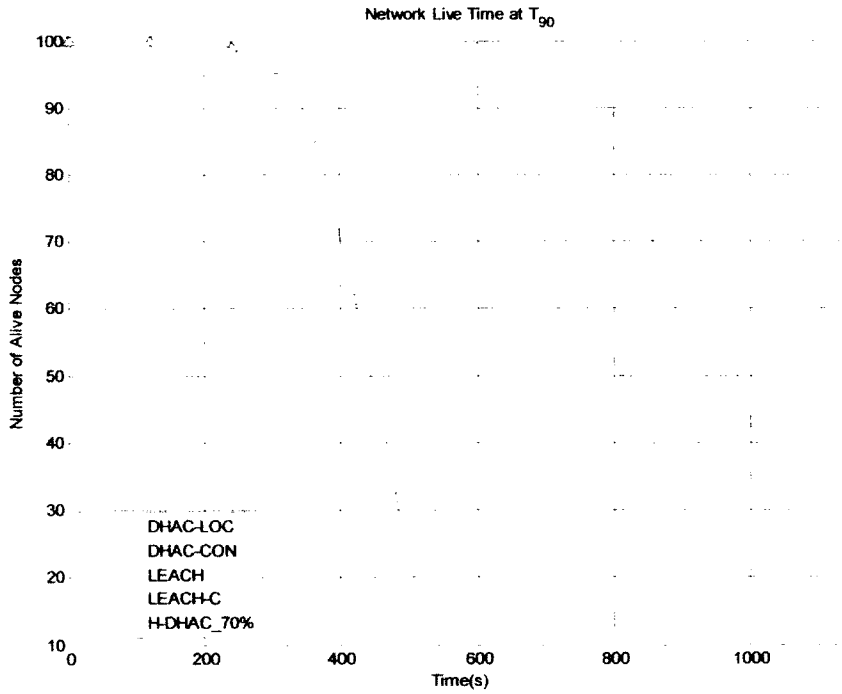
5.2.3 Different Parameters

H-DHAC can work with any percentage of GPS location data available. In this part, four different parameters of GPS location data availability for H-DHAC are used in simulation, which are 90%, 70%, 50%, and 30%. In the following sections, **H-DHAC_xx%** will be used to denote H-DHAC with xx% of GPS location data available (e.g., H-DHAC_90%). The network size is 300 × 300 m^2 as chosen and the BS is located at (150 m, 300 m). The performances of H-DHAC with four different parameters are compared with DHAC-LOC, DHAC-CON, LEACH and LEACH-C.

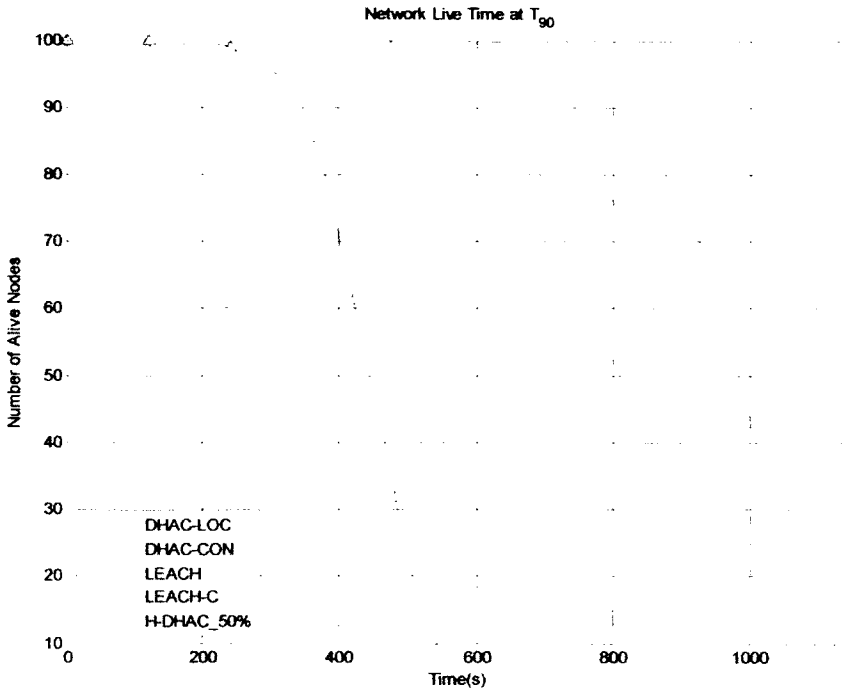
5.2.3.1 Network Lifetime



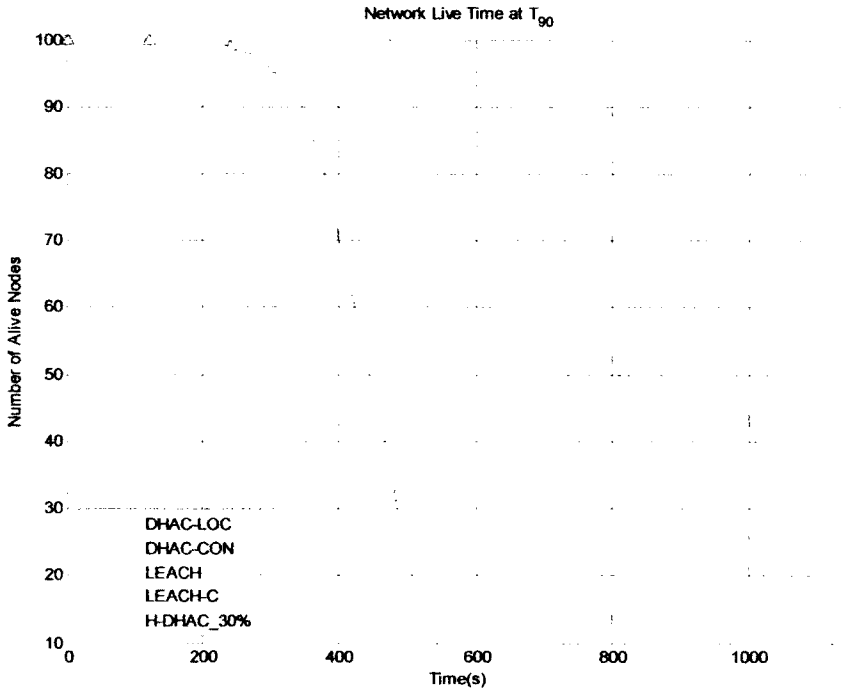
(a) H-DHAC_90%



(b) H-DHAC_70%



(c) H-DHAC_50%



(d) H-DHAC_30%

Figure 5-3 Network lifetime until T_{90} with BS at (150 m, 300 m)

The network lifetime until T_{90} including H-DHAC with four different parameters is

presented in Figure 5-3 (a) – (d), respectively. It can be observed from Figure 5-3 (a) – (d) that H-DHAC_90%, H-DHAC_70%, H-DHAC_50%, and H-DHAC_30% all have better performance than DHAC-CON, LECH-C and LEACH throughout the entire lifetime. As it is shown in Figure 5-3 (a), lifetime of H-DHAC_90% is very close to DHAC-LOC. Compared to other protocols at T_{90}; H-DHAC_90% prolongs 105.84% from LEACH, 37.02% from LEACH-C, 9.18% from DHAC-CON, and less than DHAC-LOC by only 0.45%. Moreover, H-DHAC_90% prolongs T_{10} by 123.94% from LEACH, 21.37% from LEACH-C, 25.20% from DHAC-LOC, and less than DHAC-LOC by 0.63%.

When it comes to the time that first sensor node dies (T_1), LEACH-C has the shortest T_1 since the energy consumption is higher for a sensor node to communicate with the BS at a large distance. H-DHAC_90% has a little bit higher T_1 compared to DHAC-LOC, because the cluster formation is changed when qualitative data are involved in clustering. The cluster size of H-DHAC_90% could be more evenly distributed but not necessary optimal, which is the reason that the difference between H-DHAC_90% and DHAC-LOC at T_{30} (4.28%) is larger than at T_{10} and T_{90} (the same reason also applies to H-DHAC_70%).

