

Application and Evaluation of Hierarchical Agglomerative Clustering in Wireless Sensor Networks

by
Chenjuan Zhou

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

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

January 2008



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 978-0-494-36838-1

Our file *Notre référence*

ISBN: 978-0-494-36838-1

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

The undersigned hereby recommends to
the Faculty of Graduate Studies and Research
Acceptance of thesis

**Application and Evaluation of Hierarchical Agglomerative
Clustering in Wireless Sensor Networks**

Submitted by Chenjuan Zhou
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
January, 2008

Abstract

In Wireless Sensor Networks (WSNs), hierarchical network structures have the advantage of providing scalable and resource efficient solutions. Thus, finding an optimal way to generate clusters is an important topic in WSNs. To achieve this goal, this master's thesis improves the Hierarchical Agglomerative Clustering (HAC) algorithm by proposing a Distributed the HAC (DHAC) algorithm. With simple six-step clustering, DHAC provides a bottom-up clustering approach by grouping similar nodes together before the Cluster Head (CH) is selected. DHAC can accommodate both quantitative and qualitative information types in clustering, while offering flexible combinations using four well-studied the HAC algorithm methods, SLINK, CLINK, UPGMA and WPGMA. With automatic CH rotation and rescheduling, DHAC avoids reclustering and achieves uniform energy dissipation through the whole network. Simulation results in the NS2 platform demonstrate the longer network lifetime of the DHAC than the better-known clustering protocols, LEACH and LEACH-C.

Acknowledgments

I would like to express my deep appreciation to my supervisor, Professor Chung-Horng Lung for his patience, encouragement, valuable guidance for my research. His creativity and passion for science will always serve as an influential example for me.

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

I am indebted to my grandaunt Nancy and her family: their support and understanding helps me complete this work.

Friends helped me in many ways during my study. I thank them for all the help and encouragement. Special thanks to Haoming for the support and care he gave to me.

To my father, Zhanyang, my beloved mother, Jufen, and the family of my brother, Kun: their love always encourages me to get over my depression. Uncountable are the sacrifices they made for me through the years. My love and respect to them are endless and immense.

Contents

1	Introduction	1
1.1	Hierarchical Routing Protocol Design in WSNs	2
1.2	Motivation and Thesis Contributions	5
1.3	Organization of Thesis	7
2	Background and Related Works	8
2.1	Wireless Sensor Networks	8
2.1.1	Components and Characteristics	9
2.1.2	Hardware	12
2.1.3	MAC Layer Protocols	15
2.1.4	Network Routing Protocols in WSNs	17
2.2	Related Works on Hierarchical Protocols	23
2.2.1	Random-selected-CH Protocols	24
2.2.2	Well-selected-CH Protocols	29
3	Hierarchical Agglomerative Clustering (HAC) Algorithm and its Ap- plication to WSNs	34
3.1	Basic Concepts of the HAC Algorithm	34
3.1.1	Obtain the Input Data Set	35

3.1.2	Computation of Resemblance Coefficients	37
3.1.3	Execution of the HAC Algorithm Methods	40
3.2	Cluster Formation for WSNs	44
3.2.1	Application of the HAC with Quantitative Data	45
3.2.2	Application of the HAC with Qualitative Data	47
4	Distributed Hierarchical Agglomerative Clustering (DHAC) Algorithm	50
4.1	General Assumptions	51
4.2	DHAC Procedure	51
4.3	DHAC: Cluster Formation	55
4.3.1	Cluster Formation with Quantitative Data	55
4.3.2	Cluster Formation with Qualitative Data	59
4.4	DHAC: Cluster Maintenance	61
4.4.1	Scheduling and Data Aggregation	61
4.4.2	CH Rotation	62
4.5	Evaluation of Schemes in DHAC	65
4.5.1	Evaluation of Cluster Formation	66
4.5.2	Evaluation of Cluster Maintenance	72
5	Performance Evaluation and Simulation Experiments	74
5.1	Simulation Environment	74
5.2	Experiments and Simulation Results	76
5.2.1	Sink Inside of the Network	78
5.2.2	Sink Outside of the Network	82
5.2.3	Different Sink Locations	85

5.2.4	Different Data Rates	89
5.2.5	Different Node Initial Energy	90
6	Conclusions and Future Work	92
6.1	Future Work	93
	References	94

List of Tables

2.1	Power management of two sensor platforms	15
2.2	A simple taxonomy of network routing protocols	20
3.1	Node location data matrix for the 8-node network	36
3.2	One-hop network connectivity data matrix for the 8-node network . .	37
3.3	Initial resemblance matrix with quantitative data	38
3.4	Resemblance matrix with qualitative data using SORENSON dissimilarity coefficients	39
4.1	Initial distributed resemblance matrices with quantitative data	56
4.2	The first step of DHAC: updated resemblance matrices using SLINK with quantitative data	57
4.3	The second step of DHAC: updated resemblance matrices using SLINK with quantitative data	59
5.1	Simulation parameters	75
5.2	The time of 1%, 20%, 50%, 80% and 100% node dead with different E_{initial}	91

List of Figures

2.1	A typical WSN.	9
2.2	Sensor node platform architecture	12
2.3	Two commercial sensor products	14
2.4	Two typical WSN structures	21
3.1	A simple 8-node network	36
3.2	The first step of the HAC: using SLINK with quantitative data	42
3.3	The second step of the HAC: using SLINK with quantitative data	42
3.4	The third step of the HAC: using SLINK with quantitative data	43
3.5	The fourth step of the HAC: using SLINK with quantitative data	43
3.6	The fifth step of the HAC: using SLINK with quantitative data	43
3.7	Dendrogram using different the HAC algorithm methods with quantitative data	44
3.8	Clustering steps and dendrogram using UPGMA with quantitative data	46
3.9	Clustering steps and dendrogram using UPGMA with SORENSON qualitative data	47
4.1	Pseudo code of DHAC in WSNs	53
4.2	The first step of DHAC: using SLINK with quantitative data	57
4.3	The second step of DHAC: using SLINK with quantitative data	58

4.4	Comparison of the HAC and DHAC using UPGMA with SORENSON qualitative data	60
4.5	The MAC layer structure of DHAC	62
4.6	An illumination: automatic CH rotation	63
4.7	Variation points and possible variants of applying DHAC to WSNs .	66
4.8	DHAC using UPGMA for 100-node network with different data types	67
4.9	Number of clusters versus node degree using different DHAC methods and quantitative location data	68
4.10	Maximum cluster range versus node degree using different DHAC methods and quantitative location data	69
4.11	Maximum cluster range versus node degree using quantitative and qualitative data	70
4.12	Cluster range versus node degree with different resemblance calculation methods	71
4.13	Network lifetime of the first node dead versus $P(\text{energy})(\%)$	72
5.1	Network lifetime, the sink at (50,50)	78
5.2	Number of nodes alive versus the amount of data packets received at the sink(50,50)	80
5.3	Total clustering energy dissipation per minute, the sink at (50,50) . .	81
5.4	Total energy dissipation versus the amount of data packets received to the sink(50,50)	82
5.5	Network lifetime, the sink at (50,300)	83
5.6	Number of nodes alive versus the amount of data packets received at the sink(50,300)	84

5.7	Total energy dissipation versus the amount of data packets received to the sink(50,300)	84
5.8	Time of the 20% nodes dead against the sink location	86
5.9	Time of the 50% nodes dead against the sink location	86
5.10	Time of the 80% nodes dead against the sink location	87
5.11	Time of the last nodes dead against the sink location	88
5.12	Time of the last node dead with different data rate	89

List of Acronyms

μ AMPS	micro-Adaptive Multi-domain Power-aware Sensors
AHP	Analytical Hierarchy Process
C-HEIGHT	The resemblance coefficient of two merged clusters
C-NUMBER	Number of clusters
CH	Cluster Head
CSMA	Carrier Sense Multiple Access
DD	Directed Diffusion protocol
DHAC	Distributed Hierarchical Agglomerative Clustering
DHAC-CON	DHAC with CONnectivity qualitative data
DHAC-LOC	DHAC with LOCation quantitative data
DHAC-RSS	DHAC with RSS quantitative data
ERA	Energy Residue Aware
G-MAC	Gateway-MAC
GPS	Global Positioning System
HAC	Hierarchical Agglomerative Clustering
HEED	Hybrid Energy-Efficient Distributed clustering
LEACH	Low-Energy Adaptive Clustering Hierarchy
LEACH-C	LEACH-Centralized
LMAC	Lightweight MAC

LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Medium-Access Control
PEGASIS	Power-Efficient GATHERing in Sensor Information Systems
RRCH	Round-Robin Cluster Header
RSS	Received Signal Strength
S-MAC	Sensor MAC
SPIN	Sensor Protocols for Information via Negotiation
TDMA	Time-Division Multiple Access
WINS	Wireless integrated network sensors
WSN	Wireless Sensor Network

List of Symbols

<u>Symbol</u>	<u>Explanation</u>	<u>page where symbol is defined</u>
α	confidence level	89
δ	two-sided confidence interval	89
ϵ_{fs}	Amplifier energy with free space channel model	76
ϵ_{mp}	Amplifier energy with multipath fading channel model	76
$\hat{\sigma}$	estimated standard deviation of the population	89
ResM	Resemblance matrix	52
C_{ab}	Coefficient of objects, a and b	38
C_{CLINK}	Updated coefficient by using CLINK method	40
C_{size}	The cluster_member number in a cluster	52
C_{SLINK}	Updated coefficient by using SLINK method	40
C_{UPGMA}	Updated coefficient by using UPGMA method	41
C_{WPGMA}	Updated coefficient by using WPGMA method	41
D_{ab}	Distance between nodes, a and b	38
E_{CH-rem}	Residual energy of CH	27
E_{com}	Computation energy dissipation	76

E_{fusion}	Computation energy dissipation per bit	76
E_{initial}	Average initial energy(J) of each node	64
$E_{\text{nonCH-rem}}$	Residual energy of non-CH node	27
E_{Rx}	Receiving energy dissipation	75
E_{toBS}	Communication energy from CH to the sink	27
E_{toCH}	Communication energy from cluster member to CH	27
E_{Tx}	Transmitting energy dissipation	76
ID_{CH}	The node ID of Cluster Head	52
ID_{Node}	Sensor node ID	52
M_{CH}	The cluster ID corresponding the Minimum_coefficient in a resemblance matrix	52
M_{Coeff}	The Minimum_coefficient in the resemblance matrix	52
N_s	number of samples	89
P	Predefined percentage of CHs among all nodes	24
$P(\text{energy})$	Predefined percentage of average residual energy	65
$R(\text{energy})_i$	Average residual energy of node i	65
$T(\text{seconds})$	Time period of automatic CH rotation	64
$Th(\text{energy})$	Threshold of rescheduling	65
$Th(\text{LEACH})$	Threshold in LEACH protocol	24
v	freedom of t -distribution	89

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) became an invaluable research area because they build a connection between the world of nature and that of computation by digitizing certain useful information. Research on WSNs was initially driven by military applications such as battlefield surveillance, enemy tracking and other risk-associated areas. Now WSNs are used in many civilian applications, including environment observation [1], habitat monitoring [2–4], health care [5] and traffic control [6].

The wide range of WSN applications are based on the collaborative efforts of a large number of small-size, low-cost, randomly deployed and unattended sensor nodes [7]. The design of a WSN is rather different from a conventional wireless network due to the following constraints: (i) Sensor nodes are often deployed in unreachable harsh environments. The nodes may fail, causing communication failures and consequent network topology changes. (ii) Data rates in WSNs are much lower than the multimedia traffic in conventional wireless networks. (iii) If WSNs contain a large number of nodes, they need to be randomly deployed to reduce cost. This, however, might cause unbalanced coverage of the sensing area - one section may be covered by hundreds of sensor nodes while another is only covered by a few. (iv) In

contrast to typical communication networks, almost all applications of sensor networks have highly asymmetric communication links; the collected data is mostly sent up to a particular node which is called a sink. (v) As a consequence of the above constraints, sensor nodes are often required to have ultra low power consumption.

How can numerous randomly deployed sensor nodes behave in a collaborative and organized way? This is a fundamental issue in network management and information extraction in WSNs. Each sensor node wants to maximize its own utility function. However, the entire network needs balance in resource assignments to perform in a way that is useful. Network routing protocol design becomes an important research area far more critical to WSNs performance than that of from conventional communication networks. Among numerous proposed network routing protocols in past years, hierarchical routing protocols greatly contribute to system scalability, lifetime, and energy efficiency [8].

1.1 Hierarchical Routing Protocol Design in WSNs

Based on our understanding of WSN's unique characteristics, we realize that efficiently organizing the collaborating sensor nodes has always been the most important issue in the whole development history of WSNs. Much more attention has been given to hierarchical protocols than other types of protocols. The hierarchical network architecture has been successfully used to provide scalable solutions in many large networks.

Through the years, many hierarchical routing protocols have been proposed with different design goals, clustering criteria and basic assumptions. There are some basic challenges to hierarchical routing protocol design in WSNs. Addressing those problems may require cross-layer design, especially with regard to the Medium-Access

Control (MAC) and network layers [9].

- **Inter-cluster communication.** Two kinds of schemes are adopted for inter-cluster communication: fixed transmission range and variable transmission range. Fixed transmission range forces Cluster Heads (CHs) communicate each other via gateway nodes, which are chosen from cluster members located on or close to edges of clusters. Variable transmission range allows CHs to directly communicate with each other by adjusting transmission power.
- **Intra-cluster communication.** Within a cluster, all the cluster members need to communicate with their CH. Cluster members can connect directly (one-hop) or via a transmission chain (multihop). Reducing resource consumption by providing good quality clusters is still an open issue. Moreover, a hierarchical routing protocol must consider the performance of both intra-cluster communication and inter-cluster communication to ensure sufficient network connectivity.
- **MAC layer design.** As the intermediate layer between physical layer and network layer, the MAC layer plays an important role to support the network layer. MAC layer protocol has significant influence on energy efficiency because it directly controls antenna activity. MAC layer design has the main challenge of collision avoidance.
- **CH selection scheme.** CH selection must give attention to both intra-cluster and inter-cluster communications. Some hierarchical protocols like LEACH [10, 11] uniformly pick a CH from all sensor nodes. Some other protocols, such as LEACH-C [11], use optimal algorithms to carefully choose an appropriate CH based on certain parameters, including the distance between nodes, the number of node neighbors or the node residual energy.