When compared between four different parameters of H-DHAC, the performance of H-DHAC is better with higher percentage of GPS location data available as observed from Figure 5-3 (a) – (d), i.e., H-DHAC_90% > H-DHAC_70% > H-DHAC_50% > H-DHAC_30%. The differences of lifetime between H-DHAC with different parameters

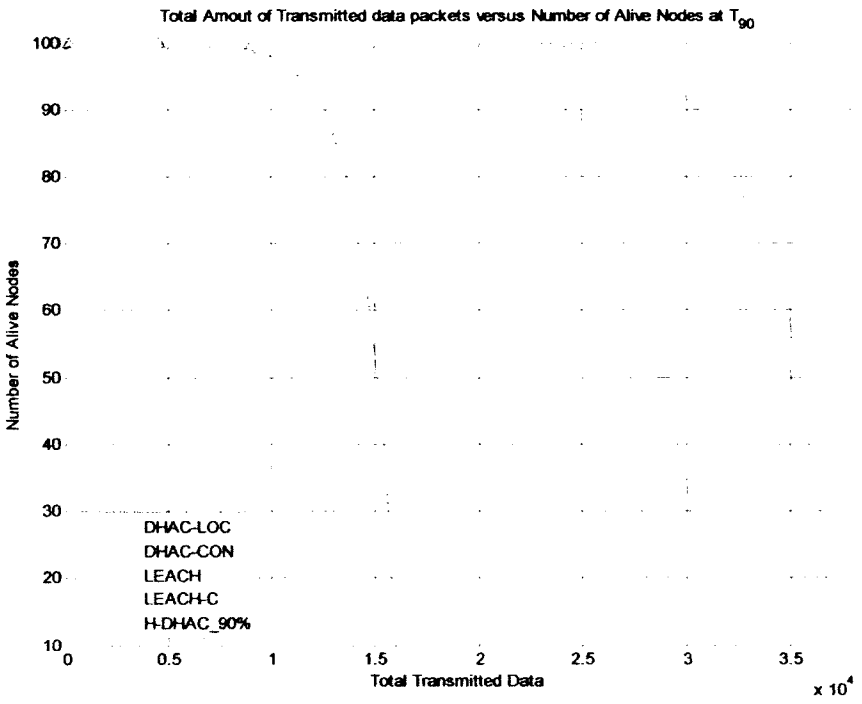
are small. At T_{90}, H-DHAC_50% extends by 1.09% from H-DHAC_30%, H-DHAC_70% extends by 2.86 % from H-DHAC_50%, and H-DHAC_90% extends by 1.61% from H-DHAC_70%. With a larger parameter, the performance of H-DHAC improves smoothly.

5.2.3.2 Total Amount of Transmitted Data

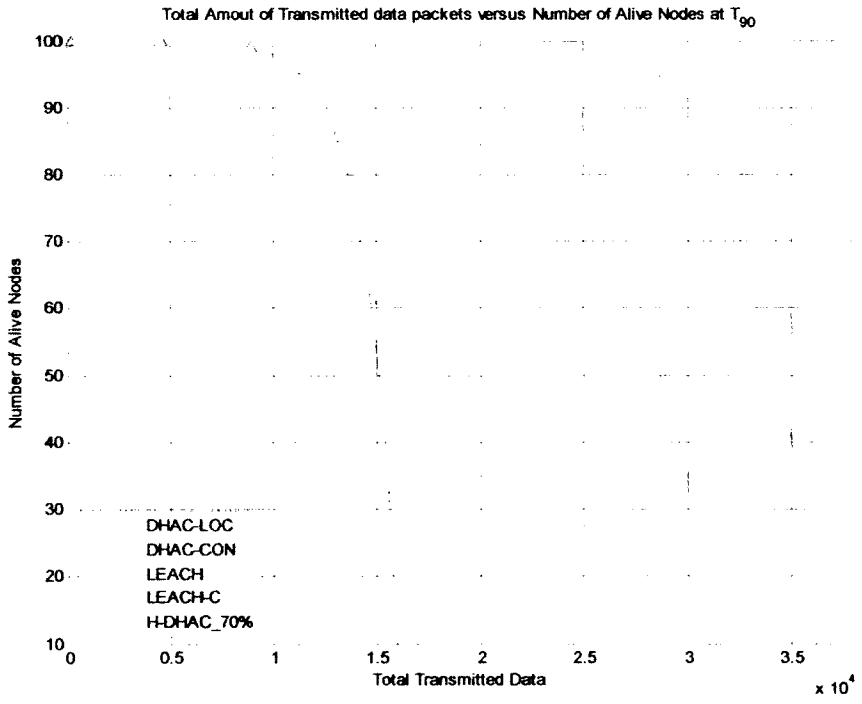
In Figure 5-4 (a) – (d), the total amount of transmitted data until T_{90} including H-DHAC with four different parameters is shown. Also, the detail of total amount of transmitted data at T_1, T_{10},T_{30}, T_{50}, T_{70}, and T_{90} is presented in Table 5-2.

Similar to the situation in network lifetime, H-DHAC with four different parameters all outperform DHAC-CON, LEACH-C and LEACH throughout the entire process. As it is presented in Figure 5-4 (a), the curve of H-DHAC_90% is very close to DHAC-LOC, and H-DHAC_90% transmitted 31 packets (only 0.08%) less than DHAC-LOC at T_{90}.

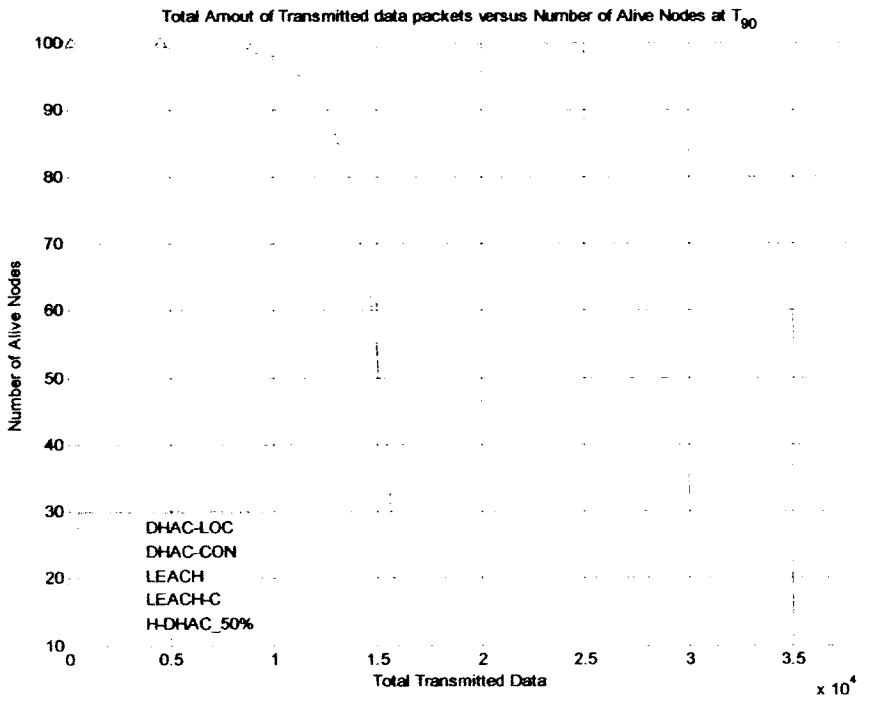
H-DHAC_30%, which has the lowest percentage of GPS location data available in the simulation, transmitted 34288 packets at T_{90}. That is 3113 packets (9.99%) more than DHAC-CON, 6093 packets (21.61%) higher than LEACH-C, 18330 packets (114.86%) over LEACH.



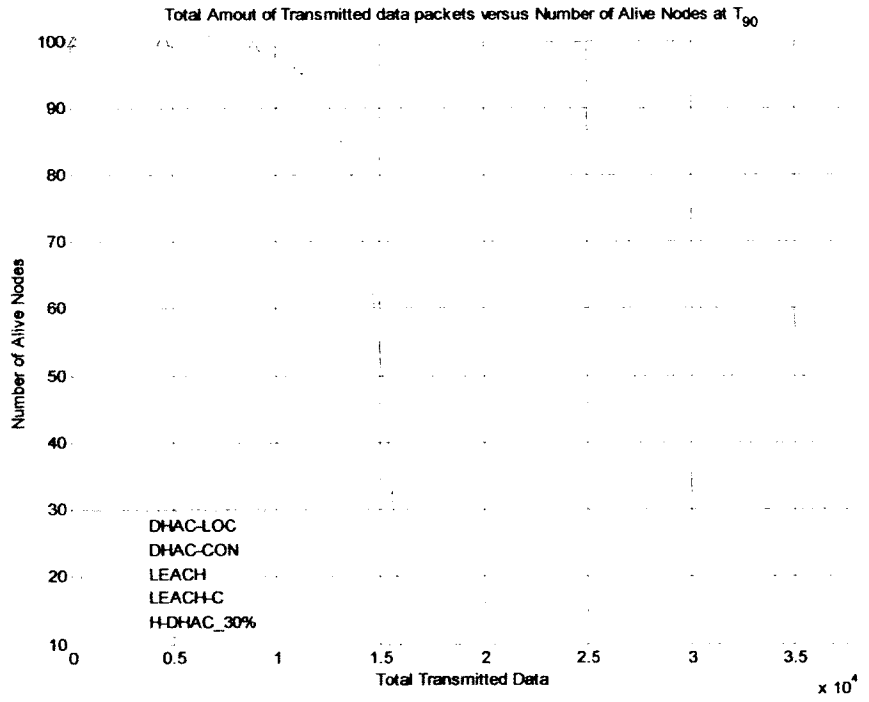
(a) H-DHAC_90%



(b) H-DHAC_70%



(c) H-DHAC_50%



(d) H-DHAC_30%

Figure 5-4 Total amount of transmitted data until T90 with BS at (150 m, 300 m)

The differences between four different parameters of H-DHAC are not large

either. At T_{90}, H-DHAC_50% transmitted 35213 packets which is 2.70% higher than H-DHAC_30%; H-DHAC_70% transmitted 36230 packets which is 2.89% over H-DHAC_50%; H-DHAC_90% transmitted 36906 packets which is 1.87% more than H-DHAC_70%. Since the network lifetime of H-DHAC_90% and H-DHAC_70% at T_1 are longer compared to DHAC-LOC (which is explained in the previous subsection), thus, H-DHAC_90% and H-DHAC_70% have high amount of transmitted data than DHAC-LOC at T_1.

Table 5-2 Total amount of transmitted data with BS at (150 m, 300 m) (T_n denotes the time when n% sensor nodes die)

Protocol	T_1	T_{10}	T_{30}	T_{50}	T_{70}	T_{90}
LEACH	9115	12712	14285	15123	15613	15958
LEACH-C	6748	24939	26522	27315	27886	28195
DHAC-CON	20789	24202	27135	28817	30153	31175
H-DHAC_30%	23651	27693	30481	32044	33690	34288
H-DHAC_50%	23692	29000	31379	33368	34616	35213
H-DHAC_70%	26013	29763	32740	34299	35657	36230
H-DHAC_90%	25176	30603	33600	35101	36440	36906
DHAC-LOC	24821	30456	34430	35495	36527	36937

As the H-DHAC protocol with highest parameter in the simulation, the performance of H-DHAC_90% is similar to DHAC-LOC in terms of both network lifetime and total amount of transmitted data. With higher percentage of GPS location data available for clustering, the performance of H-DHAC becomes better, because the cluster formation is optimized by having more quantitative location data involve in clustering. And while the parameter decreases, the performance of H-DHAC degrades gradually instead of sharply.

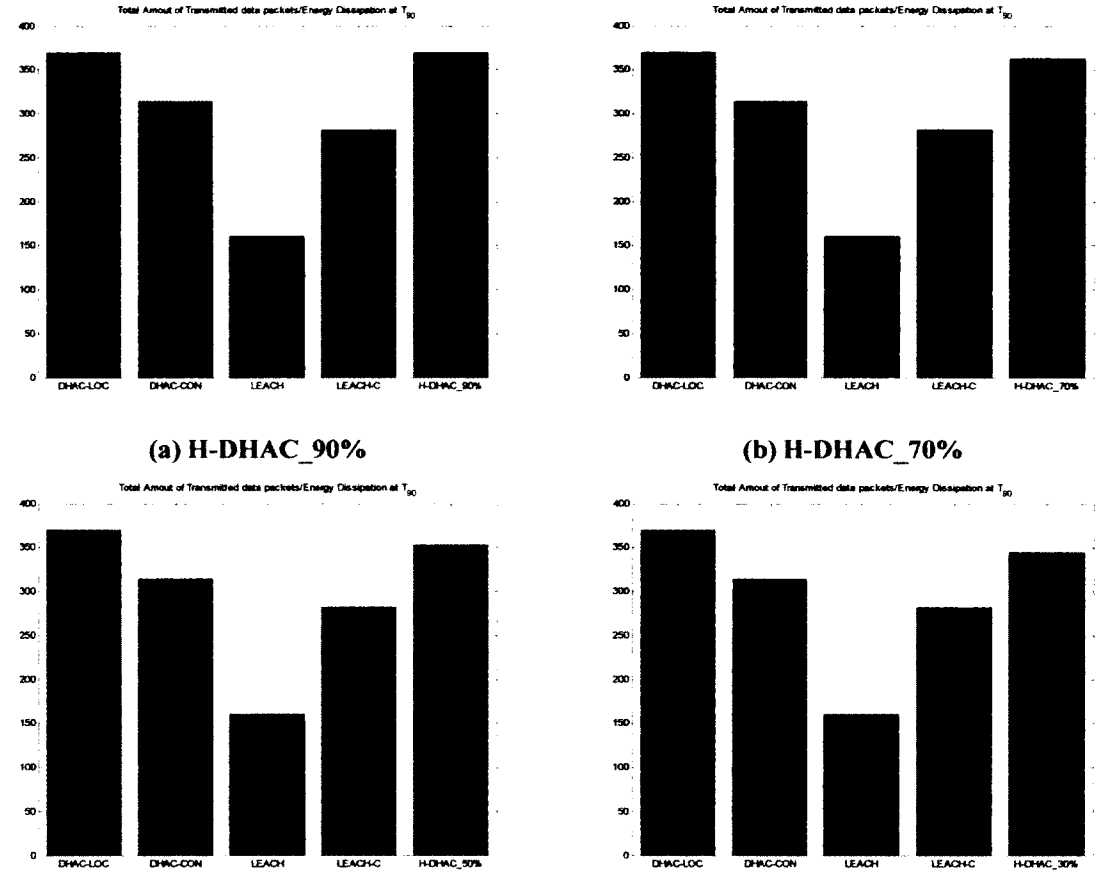
5.2.3.3 Energy Efficiency

In Figure 5-5 (a) – (d), the energy efficiency which is defined as the proportion of total amount of transmitted data / energy dissipation at T_{90}, is compared between H-DHAC with four different parameters and other four protocols.

Since at T_{90}, there are still several sensor nodes working in the network and the simulation is closing to the end. Hence, the energy efficiency status at T_{90} will be adequate to reflect the overall performance. As it can be observed, LEACH has the lowest energy efficiency, while DHAC-LOC performs the best among them. H-DHAC_90% is really close to DHAC-LOC with only 0.06% degradation. H-DHAC_70%, H-DHAC_50%, and H-DHAC_30% rank after H-DHAC_90%, with 1.92%, 2.67%, 2.26% less efficient than H-DHAC_90%, H-DHAC_70%, and H-DHAC_50%, respectively. In addition, H-DHAC_30% is 115.04% more efficient than LEACH; 22.34% better than LEACH-C; and has 9.60% higher efficiency than DHAC-CON.

As it has been shown in the previous comparisons, with 10% of sensor nodes without GPS location data, H-DHAC_90% can achieve the similar performance as DHAC-LOC. Although there is a little difference between H-DHAC_90% and DHAC-LOC during approximately T_{80} to T_{40}, the results in the end are similar with minimum distinction. While the parameter becomes lower, H-DHAC still maintains

higher performance than DHAC-CON, LEACH, and LEACH-C throughout the entire simulation in all aspects.