- **CH rotation scheme.** It is necessary to rotate the CH role through the entire network because its responsibilities require higher energy consumption. This can be implemented by reclustering or by passing the CH role to a backup node. The choice of when to rotate the CH is also an issue. Two frequently used strategies are periodic rotation and threshold. Periodic rotation requires precise clock synchronization between nodes, whereas the threshold strategy needs to collect related information from nodes such as the node residual energy, and rotate when a predefined threshold exceeded.
- **Overhead.** The overhead of a hierarchical protocol is cluster formation energy, i.e The energy, consumed during clustering by message transceiving and message processing. Usually, a centralized clustering protocol provides well-organized clusters by exchanging more information between the sink and the nodes. requiring the sink to execute more complex algorithms. A distributed clustering protocol has simpler schemes and less message exchange, but usually requires more frequent reclustering, which is also considered as overhead.

Besides the above challenges, protocols must also consider sensing coverage and redundant data [9]. Chapter 2 will present basic concepts of hierarchical protocols and discuss typical approaches with more details.

1.2 Motivation and Thesis Contributions

To our best knowledge, all current clustering protocols are top-down approaches, which first formulate an overview of a WSN system, specifying but not detailing the first-level nodes. Based on the overview of network knowledge and predefined methods, the protocols first build the upper level of clusters by selecting certain nodes as CHs. Then they set the rest of the nodes group into the designated cluster as cluster members.

The motivation of this thesis is to provide optimal clustering without requiring the network overview by reversing the clustering approach from top-down to bottom-up. With the bottom-up approach, sensor nodes collaborate and build clusters before they select CHs. Thus, the individual nodes of the system are first specified in great detail. In this manner, the bottom-up approach can be a better way to implement self-organization, scalability and flexibility. To gain the benefits of bottom-up hierarchical clustering, we focus on how to classify sensor nodes into appropriate groups. By carefully inspecting the relationships among nodes, we develop a simple and flexible distributed hierarchical clustering scheme to suit different application scenarios and thereby provide better performance with the design factors presented in Section 1.1.

This thesis tailors the Hierarchical Agglomerative Clustering(HAC) [12–14] algorithm for WSNs to generate optimized hierarchical clusters. Our research was inspired by the fact that the HAC has been successfully applied to many disciplines, and it is a conceptually and mathematically simple clustering approach. Additionally, the HAC algorithm offers flexible options within its basic six-step clustering. In this thesis, we improve the HAC to a distributed algorithm, DHAC (Distributed Hierarchical Agglomerative Clustering). Main contributions of this thesis are listed as follows.

- As a distributed clustering protocol, DHAC removes the limitation from the

sink.

- DHAC provides a bottom-up clustering approach by characterizing individual nodes in great detail without requiring global knowledge of the WSNs. In DHAC, nodes merge together by obeying certain rules, and CHs are selected after the cluster is formed. Therefore, DHAC avoids bad cluster quality, which is a typical defect of random-selected-CH protocols (Section 2.2.1).
- DHAC performs clustering only once, at the initial stage. DHAC can avoid the time and energy dissipation during reclustering.
- DHAC offers more flexible CH rotation to reduce the cost of clustering maintenance. DHAC includes an automatic CH rotation and rescheduling scheme to ensure uniform energy dissipation within clusters.
- DHAC supports various clustering algorithms, including SLINK, CLINK, WPGMA and UPGMA, to suit different applications.
- DHAC can be executed with or without knowledge of location (quantitative data and qualitative data).

Parts of this thesis have appeared in the following publications:

- C.-H. Lung, C. Zhou, and Y. Yang, “Applying Hierarchical Agglomerative Clustering to Wireless Sensor network”, International Workshop on Theoretical and Algorithmic Aspects of Sensor and Ad-hoc Networks (WTASA’07), June 2007, pp 97-105.
- C. Zhou and C.-H. Lung, “Application and Evaluation of Hierarchical Agglomerative Clustering in Wireless Sensor Networks”, to appear in Sensor and Ad-

Hoc Networks: Theoretical and Algorithmic Aspects, Makki, et. al. (editors), Springer.

1.3 Organization of Thesis

The remainder of the thesis is organized as follows.

Chapter 2 provides a background knowledge of WSNs, and reviews previous hierarchical clustering protocols in WSNs.

Chapter 3 introduces the preliminaries of the HAC algorithm by applying it in WSNs with different node attributes.

Chapter 4 presents our proposed DHAC scheme. We introduce different the HAC algorithm methods and present their analysis through applying them into WSNs. We also investigate different properties of these algorithm methods.

Chapter 5 offers the simulation results and related discussions by comparing the HAC algorithm with the better-known hierarchical routing protocols, LEACH and LEACH-C. The DHAC shows the ability to substantially improve the energy efficiency and routing performance under various WSN applications.

Chapter 6, we provide a summary and suggest some directions for future work.

Chapter 2

Background and Related Works

This chapter introduces basic concepts of WSNs and important network routing protocols, especially the emphasizing information on hierarchical clustering protocols in WSNs.

2.1 Wireless Sensor Networks

In recent years, digital communication has been improved quickly. Advances in wireless technologies, computation speed, memory and needs of mobile communication have encouraged wireless products to support various applications. Smaller and cheaper communication products provide more flexible wireless network deployment. The fast development contributes to a similarly fast-moving research field which is turning from centralized, supervised, single-node platforms to a multitude of distributed, autonomous, dense, low cost and lightweight intelligent nodes. Riding the tide of the rapid development, WSN has been an area of active research since the early 1990s. Numerous workshops and conferences related to WSNs are arranged each year in computer science and telecommunications.

2.1.1 Components and Characteristics

In WSNs, sensor nodes continuously measure desired information from their respective surroundings. Sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data instead of sending out the raw data.

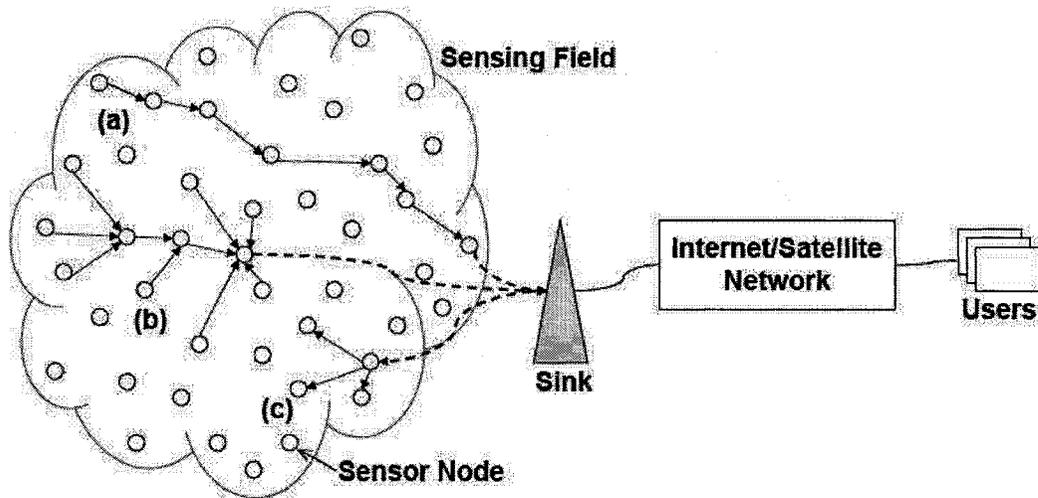


Fig. 2.1: A typical WSN.

Fig. 2.1 shows a typical WSN. There are a large number of homogeneous sensor nodes within the sensing field. WSNs can be deployed indoors, in an inaccessible area, or even on the human body. The sensor nodes collect data from their specific sensing environments. The processed data are transmitted through interconnected sensor nodes, which means each sensor node at least has one route to the sink. In the traditional wireless network, nodes are accurately deployed and a query usually will produce a response from one corresponding device. Contrarily, sensor nodes are randomly deployed, and the same phenomenon can be detected by several sensor nodes due to WSN node density that may reach up to 20 nodes/m^3 [15].

All the information gathered from the sensor network is integrated by the sink. The sink is distinguished from other components in the WSNs by having far more computational, energy and communication resources. It acts as a gateway between the sensor nodes and other networks. After collection, the sink forwards data through traditional networks and finally to the end user. We should notice that the sink is not identical with the Base Station (BS) in traditional networks because a sink is not in charge of network management as would be a BS. This special node is named “sink” because it acts as a data aggregate. The sink may be embedded within the network, in indoor applications, or it may be located on the network edge or outside the network. In some cases the sink can be mobile, carried by a plane or vehicle. A moving sink passes through the entire network to access each sensor node. Different applications place the sink using different strategies. Some application designs may include more than one sink. In most applications, there is only one sink, located far from the network area because the network is deployed in an inaccessible environment.

Although WSNs have a wide application area, different scenarios can be simply classified into three traffic patterns as shown in Fig. 2.1. Pattern (a) shows one-to-one traffic: the triggered sensor node sends data to the sink by obtaining the help of other sensor nodes. Pattern (b) is many-to-one traffic: sensor nodes aggregate and send data to the sink. And, the pattern (c) shows one-to-many traffic which is used when the sink sends commands or queries to specific sensor nodes. According to the traffic patterns, the applications of WSNs can be classified as three main types, query-driven application, event-driven application and continuous data generation. The query-driven application happens when a sink sends out the queries, and the relevant sensor nodes report information back to the sink. With this kind of application, protocols need to handle both one-to-many and one-to-one traffic patterns. In an

event-driven application, sensor nodes send information to the sink upon detecting an event. This is the many-to-one traffic since the event may be detected by several sensor nodes. Continuous data generation type means that sensor nodes periodically send information to the sink in turn. Obviously, this is possesses the many-to-one traffic.

Low cost is a marked characteristic of WSNs. For a network, there are three costs that should be taken into account: device cost, deployment cost and maintenance expense. Firstly, the cost of a sensor node should be much less than 1 U.S. dollar in order for the sensor network to be feasible [7]. The cheap price limits the capability of sensor nodes, for example, small memory, low computation speed and less battery capacity. As presented above, the hardware limitations are continuously released because of the rapid advances in fabrication techniques. Thus, the cost of hardware will not be an issue in the future. On the other hand, the expense of network deployment and maintenance cannot be ignored. Obviously, careful deployment and periodic maintenance need a huge expense. As an extreme case, maintaining a network in a remote, harsh area may require prohibitive cost. To decrease the expense, sensor nodes must be randomly deployed in the field, and cannot be repaired, or even recharged; further, they must self-organize to achieve complex tasks.

Because of this necessity for low cost and self-organization, more constraints are added in WSNs than in any other networks. WSNs have to face the issues of optimizing resource assignment and network topology control because they cannot have global level network information without centralized management. The large scale of the networks brings a big challenge of sharing a severely limited bandwidth. The sensor nodes of WSNs are very limited in energy, so the network cannot afford to use the otherwise attractive solutions of complex algorithms and schemes.

Conventional design techniques and hardware architectures only provide solutions which are insufficient for these extremely constrained systems. To resolve the issues in WSNs, many different prototypes of sensor nodes are being designed and manufactured. Energy efficient MAC protocols, topology control protocols and routing schemes are being evaluated, and many enabling technologies such as time synchronization and localization are being studied. The severe requirements necessitate designing the specific techniques at all layers of the protocol stack in WSNs.

2.1.2 Hardware

The decrease in the size and cost of sensor nodes has encouraged the possible use of a large set of disposable unattended sensor nodes. Sensor nodes are densely deployed either inside the phenomenon to be studied or very close to it. These nodes are required to be intelligent and of low energy consumption.

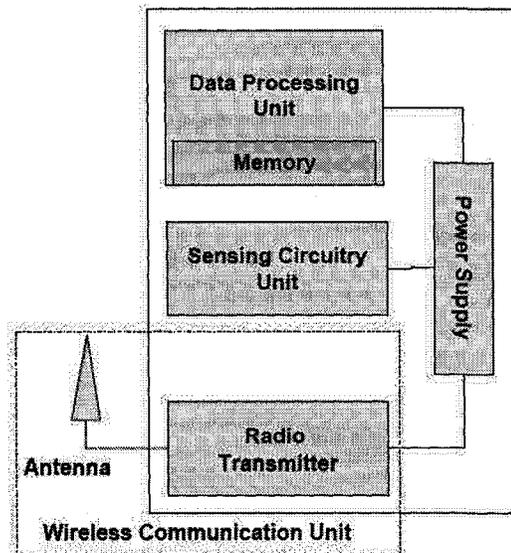


Fig. 2.2: Sensor node platform architecture

Fig. 2.2 shows the typical sensor node system architecture. Generally, a sensor

node is equipped with four functional units: sensing circuitry unit, data processing unit, wireless communication unit and power supply, usually a battery [16]. The sensing circuitry transforms the collected information into electric signals that measure ambient conditions, such as temperature, sound, light, vibration, pressure, motion or pollutants. The data processing unit uses the signals to reveal the required attributes about objects or events. The wireless communication unit is in charge of sending or forwarding useful data out of network, usually via a radio transmitter. Currently, a battery is used as a power supply in many platforms; a few nodes use solar cells as solutions in specific application scenarios. In addition to these four basic functional units, sensor nodes may add other features to achieve the tasks of an application. For instance, GPS (Global Positioning System) is added to provide precise location data in many applications.