(a) H-DHAC_90%

(b) H-DHAC_70%

(c) H-DHAC_50%

(d) H-DHAC_30%

Figure 5-5 Total amount of transmitted data/energy dissipation at T_{90} with BS at (150 m, 300 m)

5.2.4 Different BS Locations

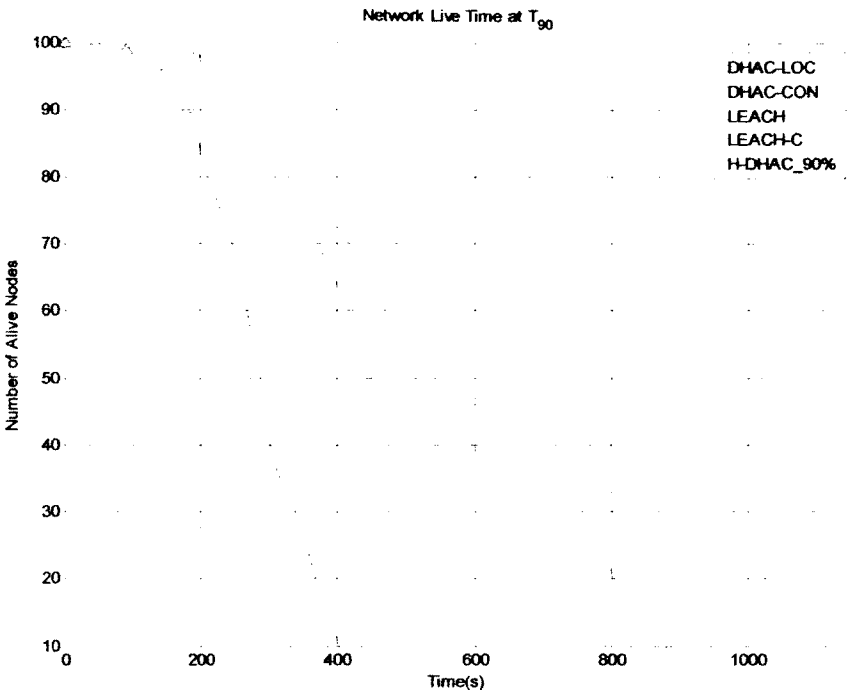
In this subsection, the network size remains at 300×300 m^2, while the BS is located at 200 meters away from the middle of the edge — (150 m, 500 m). H-DHAC with four different parameters is compared with other four protocols.

5.2.4.1 Network Lifetime

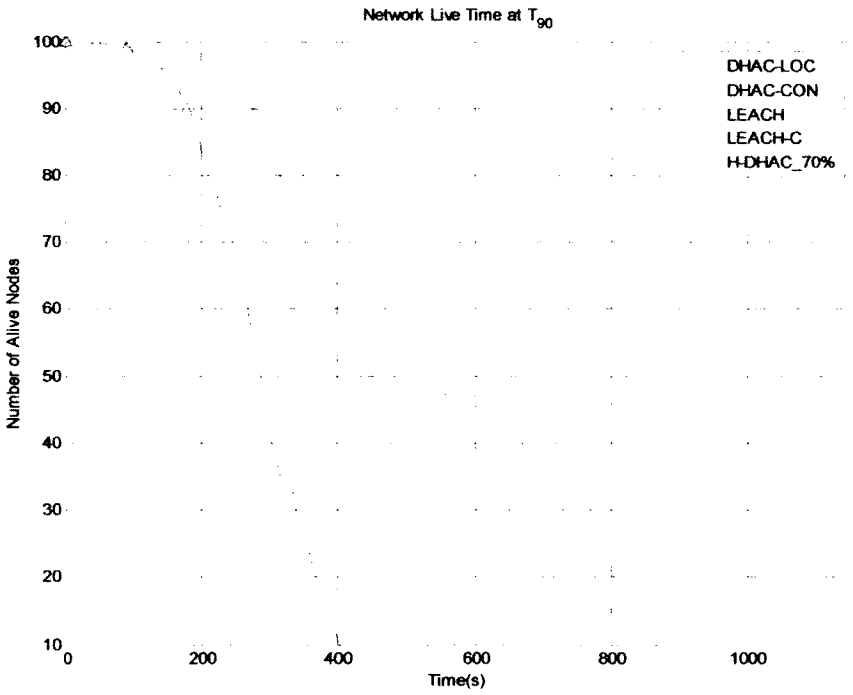
The simulation results of network lifetime when the BS is located 200 m farther away from the network is presented in Figure 5-6 (a) – (d).

The lifetime of each protocol decreases compared to Figure 5-3 in previous section, because the inter-cluster communication cost increases with the distance between sensor nodes and the BS. Similar to Figure 5-3, DHAC-LOC is the best among them, and the performance of H-DHAC_90% is still close to DHAC-LOC with only 0.92% difference at T_{90}. H-DHAC_90% and H-DHAC_70% have the overall better performance compared to DHAC-CON, LEACH-C, and LEACH. Between T_{88} to T_{50} approximately, LEACH-C has a better lifetime than DHAC-CON. This is because the cluster formation using qualitative data is not optimum, some sensor nodes may join the cluster that is not the closest, plus the effect of the increased cost of communication with the BS which result in some certain sensor nodes die earlier. However, from T_{50} to T_{90}, DHAC-CON performs better than LEACH-C again, which is 36.53% better at T_{90} specifically.

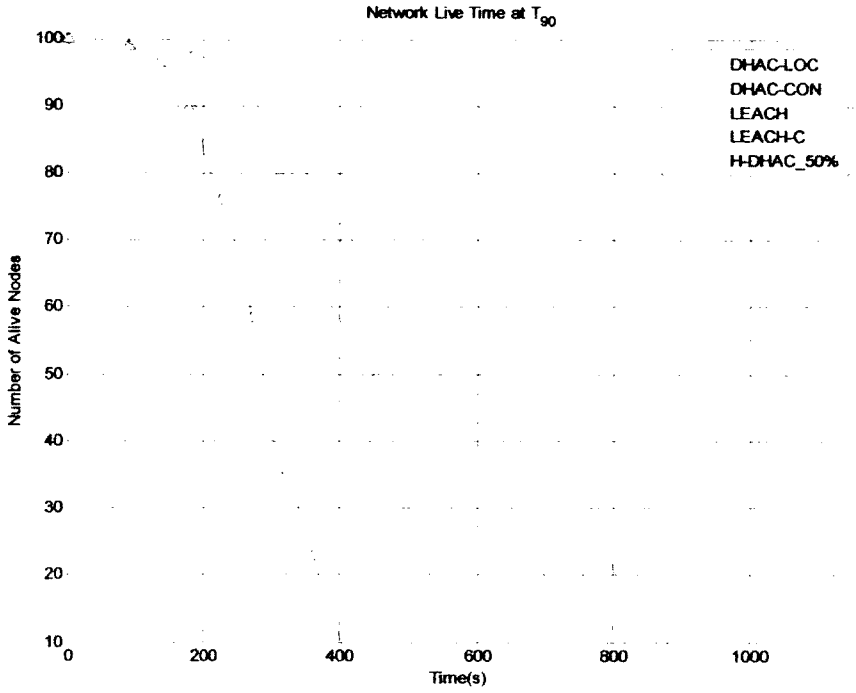
Since H-DHAC uses qualitative data for clustering, the lifetime of H-DHAC_50% and H-DHAC_30% is lower than LEACH-C for a short period of time, but the situation reverse quickly. At the end, H-DHAC_30% prolongs T_{90} 41.03% from LEACH-C, 102.30% from LEACH; and has 25.6 second (3.30%) longer lifetime than DHAC-CON at T_{90}.