Each unit faces its peculiar design challenges under extreme constraints of cost, size and energy. For each sensing unit, how to correctly catch information and describe it becomes a perpetual target for improvement. The data processing unit and memory are easily overburdened by the collected data size and algorithm complexity. The communication unit needs a highly efficient protocol to improve the power management. The whole node must be cheap enough to be discarded since battery replacement is not an option for many WSNs. The batteries need to power a sensor node for months, even years.

There are many projects seeking ways to integrate the sensing, processor, and communication kit onto a single board. The SmartDust [17,18] is a sensor node product from UC Berkeley based on a project which began in 1998 and aiming at the development of miniature sensor nodes as small as a piece of dust or a grain of sand. Other projects have started with different emphases, such as WINS (Wireless Inte-

grated Network Sensors) [19], μ AMPS (micro-Adaptive Multi-domain Power-aware Sensors) [15], and PicoRadio [20]. Based on the diverse research achievements, there are more and more mature sensor products coming into the market. There are many choices of processors, memory, sensors, radio techniques and power supplies. These flexible assemblies mean the sensor market has huge variety in terms of capability, cost and size [16].

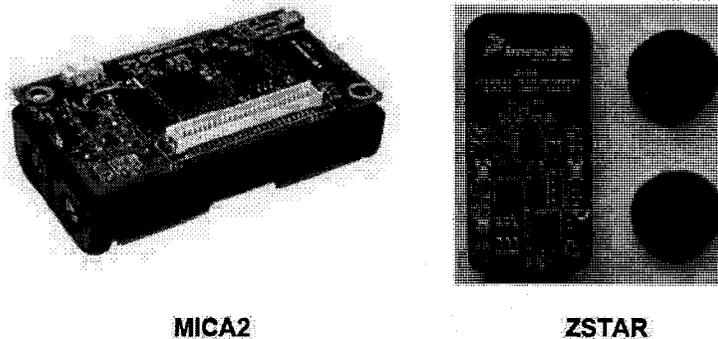


Fig. 2.3: Two commercial sensor products

To give an example, we introduce two typical commercial products from different sensor system companies in Fig. 2.3. The MICA2 [21] is from Crossbow Technology, Inc. It is derived from the family of Berkeley motes, WeC and Rene. Its size is $58 \times 32 \times 7$ (mm) and its weight is 18g. It is equipped with temperature and humidity sensors. With appropriate use of the sleep mode, the node can run more than one year using $2 \times$ AA batteries. ZSTAR [22] is a ZigBee technique product, developed by Freescale Semiconductor, Inc.. ZSTAR's size is $6 \times 6 \times 1.45$ (mm). Its weight can be from 1.5g to 6g, depending on the type of sensing unit. The size and weight of the two products differ by a factor of about 10.

All sensor products provide switchable working modes as a solution of the energy

Table 2.1: Power management of two sensor platforms

	Mode	MICA2	ZSTAR
Processor	Sleep mode	$<15\mu\text{A}$	750nA
	Wait mode	—	1mA
	Active mode	8 mA	3.5mA
Sensing Unit	Sleep mode	—	$3\mu\text{A}$
	Active mode	—	$500\mu\text{A}$
Communication Unit	Sleep mode	$<1\mu\text{A}$	$2.3\mu\text{A}$
	Idle mode	—	$500\mu\text{A}$
	Transmit mode	27mA	30mA
	Receive mode	10mA	37mA

constraint. Table 2.1 listed the energy consumption of the two example units in different working modes. In MICA2, under the sleep mode, the current drain of the processor is less than $15\mu\text{A}$, but, under the active mode, it may reach up to 8mA. Similarly, in transmit mode, the communication unit consumes more than 10,000 times the energy it does in sleep mode. Considering these facts, protocols must try to keep nodes in their low energy consumption modes. Table 2.1 also shows that the communication unit is the most energy consuming part. Thus, being highly energy efficient is the most vital target of communication protocol design.

2.1.3 MAC Layer Protocols

MAC layer protocol plays an important role as the intermediary between the physical and network layers. MAC layer protocol is responsible for reliable point-to-point and point-to-multipoint connections since it controls antenna activities.

In a conventional network, throughput, fairness, latency and bandwidth utilization are considered the primary attributes. In WSN designs, those attributes are put secondary because the application targets of WSNs do not require a high rate and

high performance network. The energy efficiency, device management and efficient resource sharing become the keys to designing MAC layer protocols for WSNs [23]. Various methods have been tried to achieve the goals.

- S-MAC (Sensor MAC) [24] is the best known MAC protocol for WSNs. It utilizes a combined scheduling and contention scheme by introducing two novel techniques: improved periodic sleep and message passing. To reduce the influence of sleep delay, S-MAC presents the adaptive listening technique. With adaptive listening, each node overhears its neighbor's transmissions and wakes up for a short time at the end of the transmission. The experimental results show that, on a source node, an 802.11-like MAC consumes 2-6 times more energy than S-MAC for traffic loads where messages are sent every 1-10s [24].
- LMAC (Lightweight MAC) [25] is a TDMA(Time-Division Multiple Access) approach. When a node wants to send a packet, it waits until its time slot comes around, then broadcasts a request message header in the control section with the details of the destination and length. After broadcasting the request message, the sender node immediately proceeds with transmitting the data. Other nodes hear the request message; they turn off their radio during the data part if they are not an intended receiver of the broadcast or unicast message. Compared with S-MAC, LMAC is able to extend the network lifetime 3.8 times [25].
- G-MAC (Gateway-MAC) [26] is a contention-based protocol similar to the IEEE 802.11 Ad Hoc mode. Its innovation is using the main idea of LEACH [10,11] to elect a gateway in each cluster to manage MAC. With this novel scheme, G-MAC can make each node wake up in time for its communication while sleeping most of the time. The gateway assigns time slots for each node's with intra-cluster

data to be sent; it collects and then forwards inter-cluster traffic. The gateway node keeps changing to maintain almost equal energy consumption levels and memory consumption levels of all nodes in the cluster. G-MAC further reduces the time spent in idle listening. Simulations show G-MAC can obtain 250 times longer network lifetime over S-MAC for unicast traffic [26].

- In 2003, the first standard of WSNs was released to avoid proliferation of incompatible proprietary protocols. IEEE 802.15.4 (ZigBee™ MAC) [27], the Low Rate Wireless Personal Area Network (LR-WPAN) standard, is the first open standard designed for WSNs. ZigBee™ MAC is a contention-based protocol and provides an optional frame structure, beaconing mode and non-beaconing modes. It includes many features designed to enable low power consumption and low-cost implementation.

Basically, all the MAC layer protocols can be classified as either fixed-assignment or contention-access protocols. Many research results show that fixed-assignment protocols have better performance with their natural advantage of having no collision or control overhead while the contention-access protocols are born with these defects. In the literature, many network routing protocols adopt the fixed-assignment scheme for their MAC layer, especially with the TDMA approach.

2.1.4 Network Routing Protocols in WSNs

Crucial design factors of network routing protocols in WSNs

As a network lacking infrastructure, it is impossible to build a global scheme for a WSN. Therefore, the classical network routing protocols cannot be applied to sensor networks. In WSNs, the data traffic may have significant redundancy since multiple

sensor nodes may generate the same data within the interested area. The redundancy needs to be exploited by the routing protocols in order to improve energy and bandwidth utilization. Moreover, the size and cost limitations of sensor nodes results in corresponding constraints on resources such as energy, memory and computational speed. The network layer protocols need to address following unique factors:

- **Self-organization.** Sensor nodes must self-organize because they are deployed in an unsupervised network. Dynamic deployment requires each sensor node to manage its transmission range and transmission schedule to maintain connectivity while avoiding collisions. In WSNs, the computation and communication protocols must be distributed to achieve autonomous operation.
- **Energy efficiency.** Compared with the rapid advances in fabrication techniques which reduce the limitations on memory and processor capability, the slow progress of power supply technology makes energy constraints unsolvable. Sensor nodes are designed to be battery driven, hence operate on an extremely low energy budget. Moreover, sensor nodes may not be rechargeable since the WSNs allow deployment of sensor nodes in inaccessible terrain or disaster relief operations. Minimizing energy consumption is pushed to the front line of research. As the communication is the single energy intensive activity, choosing an energy efficient communication protocol is a key to saving energy.
- **Flexibility.** To be adapted into various WSN scenarios, the related protocols must be flexible enough. The diverse conditions of applications, deployment environments and sensor device capabilities cause the constraints to vary widely. Although some conditions can be predetermined or evaluated before network deployment, the topology keeps changing during network operation with new

nodes joining, existing nodes failing, and the environment changing. How is it feasible to adapt to these differences by slightly adjusting the protocol? All of the possibilities must be addressed by adding the necessary options to enhance the self-regulation capability.

- **Scalability.** In a WSN, the number of sensor nodes deployed in the study field may be on the order of hundreds or even thousands and the nodes may be very close to each other. In a large scale and high density network, the bandwidth is limited and must be carefully shared among all the nodes. Facing this challenge, many researchers focus on how to effectively control the transmission range and organize the transmission schedule.
- **Fault tolerance.** In contrast to traditional networks, sensor nodes are unattended and randomly deployed. They may fail or be blocked due to environment change, hardware damage or lack of power. Facing failure of one or several nodes, the protocols must ensure a single failure will not lead to catastrophic failures or affect other parts of the network. In other words, WSNs should be robust to individual node failure and ensure the desired service quality.
- **Accuracy and Latency.** Providing accurate and real-time information is the objective of WSN applications. Out-of-date data is almost always useless. The protocol must guarantee that the data are delivered through the network within application specific time and precision constraints. There is a tradeoff between accuracy, latency and energy efficiency. An ideal protocol design will offer timely information of desired accuracy with minimum energy consumption.

Taxonomy of network routing protocols

Network routing protocols are responsible for the network structure and routing scheme. Many researchers have proposed routing solutions for WSNs. The proposed routing protocols can be broken down into different groups based on assorted criteria [8, 28]. Table 2.2 shows a simple taxonomy of network routing protocols in WSNs using the typical approaches. To focus on our specific research interests, we only present further discussion of the flat and hierarchical routing schemes for network structure.

Table 2.2: A simple taxonomy of network routing protocols

Criteria	Categories	Examples
Network structure	Flat	SPIN [29]
	Hierarchical	LEACH [10, 11]
Resource awareness	Energy-aware	HEED [30]
	Fidelity-aware	Directed Diffusion [31]
Protocol operation	Data-centric	SPIN [29]
	Geographic	GFG [32]
	QoS-awareness	SAR [33]
	Query-based	COUGAR [34]

In a flat network, all nodes are typically assigned an equal role and functionality. The desired data are sent out to the network by the multihop route as shown in Fig. 2.4(a). To eliminate many redundant transmissions through the network, flat protocols focus on how to route based on the application queries. Most flat protocols are data-centric because they are designed to ensure nodes only transmit the data which match the query attributes. In many cases, flat protocols result in more complicated routing because of the large scale and dynamic network topology of WSNs. SPIN (Sensor Protocols for Information via Negotiation) [29] and DD (Directed Dif-

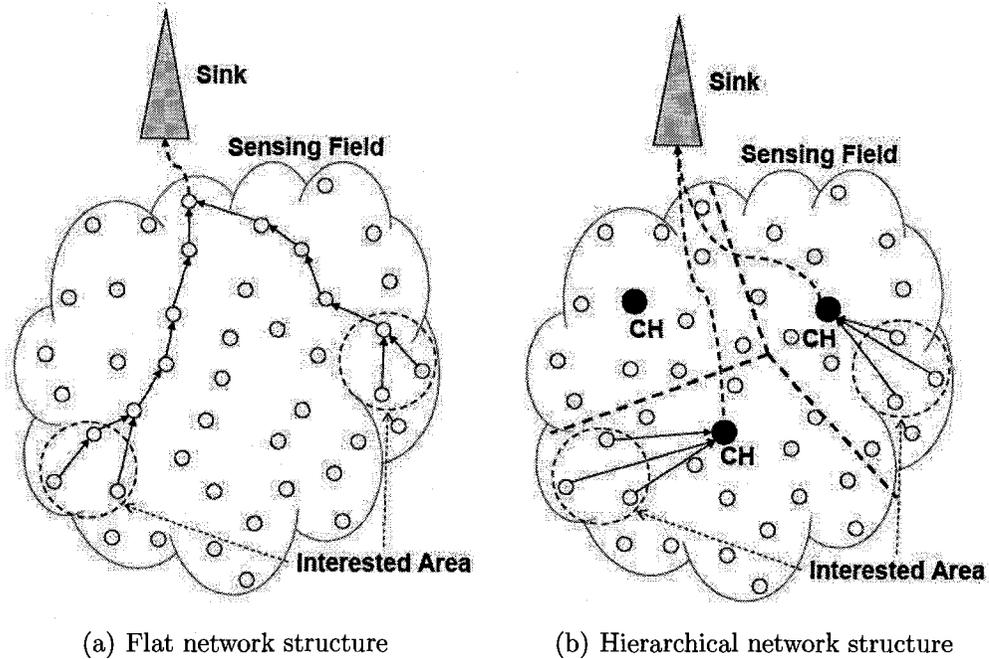


Fig. 2.4: Two typical WSN structures

fusion) [31] protocols are important flat protocols which motivated the design of many other protocols that follow similar concepts.

In hierarchical networks, nodes play different roles, such as CHs and cluster members. The higher level nodes manage the grouped lower level nodes and collect data from them. After aggregating data, these higher level nodes send it on to upper level nodes. For instance, Fig. 2.4(b) shows a two layer network. Only a small fraction of the nodes, CHs, are allowed to communicate with the sink. Each CH collects data from the nodes in its cluster, aggregates the data and transmits the aggregated data to the sink. All of the hierarchical routing protocols aim at selecting the best CH and clustering the nodes into appropriate clusters in order to save energy. Since the CHs have responsibility for the collecting, aggregating, and transmitting data over longer distances to the sink, they consume more energy compared to the other cluster members. The hierarchical clustering protocol may execute reclustering and reselecting of

CHs periodically in order to distribute the load uniformly among the whole network. More details of hierarchical routing protocols will be presented in section 2.2.

Comparing the flat and hierarchical routing protocols, hierarchical protocols exhibit better performance and conspicuous advantages in many different WSN applications. The following characteristics of hierarchical protocols are summarized from related researches [8,9,28].

- **Hierarchical network routing is an efficient way to reduce energy consumption.** Within a cluster, the CH controls a minimal number of active nodes to cover the target area and avoid redundant data. Outside of the clusters, CHs perform data aggregation and fusion locally in order to decrease the number of transmitted messages to the sink.
- **Hierarchical routing protocols assign different tasks depending on the ability of sensor nodes.** By performing energy optimization, hierarchical routing protocols offer balancing of the network load. For instance, the higher-energy nodes can be used to aggregate and send data to the sink, while lower-energy nodes can be used to perform the sensing in the proximity of the target.
- **Hierarchical routing protocols can easily perform reservation-based scheduling to achieve collision free operation.** Under the CH's management, cluster members follow their own schedule to send, receive and listen. Hence, nodes avoid much energy waste by turning to sleep mode and waking up only for their assigned time slot. Clustering reduces channel collisions and results in better network throughput under high load. Hierarchical routing protocols are particularly useful for WSN applications that require scalability to hundreds or thousands of nodes.

- **Hierarchical routing protocols offer simple routing schemes.** Although the flat protocol finds the optimal routing, there is added complexity; the low capability sensor nodes have problems executing the complex algorithms and schemes. Hierarchical routing protocols separate communication into inter-cluster and intra-cluster. CHs connect to the sink and other CHs while other nodes connect to their own CHs. The simple routing also highly reduces the amount of message exchange for the routing setup.
- **Hierarchical routing protocols make query dissemination much easier than flat protocols.** When the sink needs to disseminate a query, the sink only sends the query to CHs. CHs transfer the query to related nodes based on their properties. With this strategy, the number of transmitted messages is decreased.

Although hierarchical routing protocols have their native weaknesses such as requiring time synchronization, potential producing non-optimal routing, and utilizing higher overhead for cluster management, they show attractive advantages by dealing with the constraints in WSNs. Compared with flat protocols, they offer a more feasible solution to handle large scale networks with their enhancements to better share limited wireless channel bandwidth, balancing node energy consumption and reduce communication expense.

2.2 Related Works on Hierarchical Protocols

In this thesis, our research narrows down to the hierarchical protocols in WSNs which can well adapt to large scale networks and easily address certain system parameters. Considering the method of CH selection, the hierarchical routing protocols can be

classified into random-selected-CH protocols and well-selected-CH protocols. The former randomly select CHs and then rotate the CH task among all nodes, while the latter carefully select appropriate CHs and then gather nodes under the CHs based on the network status.

2.2.1 Random-selected-CH Protocols

LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [10, 11] protocol is proposed to balance the energy dissipation in sensor networks. There are many protocols derived from LEACH by using the same scheme to randomly pick CHs. The main idea of LEACH is that sensor nodes can be randomly selected as CH based on their previous experiences of being a CH. In the cluster formation phase, each sensor node generates a random number between 0 and 1. Each sensor node has its threshold $Th(\text{LEACH})$ which is related to the predefined percentage of CHs in a network. If the generated random number is less than $Th(\text{LEACH})$, then the node becomes the CH; otherwise, it joins a cluster to be a cluster member. How to calculate the threshold $Th(\text{LEACH})$ is the key of LEACH.

$$Th(\text{LEACH}) = \begin{cases} \frac{P}{1 - P * (r \bmod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

P is a predefined percentage of CHs among all nodes. The optimal number of P is estimated to be 5% of the total number of nodes. The current round is specified by r . G is the set of nodes that have not been selected as a CH in the last $1/P$ rounds. This equation means the node will get less, or even no chance to be CH again if it has

been a CH in past rounds. As the result, CHs change randomly in order to balance the energy dissipation of nodes. Once the CHs are decided, other nodes choose one of them to join by comparing the Received Signal Strength(RSS) of the advertisements from CHs.

After clusters are set up, the CH broadcasts a transmission schedule within the cluster and asks its member nodes to send data based on a TDMA approach. In the steady phase, CHs are responsible to aggregate and send data to the sink. All data processing is local to the cluster. After a certain period of time spent in the steady phase, the network goes to formation phase to redo the clustering. One thing must be noted: the formation phase is much shorter than the steady phase, therefore, LEACH has light overhead.

LEACH is completely distributed and requires no global knowledge of the network. LEACH achieves a factor of 7 or more reduction in energy dissipation compared to direct communication and a factor of 4 to 8 compared to the minimum transmission energy routing protocol [10]. LEACH clustering terminates within a constant number of iterations but it does not guarantee good cluster head distribution and assumes uniform energy consumption for cluster heads. Furthermore, the idea of dynamic clustering brings extra overhead, e.g. head changes and advertisements, which may diminish the gain in energy consumption.

HEED

Hybrid Energy-Efficient Distributed clustering (HEED) [30] protocol is an energy-aware hierarchical approach improved from LEACH. To avoid the problem of selecting a low energy node as a CH, HEED is proposed to set up well-distributed clusters.

In LEACH, with the strategy that every node has an equal chance to be CH,

the network may choose a “bad CH” which leads to higher energy consumption and higher probability of a crashed cluster. HEED initializes a probability for each node to be a tentative CH depending on its residual energy and makes the decision according to the energy cost based on the connectivity degree of the node. Also, HEED adopts multihop communication to further reduce energy consumption. HEED uses residual energy as the primary clustering parameter to select a number of tentative CHs. Those tentative CHs inform their neighbors of their intentions to become CHs. These advertisement messages include a secondary cost measure that is a function of neighbor proximity or node degree. This secondary cost is used to avoid elected CHs being within range of each other, and to guide the regular nodes in choosing the best cluster to join. If a CH is far from the sink, it tries to send the aggregate data to another CH instead of sending to the sink directly.

Simulation results show that HEED prolongs network lifetime, and the clusters it produces exhibit several appealing characteristics. By adjusting the parameters, HEED can be tuned to optimize resource usage according to network density and application requirements. As the number of nodes increases, HEED has been shown to outperform LEACH by about 100% to 300% regarding when the first node dies. Compared to LEACH, HEED dissipates only 30% as much energy during clustering [30].

ERA

Energy Residue Aware (ERA) [35] clustering algorithm is another energy-aware hierarchical approach. It is improved from LEACH by including the communication cost into the clustering. The communication cost includes residual energy ($E_{\text{CH-rem}}$ and $E_{\text{nonCH-rem}}$), communication energy from the CH to the sink (E_{toBS}) and commu-

nication energy from the cluster members to the CH (E_{toCH}). There is a difference from HEED: ERA uses the same CH selection scheme as LEACH but provides an improved scheme to help non-CH nodes choose a “better” CH to join. After CHs are selected, ERA adds the three factors to help non-CH nodes choose an appropriate CH. Equation 2.2 helps nodes to calculate the clustering cost and find CH according to maximum energy residue. (S_C is the set of CHs, S_N is the set of non-CH nodes).

$$\begin{cases} (E_{\text{CH-res}})_i = (E_{\text{CH-rem}})_i - (E_{\text{toBS}})_i, & i \in S_C \\ (E_{\text{nonCH-res}})_j = (E_{\text{nonCH-rem}})_j - (E_{\text{toCH}})_{ji}, & j \in S_N, \forall i \in S_C \\ \max\{(E_{\text{CH-res}})_i + (E_{\text{nonCH-res}})_j | \forall i \in S_C\}, & j \in S_N \end{cases} \quad (2.2)$$

As a result, the non-CH nodes can join the cluster where the CH is closest to it and the sink. With the improved traffic pattern of data aggregated to the sink, this protocol manages the communication cost both inter-cluster and intra-cluster. ERA has 300 rounds more network lifetime than LEACH does [35].

PEGASIS and Hierarchical PEGASIS

Power-Efficient GAttering in Sensor Information Systems (PEGASIS) [36,37] is based on LEACH and uses the greedy algorithm to organize all sensor nodes into a chain and then periodically promote the first node on the chain to be the CH. Rather than forming multiple clusters, PEGASIS forms chains from sensor nodes so that each node communicates only with its close neighbors, and allows only one designated node to send to the sink in each round. To locate the closest neighbor node, each node uses the RSS to estimate the distance to all neighboring nodes and then adjusts the signal strength so that it only communicates with its closest neighbor.

With the focus on how to reduce transmission energy consumption, nodes take turns transmitting to the sink. Building a chain to minimize the total length is similar to the traveling salesman problem, which is known to be intractable. However, with the radio communication energy parameters, a simple chain built with a greedy approach performs quite well.

PEGASIS provides performance gain through decreasing the amount of transmission and reception by aggregating data and eliminating the overhead caused by cluster formation in LEACH. The PEGASIS protocol outperforms LEACH by about 100% to 200%, regarding when 1%, 25%, 50%, and 100% of the nodes die, with different network sizes and topologies [36]. However, PEGASIS introduces an excessive delay for distant nodes in the chain, and since only one node can connect to the sink, the delay is even worse. Dynamic topology adjustment is required in PEGASIS since a sensor node needs to know the energy status of its neighbors in order to calculate where to route its data. Such topology adjustment can introduce significant overhead, especially for heavy traffic networks.

To decrease the delay incurred for packets, hierarchical PEGASIS was introduced in [37]. It proposes a solution to the data gathering problem by considering *energy* \times *metric* delay. Hierarchical PEGASIS has two improvements, using CDMA for signal coding, and spatially separating nodes. The chain-based protocol with CDMA-capable nodes constructs a chain of nodes that forms a tree-like hierarchy, and each selected node at a particular level transmits data to a node in the upper level of the hierarchy. This method ensures data transmitting in parallel and reduces the delay significantly. Since the tree is balanced, the delay will be in $O(\lg N)$ where N is the number of nodes.

RRCH

Round-Robin Cluster Header (RRCH) [38] performs cluster formation only once to avoid the high energy consumption during clustering. Within the fixed cluster, RRCH uses the round-robin method to choose the node to be the CH. RRCH uses a similar method to LEACH to set up clusters. Once the clusters are set up, the nodes follow the schedule to change their role in turn. Every node has a chance to be CH during a frame. When a node has been detected as an abnormal node, the CH modifies the scheduling information and broadcasts it to the entire cluster during frame modification; then its cluster members delete the abnormal node based on the received schedule information.

With the single clustering process, RRCH can avoid the energy dissipation of reclustering. With the round-robin method, the CH role is rotated through the whole cluster, thus the energy dissipation can be uniform. RRCH shows better performance with energy efficiency. But RRCH has the same defect of LEACH: no guarantee of cluster quality. Once clusters are set up, RRCH keeps the fixed clusters and only adjusts the schedule when there is an abnormal node. LEACH with the periodic reclustering can alleviate the deterioration of cluster quality. RRCH cannot handle clusters with bad quality, such as overlays and too small or too big a cluster size.

2.2.2 Well-selected-CH Protocols

To avoid the problem caused by random CH selection, there are many other approaches focusing on how to select appropriate CHs to achieve efficient communications.

LEACH-C

LEACH-C (LEACH-centralized) [11] is an improved version of LEACH. LEACH-C is identical to the LEACH protocol as far as formatting clusters at the beginning of each round. However, instead of nodes randomly self-selecting as a CH, a centralized algorithm is performed by the sink in LEACH-C. The sink collects location information from the nodes, then broadcasts its decision back to the nodes, specific which are to act as CHs and appointing a CH for the rest of the nodes. LEACH-C improves the performance by 20% to 40% depending on the network parameters, compared to LEACH, in terms of the total number of rounds of data collection that can be achieved before sensor nodes start to die. The overall performance of LEACH-C is better than LEACH since it moves the duty of cluster formation to the sink. However, LEACH-C is sensitive to the sink location. Once the energy cost of communicating with the sink becomes higher than the energy cost for cluster formation, LEACH-C no longer provides good performance. Section 2.1.1 already noted that the sinks are located far from the network in most WSN applications. So, the dependence on the sink location is a major disadvantage of LEACH-C.

Dominating Set

Stojmenovic, *et al.* [39] focus on one-to-many traffic patterns and propose a broadcasting algorithm based on neighbor elimination. The idea is to divide all the nodes into four levels: gateway, inter-gateway, intermediate and member. Only internal nodes are used to forward broadcast packets. Selected gateway nodes form a dominating set to ensure high efficiency information transmission. The following rules are used to divide nodes into different levels.

Rule 1: A is an intermediate node if and only if there exist two neighbors of A that

are not direct neighbors themselves. Otherwise the node is a member node.

Rule 2: Given two connected intermediate nodes A and B, if any neighbor of A is also a neighbor of B, then any path via A can be replaced by a path via B, thus node B is an inter-gateway node. We may also say that node A is “covered” by node B.

Rule 3: C and D are two inter-gateway neighbors of an inter-gateway node B. After applying Rule 2, if each neighbor of B is a neighbor of C or D, where C and D are two connected higher degree neighbors of B, then B can be eliminated from the list of gateway nodes.

In the case of ties, the protocol uses a record of degree and node location, where degree is the number of neighbors of a node. Nodes compare their degrees, and the node with higher degree has a greater chance of remaining an internal node; x-coordinates are used as secondary and y-coordinates as tertiary keys. The experiments presented in [39] indicate that about half of the nodes are gateway nodes. More precisely, the ratio varies from 60% for average degree 4 to 45% for average degree 10. By applying the localized dominating sets, this approach can significantly reduce or eliminate the communication overhead. Less communication overhead means higher energy efficiency. However, the dominating set is breakable because any change to the network may cause the entire network to update and recalculate the dominant set again.