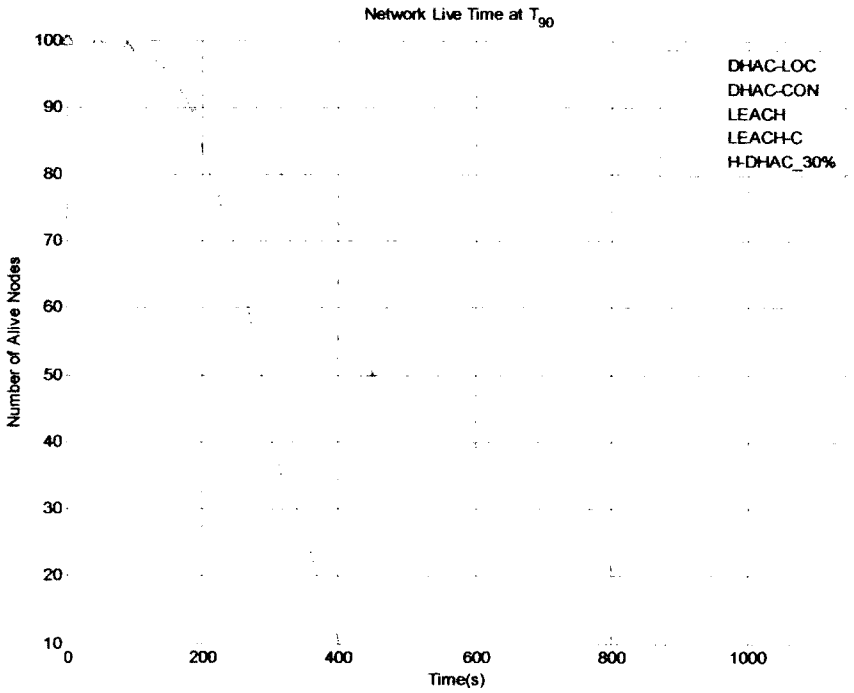
(a) H-DHAC_90%



(b) H-DHAC_70%



(c) H-DHAC_50%



(d) H-DHAC_30%

Figure 5-6 Network lifetime until T_{90} with BS at (150 m, 500 m)

The comparison of H-DHAC with different parameters reveals small differences.

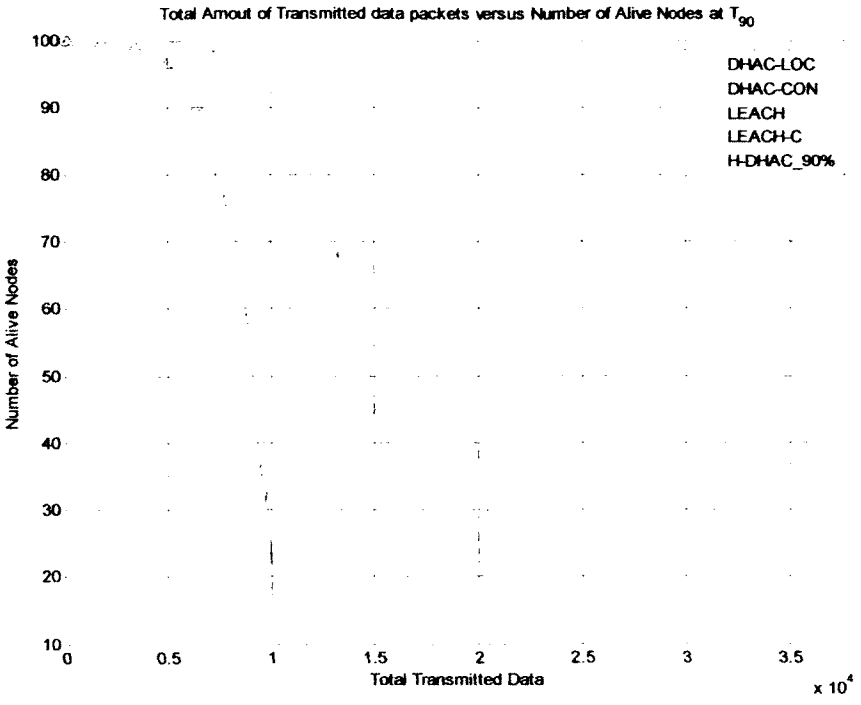
At T_{90}, H-DHAC_50% extends by 2.16% from H-DHAC_30%. H-DHAC_70% has 1.70% longer lifetime than H-DHAC_50% while H-DHAC_90% prolongs 3.32% from H-DHAC_70%.

5.2.4.2 Total Amount of Transmitted Data

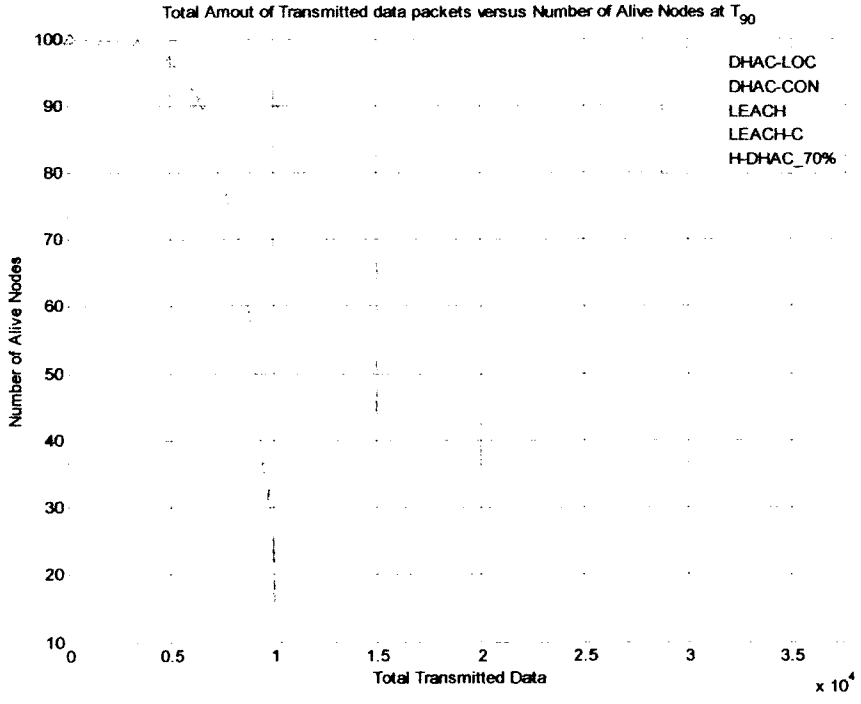
Due to the effect of shorter lifetime and higher inter-cluster data transmission, the total amount of transmitted data have also reduced as it is shown in Figure 5-7 (a) – (d); details of transmitted data amount at T_1, T_{10},T_{30}, T_{50}, T_{70} are also listed in Table 5-3.

As it can be observed, at T_{60}, H-DHAC_90% transmitted 18974 packets which is 938 packets (4.71%) less than DHAC-LOC. However, the difference is reduced to almost half at T_{90}, in which H-DHAC_90% has transmitted 562 packets (2.75%) less than DHAC-LOC.

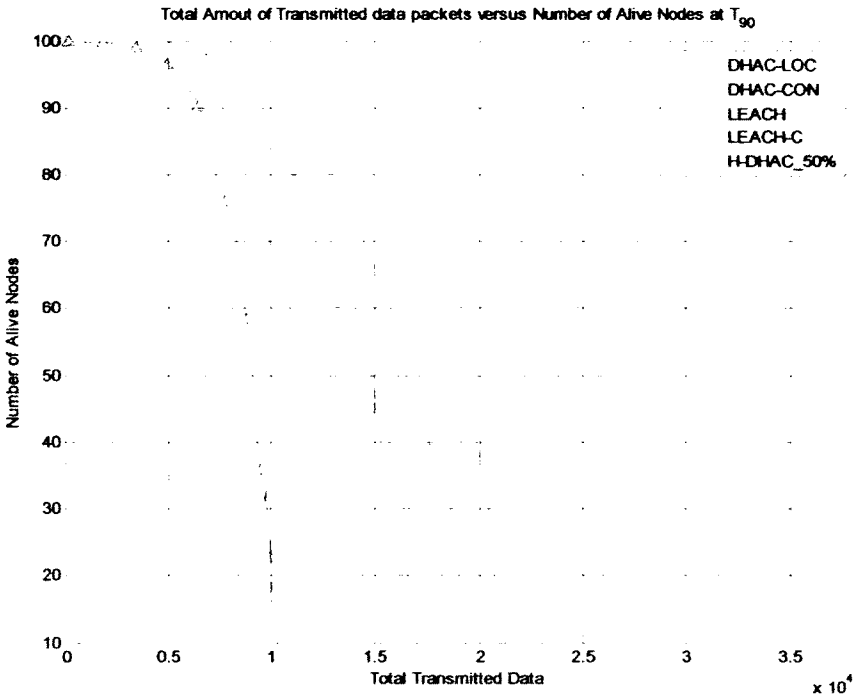
Similar to the simulation results of previous subsection (Figure 5-4 and Table 5-2), the performance of H-DHAC becomes better with higher percentage of location data available. At T_{90}, H-DHAC_70% has 2.78% advantage over H-DHAC_50% while H-DHAC_90% is 3.04% higher than H-DHAC_70%. And the difference between H-DHAC_50% and H-DHAC_30% is 2.59%. Besides, H-DHAC_30% outperforms DHAC-CON, LEACH-C, and LEACH by 1537 packets (8.89%), 2701 packets (16.74%), and 8720 packets (86.21%) at T_{90}, respectively.