AHP

Analytical Hierarchy Process (AHP) [40] is proposed by Yin, et al. as a novel cluster head selection algorithm which is performed by the sink. AHP supposes mobile sensor

nodes. Three factors are considered: energy, mobility, and the distance to the involved cluster centroid. AHP calculates local weight and global weight by using those three factors. For local weight, the evaluation matrices are scaled between 1 and 9, where 9 indicates extremely preferred and 1 indicates equally preferred. As another step, AHP also calculates the global weight of alternative sensor nodes for each factor. AHP chooses the CHs by combining the results of these two weights. The larger the final weight of a sensor node, the more importance it has in enhancing the network lifetime. Thus, the sensor node with the largest weight is selected as the CH.

To maintain the clusters, CH re-selection only occurs when selected CHs die or move to other clusters. The trigger conditions of CH re-selection are based on the mobility and remaining energy of the nodes. Compared to LEACH, AHP improves the network lifetime 160% (based on the time of the last node dead) [40]. Comparing centralized protocols, AHP is more complex than LEACH-C since AHP considers more factors. Because of the issue of increasing communication cost between nodes and the sink with cluster administration information, AHP needs to transmit more information from the network to the sink; this may cause much higher energy consumption.

The above hierarchical routing protocols are different in many aspects, but at this point, there is a big gap to fill. The advantages and disadvantages of two kinds of hierarchical routing protocols are summarized in the following paragraphs. There is no perfect solution to suit every application scenario. Using a flexible way to balance the tradeoffs is always the best design.

- **Random-selected-CH protocols.** Although random-selected-CH protocols can bring more flexibility and toleration, these approaches have three main dis-

advantages. Firstly, the randomly picked CH may have a higher communication cost because it has no knowledge of intra-cluster or inter-cluster communication. Secondly, if periodic CH rotation is used to reduce the effect of CH random selection, the re-selection itself uses extra energy to rebuild clusters. Periodic CH rotation also leads to an uneven wave of performance due to the nonstop change. Thirdly, the random selection cannot guarantee good protocol performance. In other words, the best arrangement and the worst arrangement have an equal chance to be used in the network.

- **Well-selected-CH Protocols.** The well-selected-CH Protocols can provide better cluster quality, but they usually have a more complex scheme and higher overhead to optimize the CH selection. Some approaches use the sink to help choose CHs by frequently collecting information from nodes. However, the sink performing the algorithm introduces another issue that increases communication cost between the nodes and the sink because they need to frequently exchange administrative information. Other researchers have tried to use the optimization algorithms to distinguish the roles of nodes. But there may not be enough fault tolerance in these schemes because any change to the network may cause the entire network to update information and perform reclustering.

Chapter 3

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

In this chapter, we advocate the HAC approach by illustrating the steps of HAC algorithm execution in WSNs.

3.1 Basic Concepts of the HAC Algorithm

Our world is fully filled by information. In order to improve our understanding, people try to classify objects or phenomena by considering their similarities or dissimilarities according to certain rules. Through the long history of human development, classification has always played an important role. There are many different methods to help us sort, analyze and manage a large amount of information.

HAC [12–14] is a conceptually and mathematically simple clustering approach to data analysis. It organizes data into a hierarchical structure according to a resem-

blance matrix. The the HAC algorithm groups data objects from singleton clusters into a cluster including all individuals. Through a series of merging operations, the HAC algorithm can provide very informative descriptions and visualization for the potential data clustering structures, especially when real hierarchical relationships exist in the data [41]. The HAC algorithm has been successfully applied to many disciplines due to its use of simple numerical methods to make clustering decisions.

Execution of the HAC usually comprises three common key steps: obtain the data set, compute the resemblance coefficients, and execute the clustering algorithm. For each step, there are various choices. For example, input data types can be either quantitative or qualitative. Resemblance coefficients also have two types, dissimilarity coefficients and similarity coefficients. We will describe the related concept of the HAC algorithm adopted in our research for WSNs, and illustrate the use of both quantitative and qualitative input data, in following sections.

3.1.1 Obtain the Input Data Set

An input data set for the HAC algorithm is formed as a component-attribute data matrix. Components are the entities that can be grouped based on their similarities. Attributes are the properties of the components. For example, the attributes can be the location of nodes, the residual energy of nodes, or other features. Fig 3.1 shows a simple randomly generated 8-node network in a 10×10 m² field. The components are the nodes and the attributes are their locations as illustrated in Table 3.1 where the location information is used as the quantitative input data set.

In the same network, the simplest form of qualitative data also can be used for the HAC algorithm. This qualitative data is a binary representation, the value of which must be either 0 or 1. For example, in Table 3.2 one-hop connectivity is used as the

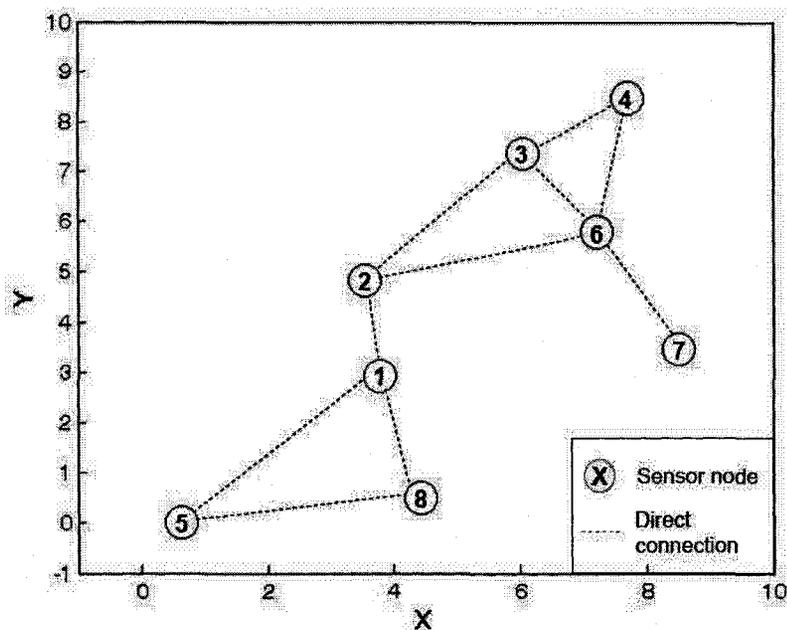


Fig. 3.1: A simple 8-node network

Table 3.1: Node location data matrix for the 8-node network

Component (Node)	Attribute	
	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

qualitative input data set for the 8-node network. The '1' value represents a one-hop connection and the '0' value represents no direct connection. Note that a '1' value is also used for a node to itself and for the entries along the main diagonal to give more weight for the resemblance coefficient computation.

Table 3.2: One-hop network connectivity data matrix for the 8-node network

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	0	1	1	0
{8}	1	0	0	0	1	0	0	1

The resemblance matrix can easily add or remove components, or modify attributes from the data set for different applications. Obviously, the more factors that are considered, the more restricted assumptions and additional computations are needed.

3.1.2 Computation of Resemblance Coefficients

A resemblance coefficient for a given pair of components indicates the degree of similarity or dissimilarity between these two components, depending on how the data is represented. The smaller the dissimilarity coefficient, the more similar the two components are. A similarity coefficient is the opposite. A resemblance coefficient can be calculated from the quantitative or qualitative input data. Table 3.3 shows the dissimilarity quantitative coefficients which measure the literal distance between two components when they are viewed as points in a two-dimensional array formed by the input attributes. The coefficients are calculated using the Euclidean distance based on the input data shown in Table 3.1. The values of distance between any pair of nodes in Table 3.3 are given by the Pythagorean theorem. Equation 3.1 shows the Pythagorean theorem which uses the coordinates to calculate the distance between node a and b.

$$D_{ab} = [(a_x - b_x)^2 + (a_y - b_y)^2]^{1/2}, \quad (3.1)$$

Table 3.3: Initial resemblance matrix with quantitative data

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

To deal with the qualitative input data, there are various ways to calculate the resemblance coefficients. Three typical methods are [3, 6, 14]:

- JACCARD Coefficient:

$$C_{ab} = N_{1-1}/(N_{1-1} + N_{1-0} + N_{0-1}), \quad (3.2)$$

- SORENSON Coefficient:

$$C_{ab} = 2N_{1-1}/(2N_{1-1} + N_{1-0} + N_{0-1}), \quad (3.3)$$

- Simple Matching Coefficient:

$$C_{ab} = (N_{1-1} + N_{1-0})/(N_{1-1} + N_{1-0} + N_{0-1} + N_{0-0}), \quad (3.4)$$

N_{1-1} , N_{1-0} , N_{0-1} , N_{0-0} are counts of 1-1, 1-0, 0-1, and 0-0 matches of attribute-pair between any two components x and y .

Table 3.4: Resemblance matrix with qualitative data using SORENSON dissimilarity coefficients

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

When quantitative data, such as location information, is not available, qualitative data can be adopted. Table 3.2 is using the connectivity information as input. Each node merely knows its neighbors without distance or any other information. Note that the original qualitative coefficients are similarity coefficients. To be consistent with the resemblance coefficients using quantitative data, Table 3.4 demonstrates use of a dissimilarity coefficient instead of a similarity coefficient to indicate the difference between any two clusters. For nodes 5 and 8, for instance, the parameters and their values are $N_{1-1} = 3$, $N_{1-0} = 0$, $N_{0-1} = 0$, $N_{0-0} = 5$. Using the SORENSON method, the similarity coefficients between 5 and 8 are 1, hence, their dissimilarity coefficients become 0 (1 - original SORENSON coefficient). The coefficients in Table 3.4 are calculated by using the SORENSON method, given by Equation 3.1.2. Further related discussion of qualitative input data will be presented in Section 3.2.2.

3.1.3 Execution of the HAC Algorithm Methods

After building up the resemblance matrix, the HAC algorithm uses the minimal coefficient in the matrix, then executes the clustering algorithm to assign the nodes into a “tree”. Each step merges two clusters together and updates the resemblance matrix. Updating the resemblance matrix is an important step, and various algorithm methods can be adopted. There are four main types of the HAC algorithm methods [12].

- Single LINKage (SLINK) method, also called the nearest neighbor method. It defines the similarity measure between two clusters as the minimum resemblance coefficient among all pair entities in the two clusters.

$$C_{\text{SLINK}} = \text{Min}(C_1, C_2, \dots, C_n), \quad (3.5)$$

- Complete LINKage (CLINK) method, also called the furthest neighbor method. It defines the similarity measure between two clusters as the maximum resemblance coefficient among all pair entities in the two clusters.

$$C_{\text{CLINK}} = \text{Max}(C_1, C_2, \dots, C_n), \quad (3.6)$$

- Un-weighted Pair-Group Method using arithmetic Averages (UPGMA). This defines the similarity measure between two clusters as the arithmetic average of resemblance coefficients among all pair entities in the two clusters. UPGMA is the most commonly adopted clustering method.

$$C_{\text{UPGMA}} = \frac{1}{n} \sum_{i=1}^n C_i, \quad (3.7)$$

- Weighted Pair-Group Method using arithmetic Averages (WPGMA). This is the simple arithmetic average of resemblance coefficients between two clusters without considering the cluster size.

$$C_{\text{WPGMA}} = \sum_{i=1}^n W_i C_i, \quad (3.8)$$

To explain how to execute the HAC algorithm, we use SLINK as the execution method with quantitative data in the following example. After the clustering starts, we check the resemblance matrix to find the minimum coefficient, $D_{12} = 1.94$. In other words, we can easily know that $\{1\}$ and $\{2\}$ are the two closest nodes by checking Table 3.3. The HAC algorithm merges $\{1\}$ and $\{2\}$ together and updates the resemblance matrix by using the SLINK method. Fig. 3.2 shows the new cluster and the updated matrix in the first step. After the first step of merging, there are seven clusters $\{1,2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{6\}$, $\{7\}$, $\{8\}$. In the following steps, the HAC algorithm repeats all the actions: choose the minimum coefficient, merge clusters, and update the matrix. Figure 3.2 to 3.6 show generated clusters and the corresponding updated matrices from the first to fifth merging step.