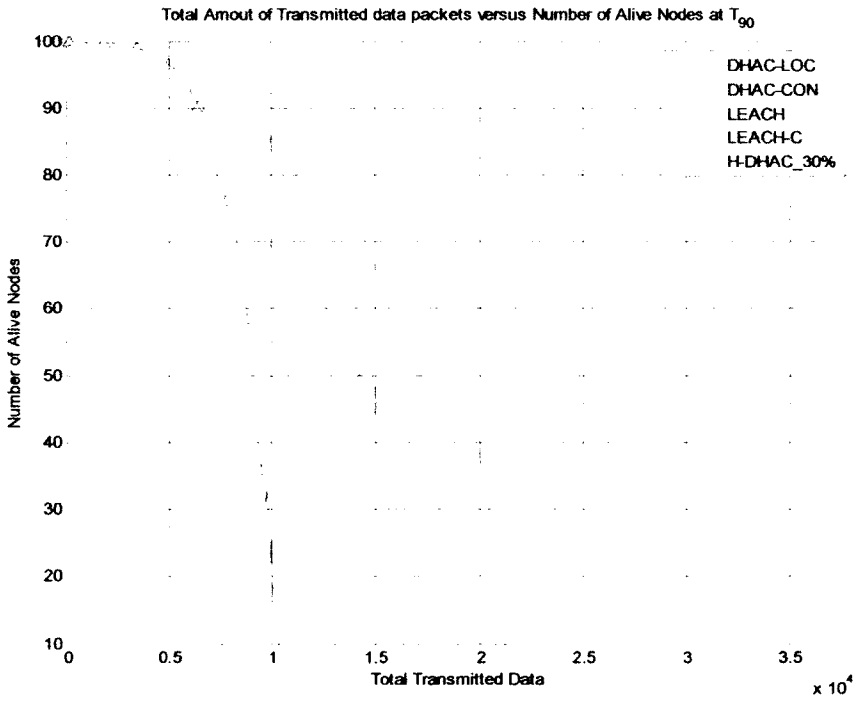
(a) H-DHAC_90%



(b) H-DHAC_70%



(c) H-DHAC_50%



(d) H-DHAC_30%

Figure 5-7 Total amount of transmitted data until T90 with BS at (150 m, 500 m)

H-DHAC_70% transmitted more packets than H-DHAC_90% and DHAC-LOC

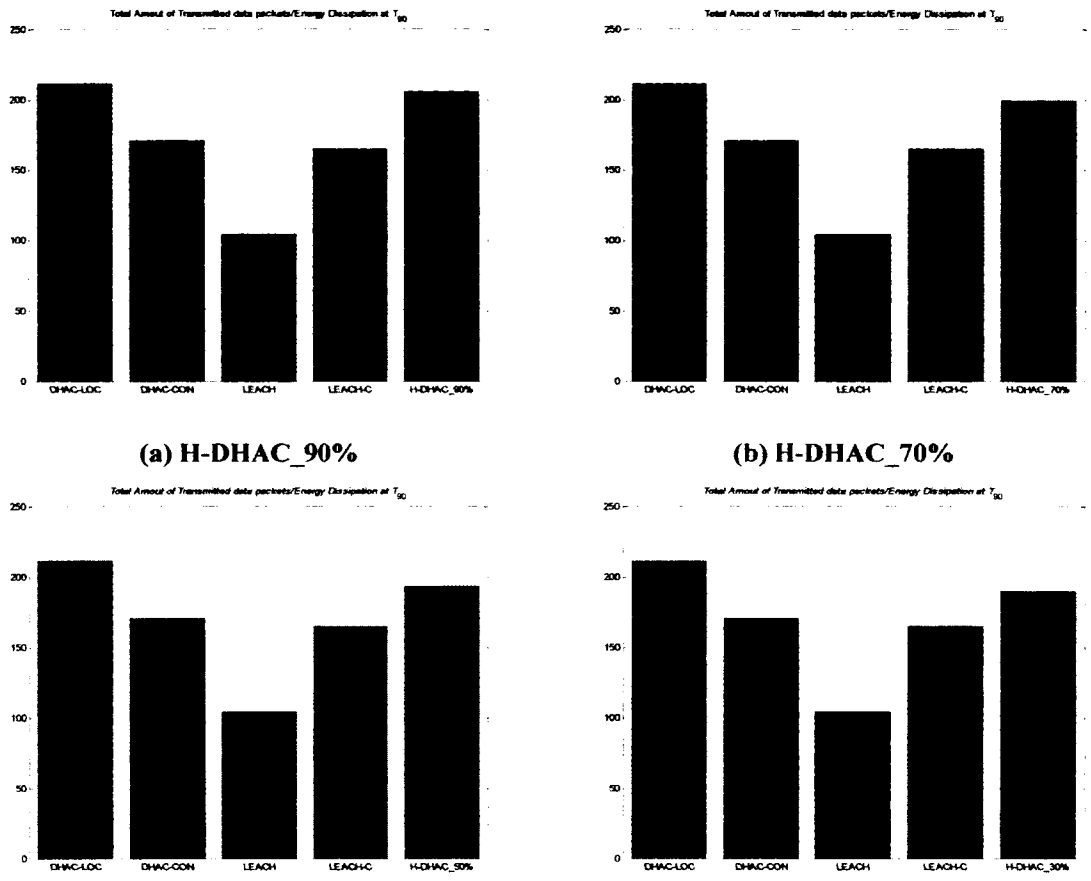
at T_1, which is possibly cause by the situation of more evenly distributed but not optimal cluster size (as it has been explained in subsection 5.2.3.1 and 5.2.3.2).

Table 5-3 Total amount of transmitted data with BS at (150 m, 500 m) (T_n denotes the time when n% sensor nodes die)

Protocol	T_1	T_{10}	T_{30}	T_{50}	T_{70}	T_{90}
LEACH	3627	6593	8361	9152	9791	10115
LEACH-C	3508	5967	12902	14801	15619	16134
DHAC-CON	2377	7530	9922	13030	16355	17298
H-DHAC_30%	5343	9294	11628	14374	18102	18835
H-DHAC_50%	6104	9691	12301	14827	18625	19322
H-DHAC_70%	6890	10211	12711	15198	19193	19859
H-DHAC_90%	6522	10507	13086	15941	19821	20463
DHAC-LOC	6777	10737	14599	17076	20481	21025

5.2.4.3 Energy Efficiency

The results of energy efficiency with BS is located at (150 m, 300 m) are shown in Figure 5-8 (a) – (d). DHAC-LOC (which works similar to H-DHAC_100%) achieves best energy efficiency among these protocols; the following rankings are: H-DHAC_90%, H-DHAC_70%, H-DHAC_50% and H-DHAC_30% rank second to fifth, and DHAC-CON, LEACH-C, LEACH occupied the last three places. Although the location of the BS is changed, H-DHAC_90% can still keep up with DHAC-LOC with only 2.39% difference. While the H-DHAC protocol with lowest GPS location data availability in our simulation — H-DHAC_30% is 87.17% more efficient than LEACH, has 14.95% higher efficiency than LEACH-C, and 11.01% better than DHAC-CON.



(a) H-DHAC_90% **(b) H-DHAC_70%**

(c) H-DHAC_50% **(d) H-DHAC_30%**

Figure 5-8 Total amount of transmitted data/energy dissipation at T_{90} with BS at (150 m, 500 m)

The simulation results indicate the consistency of H-DHAC. Even though the BS is placed farther away and the communication cost is higher, H-DHAC_90% can still achieve similar performance as DHAC-LOC. H-DHAC performs better with higher percentage of GPS location data available which suggests the stability of H-DHAC. With the help of using location data for clustering, H-DHAC outperforms DHAC-CON in all aspects.