The results of the HAC algorithm are usually depicted by a binary tree or dendrogram. The root node of the dendrogram represents the whole data set and each leaf node is regarded as a data object. The intermediate nodes, thus, describe the extent that the objects are proximal to each other, and the height of the dendrogram usually

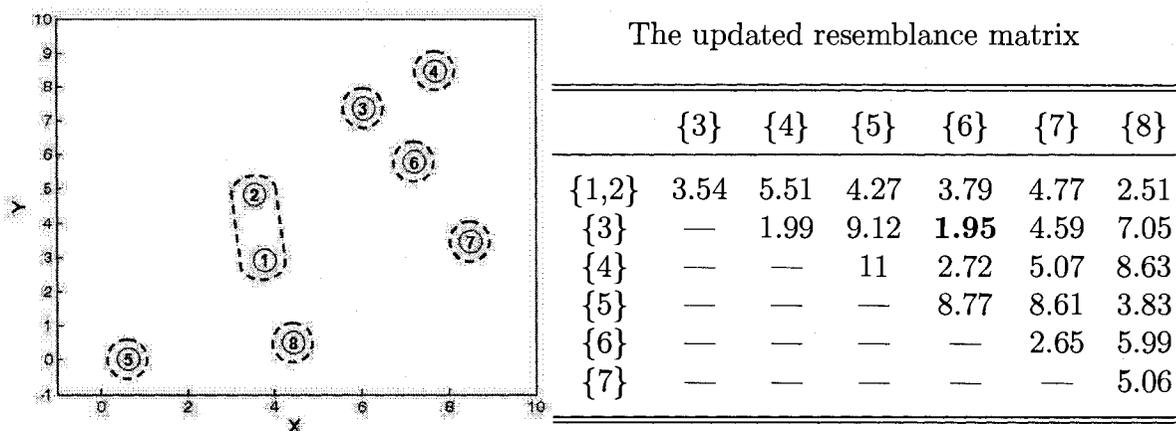


Fig. 3.2: The first step of the HAC: using SLINK with quantitative data

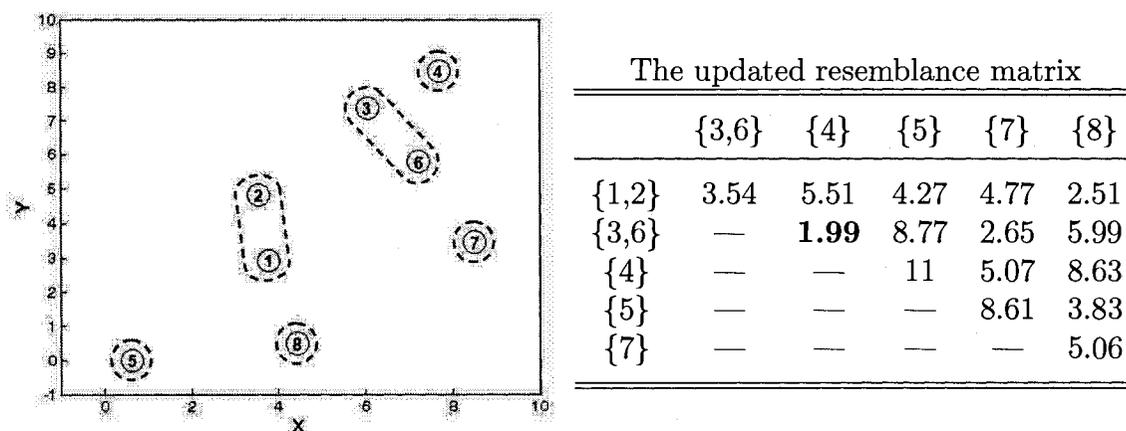
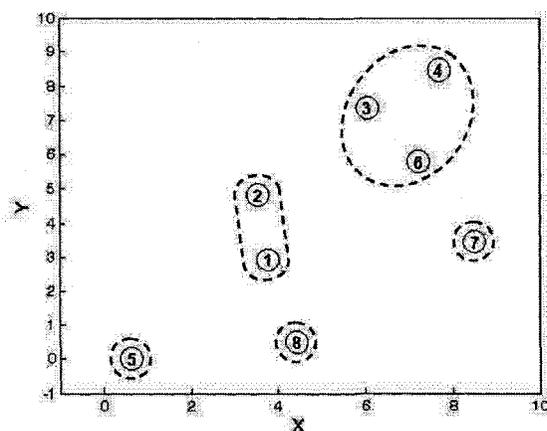


Fig. 3.3: The second step of the HAC: using SLINK with quantitative data

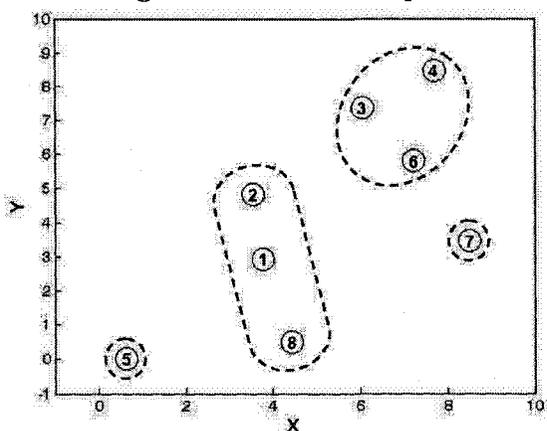
expresses the distance between each pair of objects or clusters, or an object and a cluster. The ultimate clustering results can be obtained by cutting the dendrogram at different levels. In Fig. 3.7, the height is the resemblance coefficient of two merged clusters. Using the SLINK approach (a), for instance, there are three clusters $\{1,2,8\}$, $\{3,6,4,7\}$ and $\{5\}$ after the fifth clustering step, as shown in Fig. 3.6. With the same data set, the HAC algorithm may get different clustering results by using different the HAC algorithm methods, e.g., (b) and (c).



The updated resemblance matrix

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

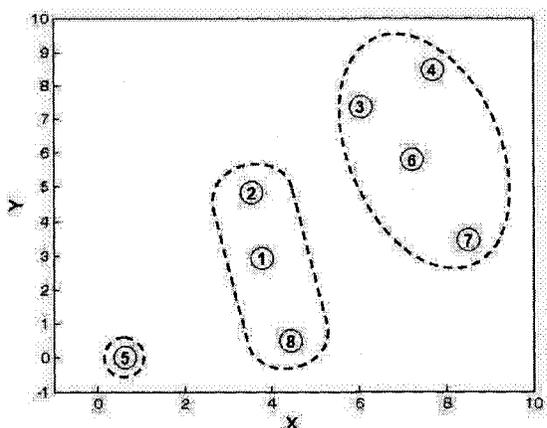
Fig. 3.4: The third step of the HAC: using SLINK with quantitative data



The updated resemblance matrix

	{3,6,4}	{5}	{7}
{1,2,8}	3.54	3.83	4.77
{3,6,4}	—	8.77	2.65
{5}	—	—	8.61

Fig. 3.5: The fourth step of the HAC: using SLINK with quantitative data



The updated resemblance matrix

	{3,6,4,7}	{5}
{1,2,8}	3.54	3.83
{3,6,4,7}	—	8.61

Fig. 3.6: The fifth step of the HAC: using SLINK with quantitative data

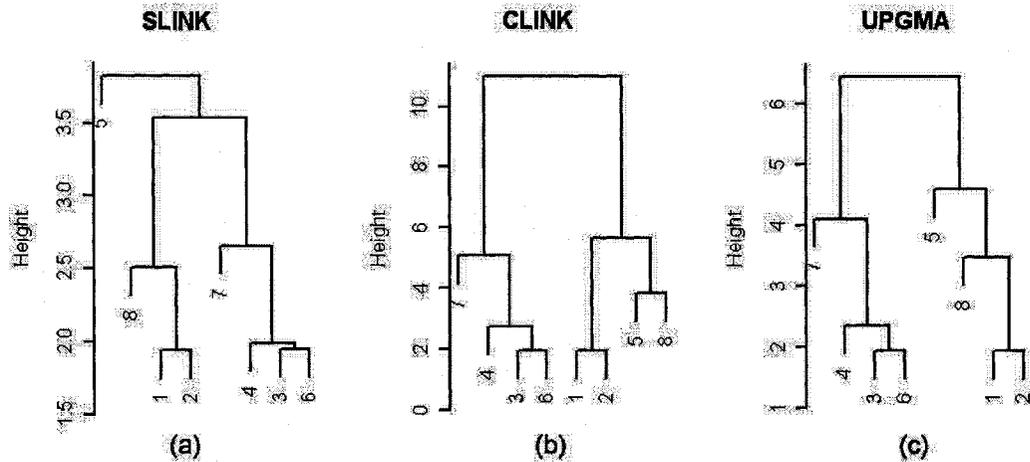


Fig. 3.7: Dendrogram using different the HAC algorithm methods with quantitative data

3.2 Cluster Formation for WSNs

As stated before, the HAC algorithm tends to generate a hierarchical cluster including all individuals. To apply the HAC algorithm in WSNs, we add three more steps during cluster formation to finally generate appropriate clusters: cutting the hierarchical cluster tree, controlling minimum cluster size and choosing CHs. The total six steps are presented as follows.

Step 1: Obtain the input data set. Nodes exchange messages until they obtain all other nodes' information. The information can be location, RSS or connectivity.

Step 2: Compute the resemblance coefficients. Based on the input data set, we can choose a different method to build a resemblance matrix.

Step 3: Execute the HAC algorithm. Run the HAC algorithm and generate a hierarchical cluster tree.

Step 4: Cut the hierarchical cluster tree. Make a cut using a pre-configured threshold value to determine clusters, (e.g. transmission radius, number of

clusters, or cluster density)

Step 5: Control the minimum cluster size. If cluster size is less than a predefined threshold, `Minimum_Cluster_Size`, the cluster merges with its closest neighboring cluster.

Step 6: Choose CHs. Once clustering is finished, CHs can be initially determined by different strategies. Here, CHs are the nodes which satisfy two conditions: (i) the node is one of two nodes which are merged into the current cluster in the first step or at the lowest level, e.g., {3, 6} or {1, 2} in Fig. 3.7; (ii) the node has the lower ID or the shorter distance to the sink. The nodes merged at the lowest level have the highest resemblance coefficient or are closest. Generally, they are also physically closer to the “center” of a cluster.

3.2.1 Application of the HAC with Quantitative Data

To apply the HAC with quantitative data in WSNs, the HAC algorithm needs to collect information about the distance between pairs of nodes. The HAC algorithm can use the location information to calculate the distance between any pair of nodes. The HAC algorithm can also use the RSS instead of the location information. Using the same example of section 3.1.1, the HAC algorithm adopts the node location as the input data and the node’s transmission radius as the threshold to cut the cluster tree. The `Minimum_Cluster_Size` is 2. The following steps and Fig. 3.8 demonstrate how to generate the clusters and determine the CHs.

Step 1: Each node broadcasts a HELLO message and exchanges information with all other nodes. The message can include the node location information (Table 3.1).

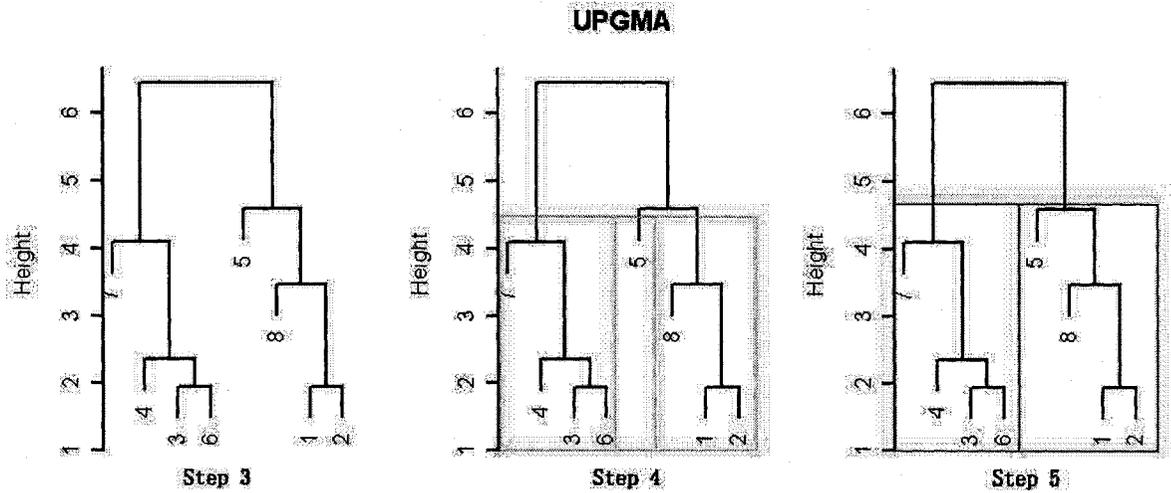


Fig. 3.8: Clustering steps and dendrogram using UPGMA with quantitative data

Step 2: After exchanging HELLO messages, nodes establish a resemblance matrix with quantitative data by using location or RSS information (Table 3.3).

Step 3: Execute the HAC and generate a hierarchical cluster tree by using the UPGMA method with quantitative data (Fig. 3.8, step 3).

Step 4: Cut the cluster tree with the threshold of node transmission radius, for example, using 4.5 as the node transmission radius. The transmission radius can be based on actual transmission capacity, hardware specifications, or the application specific pre-configured value. As a result, three clusters, $\{3, 6, 4, 7\}$, $\{5\}$, and $\{1, 2, 8\}$ are generated, as shown in Fig. 3.8, step 4.

Step 5: The size of cluster 5 is smaller than the `Minimum_Cluster_Size`, 2; it is then combined with its closest cluster $\{1, 2, 8\}$. After this step, there are two clusters finally generated: $\{3, 6, 4, 7\}$, and $\{1, 2, 8, 5\}$ (Fig. 3.8, step 5).

Step 6: At the same time of step 5, the HAC algorithm chooses node 3 and node 1

as the CHs of the final two clusters, since they both were grouped with another node (nodes 6 and 2, respectively) in the first step and both have lower ID.

Those two clusters also correspond to the clustering sequence of the nodes in each cluster, which represents the cluster chain that can be used for the scheduling arrangement. Based on the scheduling arrangement, the cluster members send their data to CH in the appropriate time slot. For instance, cluster $\{3, 6, 4, 7\}$ demonstrates that the sequence of transmission in this cluster is 3, 6, 4, and 7. Without any extra scheduling process, the CHs election and announcement, cluster establishment, and scheduling can all be completed at the same time.

3.2.2 Application of the HAC with Qualitative Data

In the absence of location information, the connectivity information can be used as input. In other words, each connection can be represented with a binary value.