5.2.5 Confidence Intervals

The simulation results in subsection 5.2.3 and 5.2.4 is based on the average of several runs of simulation. There are 10 runs of simulation tested for DHAC-LOC, DHAC-CON, LEACH, and LEACH-C. For 10 samples, the two-sided 95% confidence interval (CI) can be calculated by:

$$95\%CI = \mu \pm 2.262\frac{\sigma}{\sqrt{10}} \tag{5-4}$$

While H-DHAC has 100 runs of simulation for each parameter. For 100 samples, the two-sided 95% confidence interval (CI) can be derived from the following:

$$95\%CI = \mu \pm 1.984\frac{\sigma}{\sqrt{100}} \tag{5-5}$$

In equation (5-4) and (5-5), μ is the mean of the sample data; σ is the standard deviation.

In Table 5-4 and Table 5-5, the confidence intervals of network lifetime and total amount of transmitted data is presented, respectively. The network size is 300×300 m^2 with two different BS locations — (150 m, 300 m) and (150 m, 500 m).

H-DHAC with four different parameters has the smallest confidence interval, because the sample size for each parameter of H-DHAC is 10 times larger than other four protocols. With more samples and much smaller confidence interval, the simulation results of H-DHAC have better credibility than other four protocols.

Table 5-4 Confidence intervals of network lifetime at T_{90} with different BS locations

Base Station Location	Protocol	Mean (μ)	Standard Deviation (σ)	95% Confidence Interval (CI)
(150 m,300 m)	LEACH	545.5	25.65	545.5±18.35
	LEACH-C	819.5	26.82	819.5±19.18
	DHAC-CON	1028.5	52.23	1028.5±37.36
	H-DHAC 30%	1062.8	37.37	1062.8±7.42
	H-DHAC 50%	1074.4	33.36	1074.4±6.62
	H-DHAC 70%	1105.1	26.505	1105.1±5.26
	H-DHAC 90%	1122.9	27.17	1122.9±5.39
	DHAC-LOC	1128	26.16	1128±18.72
(150 m,500 m)	LEACH	396	42.41	396±30.34
	LEACH-C	568	24.63	568±17.62
	DHAC-CON	775.5	56.54	775.5±40.45
	H-DHAC 30%	801.1	42.77	801.1±8.49
	H-DHAC 50%	818.4	35.4	818.4±7.02
	H-DHAC 70%	832.3	38.25	832.3±7.59
	H-DHAC 90%	859.9	36.49	859.9±7.24
	DHAC-LOC	868	49.45	868±35.38

Table 5-5 Confidence intervals of total amount of transmitted data at T_{90} with different BS locations

Base Station Location	Protocol	Mean (μ)	Standard Deviation (σ)	95% Confidence Interval (CI)
(150 m,300 m)	LEACH	15958	1173.62	15958±839.55
	LEACH-C	28195	4270.89	28195±3055.21
	DHAC-CON	31175	2299.62	31175±1645.05
	H-DHAC 30%	34288	1496.48	34288±296.93
	H-DHAC 50%	35213	1098.47	35213±217.96
	H-DHAC 70%	36230	937.29	36230±185.98
	H-DHAC 90%	36906	952.76	36906±189.05
	DHAC-LOC	36937	1243.09	36937±889.25
(150 m,500 m)	LEACH	10115	788.81	10115±564.28
	LEACH-C	16134	1703.78	16134±1218.81
	DHAC-CON	17298	1715.25	17298±1227.01
	H-DHAC 30%	18835	1402.27	18835±278.24
	H-DHAC 50%	19322	1273.87	19322±252.76
	H-DHAC 70%	19859	1157.47	19859±229.67
	H-DHAC 90%	20463	1336.02	20463±265.10
	DHAC-LOC	21025	1307.86	24025±935.89

Chapter 6

Conclusions and Future Work

Clustering is an efficient way to organize sensor nodes in WSNs. As an interesting topic, clustering has brought many researchers' attention. In many clustering protocols, location data is used for measuring the distance dissimilarity between sensor nodes in order to classify sensor nodes into clusters. Alternatively, some studies use RSS or RSSI as distance estimator since they assume that there is no or minimum error in RSS and RSSI. However, as an easy and accurate way to obtain location data, GPS can possibly fail due to environmental effect or device quality as it has been discussed in section 4.3. Moreover, RSS or RSSI is not a good candidate to substitute GPS since they have been proven unreliable as distance estimators in practice by many studies (section 4.2).

The H-DHAC protocol is proposed in this thesis to mitigate the problems of GPS unavailability and low reliability for distance estimation using RSS or RSSI without compromising much in energy efficiency. H-DHAC adopts the mathematical concept of HAC in clustering and extending it to support hybrid data for every percentage of location data availability (0% — 100%). As a distributed approach, in H-DHAC, sensor

nodes have self-organization capability and local knowledge without involvement of centralized controller and global knowledge. To enhance robustness and minimize errors, H-DHAC introduces two novel control parameters — confidence level and C_{MIN} into clustering. The quantitative coefficient estimation schemes for four well-known HAC approach: SLINK, CLINK, UPGMA, and WPGMA are also presented.

The simulation is carried out in the topologies that are randomly generated. H-DHAC is compared with LEACH, LEACH-C, DHAC-LOC, and DHAC-CON. The performance of H-DHAC with four different percentages of GPS location data available is studied. The simulation results showed that H-DHAC outperforms LEACH, LEACH-C, and DHAC-CON in all scenarios and with minimum compromise compared to DHAC-LOC. H-DHAC does not require every sensor node to have location data and does not depend on RSS or RSSI to estimate distance. Hence H-DHAC is a more practical approach that can reduce the cost and increase the reliability of a WSN.

6.1 Future Work

There are several interesting topics that are worth to be explored in the future:

- Multiple sink/BS data gathering scheme

  The data communication between sensor nodes and the BS consume a large portion of energy. The data gathering efficiency and energy balance can be improved by having multiple BS to receive data.

- Different types of qualitative coefficients

  There are many different qualitative coefficients can be used for clustering, further investigation is needed to study the effect of different qualitative coefficients on WSNs. Also, there are smoothing techniques for qualitative data to reduce the error and noise, whether these techniques are beneficial to WSNs is worth to study as well.

- Better MAC layer scheme

  H-DHAC can integrate with MAC layer protocols that are more energy efficient to improve the performance in data gathering and energy balancing.

- Support for mobile sensor nodes

  In some application of WSNs, sensor nodes are mobile (e.g., vehicle sensors). A clustering approach with support for mobile sensor nodes can be developed based on H-DHAC.

- Effects of complicated network environments

  The network environments in practice can be complicated; there may be obstacles, shadows, asymmetrical links, etc. The effects of complicated network environments on H-DHAC are interested to be investigated.

References

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine,* vol. 40, pp. 102-114, 2002.

[2] J. Nemeroff, L. Garcia, D. Hampel, and S. DiPierro, "Application of sensor network communications," *Proc. of Military Communications Conference. (MILCOM 2001).* McLean, VA, USA, pp. 336-341, 2001.

[3] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole, "Research challenges in environmental observation and forecasting systems," *Proc. of the 6th International Conference on Mobile Computing and Networking*, Boston, Massachusetts, United States, pp. 292-299, 2000.

[4] A. N. Knaian, "A wireless sensor network for smart roadbeds and intelligent transportation systems," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, USA, 2000.

[5] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," *Proc. of the 7th International Conference on Mobile Computing and Networking*, Rome, Italy, pp. 151-165, 2001.

[6] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications,* vol. 11, pp. 6-28, 2004.

[7] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks,* vol. 3, pp. 325-349, 2005.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications,* vol. 1, pp. 660-670, 2002.

[9] P. Ding, J. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," *Proc. of the First IEEE International Conference on Distributed Computing in Sensor Systems*, Marina del Rey, CA, pp.

322-339, 2005.

[10] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, pp. S8-13, 2005.

[11] Y. Mao, L. Chengfa, C. Guihai, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks," *Proc. of the 24th IEEE International Performance, Computing, and Communications Conference*, pp. 535-540, 2005.

[12] L. Chengfa, Y. Mao, C. Guihai, and W. Jie, "An energy-efficient unequal clustering mechanism for wireless sensor networks," *Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, pp. 8 pp.-604, 2005.

[13] C. H. Lung and C. Zhou, "Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach," *Ad Hoc Networks*, vol. 8, pp. 328-344, 2010.

[14] C. Zhou, "Application and Evaluation of Hierarchical Agglomerative Clustering in Wireless Sensor Networks," MASc Thesis, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 2008.

[15] C. Romesburg, *Cluster analysis for researchers*: Lulu. com, 2004.

[16] R. Nowak and U. Mitra, *Boundary Estimation in Sensor Networks: Theory and Methods Information Processing in Sensor Networks* vol. 2634: Springer Berlin / Heidelberg, 2003.

[17] M. A. M. Vieira, C. N. Coelho, Jr., D. C. da Silva, Jr., and J. M. da Mata, "Survey on wireless sensor network devices," *Proc. of IEEE Conference on Emerging Technologies and Factory Automation*, pp. 537-544 vol.1, 2003.

[18] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "Smart Dust"," *Proc. of the 5th ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, United States, pp. 271-278, 1999.

[19] J. M. Rabaey, M. J. Ammer, J. L. d. Silva, D. Patel, and S. Roundy, "PicoRadio

Supports Ad Hoc Ultra-Low Power Wireless Networking," *Computer,* vol. 33, pp. 42-48, 2000.

[20] P. Sung, I. Locher, A. Savvides, M. B. Srivastava, A. Chen, R. Muntz, and S. Yuen, "Design of a wearable sensor badge for smart kindergarten," *Proc. of the 6th International Symposium on Wearable Computers*, pp. 231-238, 2002.

[21] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *Proc. of the 7th International Conference on Mobile Computing and Networking*, Rome, Italy, pp. 272-287, 2001.

[22] G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, W. J. Kaiser, and H. O. Marcy, "Wireless integrated network sensors: Low power systems on a chip," *Proc. of the 24th European Solid-State Circuits Conference*, pp. 9-16, 1998.

[23] I. Kurtis Kredo and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks,* vol. 51, pp. 961-994, 2007.

[24] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine,* vol. 44, pp. 115-121, 2006.

[25] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *Proc. of the 21st Joint Conference of the IEEE Computer and Communications Societies*, pp. 1567-1576 vol.3, 2002.

[26] L. F. W. Van Hoesel and P. J. M. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," *Proc. of the 1st International Workshop on Networked Sensing Systems (INSS)*, Tokyo, Japan, pp. 205-208, 2004.

[27] M. I. Brownfield, K. Mehrjoo, A. S. Fayez, and N. J. I. Davis, "Wireless sensor network energy-adaptive mac protocol," *Proc. of the 3rd IEEE Consumer Communications and Networking Conference*, pp. 778-782, 2006.

[28] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," *Proc. of the 5th*

ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, Washington, United States, pp. 174-185, 1999.

[29] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proc. of the 6th International Conference on Mobile Computing and Networking*, Boston, Massachusetts, United States, pp. 56-67, 2000.

[30] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366-379, 2004.

[31] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," *Proc. of IEEE Aerospace Conference*, Big Sky, Montana, USA, pp. 1125-1130, 2002.

[32] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 924-935, 2002.

[33] Y. Jiang, C. H. Lung, and N. Goel, "A Tree-Based Multiple-Hop Clustering Protocol for Wireless Sensor Networks," *Proc. of the 2nd International Conference on Ad Hoc Networks (ADHOCNETS)*, Victoria, BC, Canada, pp. 371-383, 2010.

[34] Y. Yaoyao, S. Juwei, L. Yinong, and Z. Ping, "Cluster Head Selection Using Analytical Hierarchy Process for Wireless Sensor Networks," *Proc. of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, Helsinki, Finland, pp. 1-5, 2006.

[35] D. C. Hoang, R. Kumar, and S. K. Panda, "Fuzzy C-Means clustering protocol for Wireless Sensor Networks," *Proc. of IEEE International Symposium on Industrial Electronics (ISIE)*, Bari, Italy, pp. 3477-3482, 2010.

[36] B. S. Everitt, S. Landau, M. Leese, and D. D. Stahl, *Cluster analysis*, 5 ed.: WILEY, 2010.

[37] X. Rui and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, pp. 645-678, 2005.

[38] S. S. Choi, S. H. Cha, and C. Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal on Systemics, Cybernetics and Informatics,* vol. 8, pp. 43-48, 2010.

[39] D. Deb, S. B. Roy, and N. Chaki, "LACBER: A new location aided routing protocol for GPS scarce MANET," *International Journal of Wireless & Mobile Networks (IJWMN),* vol. 1, pp. 22–36, 2009.

[40] K. Heurtefeux and F. Valois, "Is RSSI a Good Choice for Localization in Wireless Sensor Network?," *Proc. of the IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)*, Fukuoka, Japan, pp. 732-739, 2012.

[41] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is RSSI a Reliable Parameter in Sensor Localization Algorithms: An Experimental Study," *Proc. of Field Failure Data Analysis Workshop*, Niagara Falls, USA, pp. 471-478, 2009.

[42] M. Hosseini, H. Chizari, S. Chai Kok, and R. Budiarto, "RSS-based distance measurement in Underwater Acoustic Sensor Networks: An application of the Lambert W function," *Proc. of the 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1-4, 2010.

[43] N. Patwari and I. Alfred O. Hero, "Using proximity and quantized RSS for sensor localization in wireless networks," *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, San Diego, CA, USA, pp. 20-29, 2003.

[44] S. Fazackerley, A. Paeth, and R. Lawrence, "Cluster head selection using RF signal strength," *Proc. of Canadian Conference on Electrical and Computer Engineering.*, pp. 334-338, 2009.

[45] D. Lymberopoulos, Q. Lindsey, and A. Savvides, "An Empirical Characterization of Radio Signal Strength Variability in 3-D IEEE 802.15.4 Networks Using Monopole Antennas Wireless Sensor Networks." vol. 3868, K. Römer, H. Karl, and F. Mattern, Eds., ed: Springer Berlin / Heidelberg, pp. 326-341, 2006.

[46] W. Rong-Hou, L. Yang-Han, T. Hsien-Wei, J. Yih-Guang, and C. Ming-Hsueh, "Study of characteristics of RSSI signal," *Proc. of IEEE International Conference on Industrial Technology*, Chengdu, China, pp. 1-3, 2008.

[56] E. S. Nadimi, H. T. Søgaard, T. Bak, and F. W. Oudshoorn, "ZigBee-based wireless sensor networks for monitoring animal presence and pasture time in a strip of new grass," *Computers and Electronics in Agriculture,* vol. 61, pp. 79-87, 2008.

[57] M. Moghaddam, D. Entekhabi, Y. Goykhman, L. Ke, L. Mingyan, A. Mahajan, A. Nayyar, D. Shuman, and D. Teneketzis, "A Wireless Soil Moisture Smart Sensor Web Using Physics-Based Optimal Control: Concept and Initial Demonstrations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 3, pp. 522-535, 2010.

[58] A. Dempster, "How Vulnerable is GPS?," *Position,* vol. 20, pp. 64-67, 2005.

[59] D. Hoey and P. Benshoof, "Civil GPS Systems and Potential Vulnerabilities," *Proc. of the 18th International Technical Meeting of the Satellite Division*, Long Beach, USA, pp. 1291 - 1295, 2005.

[60] U. I. Bhatti and W. Y. Ochieng, "Failure Modes and Models for Integrated GPS/INS Systems," *The Journal of Navigation,* vol. 60, pp. 327-348, 2007.

[61] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," PhD Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA, 2000.

[62] A. Förster, A. Förster, and A. L. Murphy, "Optimal Cluster Sizes for Wireless Sensor Networks: An Experimental Analysis," *Ad Hoc Networks,* vol. 28, pp. 49-63, 2010.

[63] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless Sensor Network Simulators: A Survey and Comparisons," *International Journal of Computer Networks (IJCN),* vol. 2, pp. 249-265, 2011.