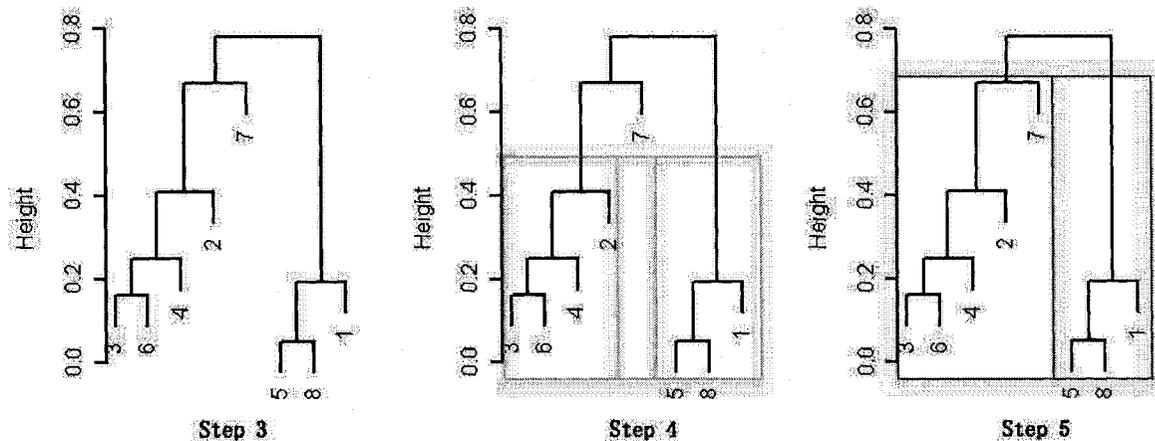


Fig. 3.9: Clustering steps and dendrogram using UPGMA with SORENSON qualitative data

After establishing the resemblance matrix, the HAC algorithm can use four different methods, SLINK, CLINK, UPGMA, and WPGMA, to update the resemblance

coefficients and perform the clustering. Fig. 3.9 shows the result of using the SORENSON method to calculate the resemblance coefficients and UPGMA for clustering computation.

Step 1: Each node broadcasts a HELLO message and exchanges information with all other nodes. After exchanging HELLO messages, each node builds up its neighbor list and broadcasts it. Compared to quantitative data, qualitative data needs more message exchanging to obtain the data set (Table 3.2).

Step 2: After exchanging HELLO messages, nodes establish their resemblance matrices with quantitative data by using location or RSS information. (Table 3.4)

Step 3: Execute the HAC algorithm and generate a hierarchical cluster tree by using the UPGMA method with qualitative data. The height value in Fig. 3.9 is the reversed SORENSON coefficient which indicates the dissimilarity between two clusters (Fig. 3.9, Step 3).

Step 4: After using the threshold 0.5 to cut the cluster tree, the result shows that three clusters can be obtained. They are {3, 6, 4, 2}, {7}, and {5, 8, 1} (Fig. 3.9, Step 4).

Step 5: The size of cluster {7} is smaller than the Minimum_Cluster_Size. Therefore cluster {7} merges with its closest cluster {3, 6, 4, 2}. After adjustment, there are two clusters {3, 6, 4, 2, 7}, {5, 8, 1} (Fig. 3.9, Step 5).

Step 6: Node 3 and node 5 are the CHs of the two final clusters since they both were grouped with other nodes, 6 and 8 respectively, in the first step and have lower IDs.

The result is similar to that obtained using the quantitative data as presented in the previous section, except that node 2 is now in the other cluster. Node 2 is sitting in between two groups and can be clustered with either group. Discrepancies in clustering results are common when different input data or clustering methods are used, since there are various ways to group the data. The most important point is to group nodes that share more commonalities or are physically closer.

Chapter 4

Distributed Hierarchical Agglomerative Clustering (DHAC) Algorithm

As stated in Chapter 3, the results obtained from the HAC algorithm methods demonstrated the feasibility of the HAC approach. However, to execute the HAC algorithm, each node must have global knowledge of all the nodes. This assumption is not realistic for WSNs. In this chapter, we propose the DHAC Algorithm by modifying the HAC algorithm for distributed environments.

The idea of DHAC is that for WSNs, the HAC algorithm does not actually need the global knowledge. To be specific, a node can make use of only one-hop neighbor knowledge for the computation. Nodes that are far apart will not be grouped into the same cluster anyway. Therefore, it is not necessary to include all the information for clustering. As a result, DHAC can remove the assumption that each node has the global information of every other node. We present the DHAC with the concept of distributed clustering for WSNs based on the quantitative location data and

qualitative connectivity knowledge using the different methods.

4.1 General Assumptions

To illustrate the feasibility of DHAC in WSNs, DHAC sticks with the “general assumptions” of WSNs as follows:

- The nodes in the network are quasi-stationary.
- Propagation channel is symmetric.
- The transmission range of nodes are adjustable. All the nodes have the capability to communicate with the sink directly.
- Nodes are left unattended after deployment.
- All nodes have similar capabilities, processing, communication and initial energy.
- The sink is static.
- Each node only has local information, the identification of its one-hop neighbor nodes.

4.2 DHAC Procedure

Without global knowledge, DHAC can make use of the neighboring information to determine if a node actually needs to perform the clustering task. The rationale is that every node knows its one-hop neighbors. Fig. 4.1 illustrates the pseudo code of the DHAC implementation for WSNs.

Definitions:

ID_{Node} : Node ID

ID_{CH} : Cluster Head ID

ResM: Resemblance Matrix

$M_{C_{coeff}}$: The Minimum coefficient in the ResM

M_{CH} : The ID_{CH} of the cluster corresponds to the $M_{C_{coeff}}$ in the ResM

C_{size} : The number of cluster member in a cluster

```

1: procedure SET UP RESM                                ▷ exchange information with neighbors
2:    $ID_{CH} \leftarrow ID_{Node}$                             ▷ self form a singleton cluster
3:   Send HELLO message to one-hop neighbors
4:   .....                                              ▷ keep listening from neighbors
5:   Receive HELLO messages
6:   Establish ResM
7: end procedure

8: procedure EXECUTE DHAC
9:   while  $M_{C_{coeff}} < Threshold$  do
10:    if  $ID_{CH} == ID_{Node}$  then                            ▷ Sending message
11:       $M_{C_{coeff}} \leftarrow$  minimum value in ResM
12:       $M_{CH} \leftarrow ID_{CH}$  with  $M_{C_{coeff}}$ 
13:      if  $ID_{CH} < M_{CH}$  then
14:        Send INVITE message to  $M_{CH}$ 
15:      end if
16:    end if

```

```

.....                                     ▷ Receiving message
17:   if Received INVITE message then
18:     if  $M_{CH} = \text{Sender } ID_{CH}$  then
19:       send CONFIRM message
20:        $ID_{CH} \leftarrow \text{Sender } ID_{CH}$ 
21:     else
22:       Send REJECT message
23:     end if
24:   else if Received CONFIRM message then GoTo MERGE_CLUSTERS
25:   else if Received INFORM message then Update ResM
26:   else if Received REJECT message then Stop send INVITE message
    until ResM updated
27:   end if
28: end while
29: end procedure

30: procedure MERGE_CLUSTERS                                     ▷ cluster merging procedure
31:   Merge two clusters
32:   Combine neighboring information of two clusters
33:   Update ResM by using chosen method
34:   Broadcast INFORM message
35: end procedure

```

Fig. 4.1: Pseudo code of DHAC in WSNs

To set up the local resemblance matrix, in the beginning, each node elects itself

as a cluster and exchanges the information via HELLO messages with its neighbors. In Fig. 4.1, lines 1 - 7 initialize the clustering process by setting the current ID_{CH} as ID_{Node} and exchanging HELLO messages. Each singleton cluster builds its own local resemblance matrix only with the information from its neighbors. The procedure “Set up ResM” corresponds to “obtain input data set and build resemblance matrix”.

After establishing the local resemblance matrix, each cluster can easily find the $Minimum_coefficient(M_{C_{coeff}})$ in its matrix, and then determine the corresponding $Minimum_Cluster_Head(M_{CH})$ associated with $M_{C_{coeff}}$. If M_{CH} satisfies the requirements, the cluster will send an INVITE message to the M_{CH} . Lines 10 - 16 specify two requirements: the sender node must be the current cluster head, and the sender ID_{Node} must be smaller than M_{CH} .

During the clustering process, all clusters keep listening and waiting for messages. When a CH receives a message, it reacts based on the message type.

- In lines 17 - 23, when receiving an INVITE message, the CH checks the source of the message. If the source is its M_{CH} , the CH sends a CONFIRM message back to the source, elects the source to be the new CH, and turns to the sleep mode; otherwise the CH sends back a REJECT message.
- Once a cluster receives a REJECT message from its M_{CH} , the cluster stops sending the INVITE message until its resemblance matrix has been updated.
- After receiving a CONFIRM message, the cluster updates its neighbor list and resemblance matrix, then merges with the cluster headed by M_{CH} to form a new cluster (line 30 - 35). The resemblance matrices of two clusters are updated by using the chosen the HAC algorithm method.
- The CH of the new cluster broadcasts an INFORM message to notify all its

neighbors to update their resemblance matrices (lines 25 - 25 and 34). The INFORM message includes the new cluster information and the combined neighbor list. When a cluster receives an INFORM message, it updates its resemblance matrix with the new cluster information.

Using a predefined threshold, the while loop in lines 9 - 28 controls the size of clusters. The control conditions correspond to the step of cutting the hierarchical cluster tree. After clusters are generated by running DHAC, the `Minimum_Cluster_Size` can also be used to limit the minimum cluster size.

4.3 DHAC: Cluster Formation

The DHAC procedure (Fig. 4.1) demonstrates the DHAC without the assumption of having global information. The clusters can update their resemblance matrices by simply exchanging messages with their one-hop neighbors in each clustering step. In this section, we use the same network shown in Fig. 3.1 to present the DHAC cluster formation.

4.3.1 Cluster Formation with Quantitative Data

To set up the local resemblance matrices, in procedure “Set up ResM,” each node sets itself as a singleton cluster, exchanges messages with its one-hop neighbors, and establishes a local resemblance matrix. Table 4.1 shows the initial distributed resemblance matrices of the singleton clusters after exchanging HELLO messages. The resemblance coefficients are the quantitative euclidean distance which is based on the location information from one-hop neighbors.

In the matrices, the clusters only have the resemblance coefficients for their neighbors. For example, cluster{1} only has the coefficients for clusters {2}, {5}, and {8}

Table 4.1: Initial distributed resemblance matrices with quantitative data

Cluster {1}		Cluster {2}		Cluster {3}	
Neighbor	Coefficient	Neighbor	Coefficient	Neighbor	Coefficient
{2}	1.94	{1}	1.94	{2}	3.54
{5}	4.27	{3}	3.54	{4}	1.99
{8}	2.51	{6}	3.79	{6}	1.95

Cluster {4}		Cluster {5}		Cluster {6}	
Neighbor	Coefficient	Neighbor	Coefficient	Neighbor	Coefficient
{3}	1.99	{1}	4.27	{2}	3.79
{6}	2.92	{8}	3.83	{3}	1.95
				{4}	2.72
				{7}	2.65

Cluster {7}		Cluster {8}	
Neighbor	Coefficient	Neighbor	Coefficient
{6}	2.65	{1}	2.51
		{5}	3.83

since it exchanges information with only its neighbors 2, 5 and 8. Still using the resemblance matrices, each cluster can perform the DHAC to merge with other clusters. As Table. 4.1 shows the matrix of cluster {1} has the smallest coefficient 1.94 with cluster {2}, so cluster {2} is its M_{CH} , and vice versa. Based on the predefined principle, the cluster with the smaller ID launches an INVITE message. So, cluster {1} sends an INVITE message to cluster {2}. Similarly, clusters {3} and {5} also find their corresponding M_{CHS} and they send INVITE messages to cluster {6} and cluster {8}, respectively.

Table 4.2: The first step of DHAC: updated resemblance matrices using SLINK with quantitative data

Cluster {1,2}		Cluster {3,6}		Cluster {4}	
Neighbor	Coefficient	Neighbor	Coefficient	Neighbor	Coefficient
{5}	4.27	{1,2}	3.54	{3,6}	1.99
{8}	2.51	{4}	1.99		
{3,6}	3.54	{7}	2.65		
Member 2	1.94	Member 6	1.95		

Cluster {5}		Cluster {7}		Cluster {8}	
Neighbor	Coefficient	Neighbor	Coefficient	Neighbor	Coefficient
{1,2}	4.27	{3,6}	2.65	{1,2}	2.51
{8}	3.83			{5}	3.83

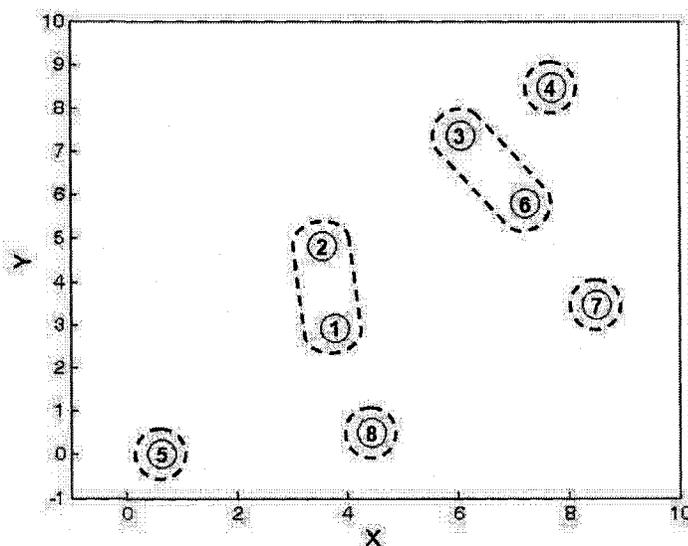


Fig. 4.2: The first step of DHAC: using SLINK with quantitative data

After receiving their INVITE messages, clusters {2} and {6} send back CONFIRM messages, but cluster {8} sends back a REJECT message because cluster 8 finds that

the distance between itself and cluster {5} is not the smallest. After receiving the CONFIRM messages, the related clusters execute procedure “MERGE_CLUSTERS”. Fig. 4.2 and Table 4.2 show the results of the new clusters and updated resemblance matrices after the first clustering step using SLINK with quantitative data.

Table 4.2 demonstrates that after receiving a CONFIRM message, cluster {1} merges with cluster {2} by updating its neighbor list and resemblance matrix. The new cluster {1, 2} broadcasts an INFORM message to its neighbors. The neighbor clusters then update their resemblance matrices based on the INFORM message. Cluster {3} performs the same merging procedure with cluster {6}. At the end of the first step, DHAC gets two new clusters, {1, 2} and {3, 6}. In the second step of the clustering, every cluster can easily find its $M_{C_{\text{coeff}}}$ and M_{CH} in its updated matrix. Fig. 4.3 and Table 4.3 shows the second step results of DHAC algorithm using SLINK. All clusters repeat lines 9 - 28 in the pseudo code, until the $M_{C_{\text{coeff}}}$ in matrices reaches the predefined threshold.

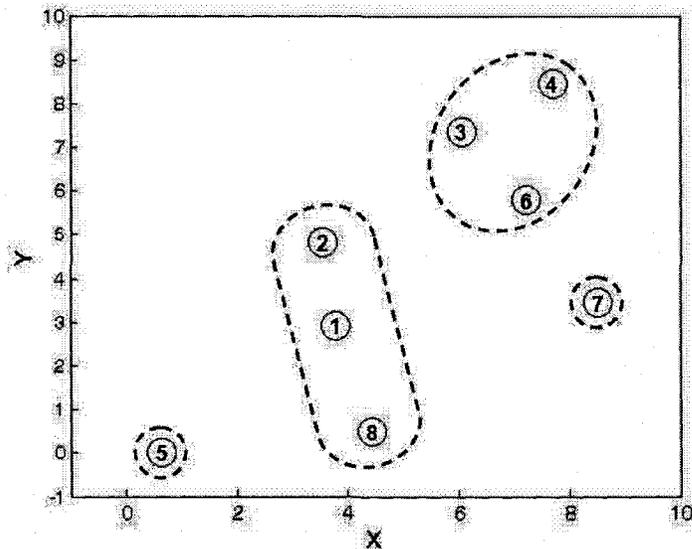


Fig. 4.3: The second step of DHAC: using SLINK with quantitative data

From the above descriptions of the sample example, observe that DHAC has simi-

Table 4.3: The second step of DHAC: updated resemblance matrices using SLINK with quantitative data

Cluster {1,2,8}		Cluster {3,6,4}	
Neighbor	Coefficient	Neighbor	Coefficient
{5}	3.83	{1,2,8}	3.54
{3,6,4}	3.54	{7}	2.65
Member 2	1.94	Member 6	1.95
Member 8	2.51	Member 4	1.99
Cluster {5}		Cluster {7}	
Neighbor	Coefficient	Neighbor	Coefficient
{1,2,8}	3.83	{3,6,4}	2.65

lar result to the HAC, while providing faster clustering. Comparing the results of the HAC and DHAC, we notice that the first clustering step result of the DHAC is the same as the result of the second step of the HAC (see Fig. 3.3 and Fig. 4.2). The result of the second clustering step of DHAC is the result of the fourth step of the HAC (see Fig. 3.5 and Fig. 4.3). As mentioned before, there is no necessity to obtain global information because the far apart nodes would not be grouped together. With localized information, the clustering can be executed synchronously and therefore faster.

4.3.2 Cluster Formation with Qualitative Data

In Chapter 3, Table 3.2 and Table 3.4 demonstrated the input data for general clustering methods with qualitative data. Entry value '1' indicated a connection between two corresponding nodes. The only information that is missing for qualitative data is the total number of nodes. Without global knowledge, each node needs to know how

many '0's to fill in the connectivity table for nodes that are not directly connected. This can be pre-configured with a specific number depending on the total number of nodes. In other words, each node fills '1' for its one-hop nodes and '0' for all other nodes by using the pre-configured total number of nodes. Thus, the connectivity table based on one-hop knowledge has no difference with the table based on global knowledge.

By following the same procedure, each node can establish its distributed resemblance matrices. After establishing resemblance matrices, each CH performs DHAC as Fig. 4.1 illustrates. Fig. 4.4 shows a comparison between the centralized clustering and distributed clustering with qualitative data for the same 8-node network shown in Fig. 3.1. The results are very similar. However, we should note that the result obtained from the distributed algorithm may not be identical to that of the centralized clustering algorithm because of having less information.

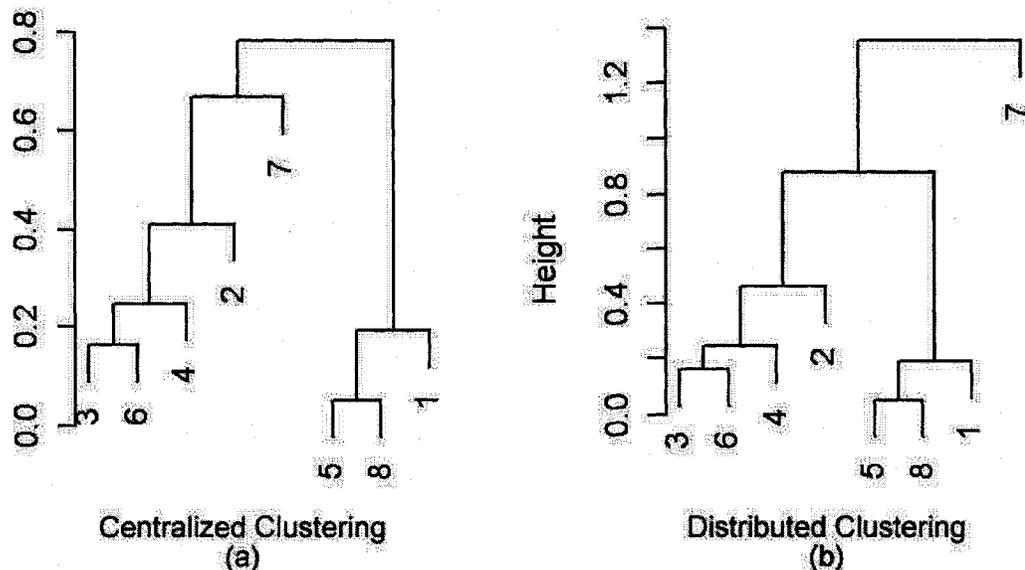


Fig. 4.4: Comparison of the HAC and DHAC using UPGMA with SORENSON qualitative data

4.4 DHAC: Cluster Maintenance

After clustering, each node gets its assigned role and starts to send data out. Cluster maintenance can also be seen as the steady-state phase which keeps the network working based on the established clusters.

4.4.1 Scheduling and Data Aggregation

In a cluster, the scheduling helps a cluster member know how to report data to its CH. With DHAC protocol, once clustering has stopped, the selected CHs broadcast schedules to their cluster members. DHAC uses the sequence of nodes merging into the current cluster as the scheduling tool to control cluster members sending data. In Fig. 3.7, the dendrogram illustrates the sequence of nodes merging. Notice that the sequence is a logical binary tree which indicates the location relationships of all the nodes.

Following received schedules, nodes start to report collected information to the sink via CHs. CHs respond to received data from each cluster member. After collecting all the data, CHs aggregate it into one message and send it to the sink. To provide high efficiency in data collection, DHAC adopts the “TDMA + CDMA” model as the MAC layer protocol. Fig. 4.5 shows the MAC layer protocol structure of DHAC. In the cluster formation phase, DHAC execution requires nodes to exchange messages based on the CSMA (Carrier Sense Multiple Access) MAC layer scheme. During the steady-state phase, DHAC adopts the TDMA frame as its MAC layer model for intra-cluster communication since it has the natural advantage of having no collision or control overhead. Within a cluster, the communication is broken into frames where cluster member nodes send their data to CHs at their one assigned time slot per frame. For inter-cluster communication, DHAC uses the CDMA MAC layer protocol to avoid the

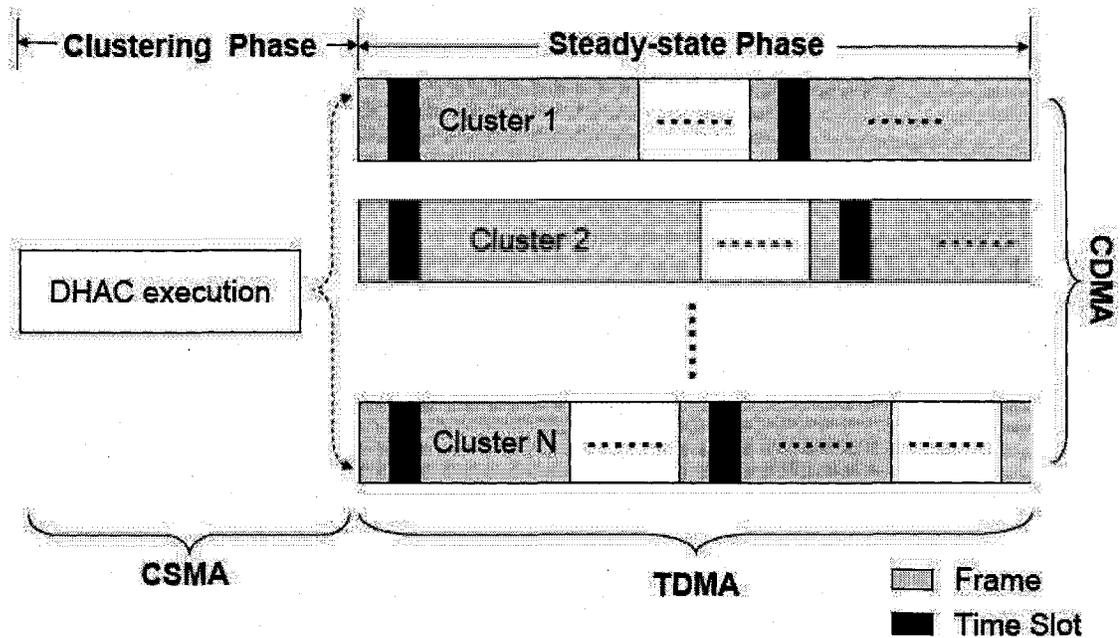


Fig. 4.5: The MAC layer structure of DHAC

collision of communications among different clusters. Within a frame, each time slot in which a node communicates with the CH is fixed. The frame size depends on the number of members in the cluster. In order to avoid the frame being too long or too short, DHAC uses the predefined Threshold and Minimum_Cluster_Size parameters to limit the cluster size by controlling cluster agglomeration. The threshold is the upper bound and the Minimum_Cluster_Size acts as the lower bound.

4.4.2 CH Rotation

Within a cluster, the CH manage cluster members by isolating intra-cluster communication. Furthermore, the CH collects the sensed data from cluster members, aggregates all the data and sends a single message to the sink. Since all the responsibilities of the CH require high energy dissipation, the CH role is rotated to different nodes, thus DHAC can make energy dissipate uniformly through the whole network.

To solve the issue of when and how to rotate CH role, two kinds of rotation methods are proposed: automatic CH rotation and rescheduling.

Automatic CH Rotation

Automatic rotation uses time to control the rotation. After the cluster has run for several frames, the CH role automatically passes to the next node in the schedule. When the CH role moves, every node still follows the same time slot, the previous CH taking a sleep at the end of its time slot and the new CH waking up at the appropriate time and collecting data from other cluster members. This CH rotation does not need any message exchange because every node of the cluster already has the schedule from when the cluster formatted. In other words, automatic CH rotation does not consume extra energy.

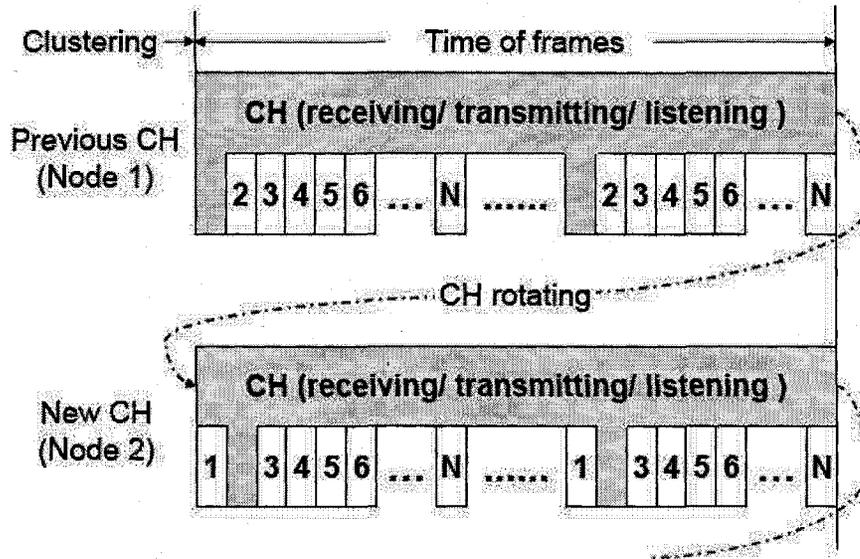


Fig. 4.6: An illumination: automatic CH rotation

Fig. 4.6 provides an example of automatic CH rotation. The cluster has a CH, node 1, and $N - 1$ cluster members. After clustering, node 1 acts as the CH and manages the cluster. The CH keeps communicating with cluster members during the white time

slots. The grey time slot is an empty slot to let the CH listen to information from outside the cluster, such as other nodes requesting to join the cluster. After several frames, the CH role automatically passes to the next node in the schedule. In Fig. 4.6, node 2 becomes the new CH. The corresponding time slot of node 2 becomes an empty slot and the other nodes still follow the same schedule. For the slot assignment within the communication frame DHAC uses the result from [42, page 92-94], which demonstrated how to choose the frequency of CH rotation. Equation 4.1, the result of [42, page 94], is used in our experiments to calculate the period of automatic CH rotation. E_{initial} is the average initial energy(J) of each node. $T(\text{seconds})$ is the time period of automatic CH rotation.

$$T(\text{seconds}) = 0.08s * \frac{E_{\text{initial}}(\text{J})}{0.009\text{J}} \quad (4.1)$$

Rescheduling

Rescheduling is designed to keep energy dissipation more uniform within the cluster. Under certain conditions, the CH informs cluster members of a new schedule. Then, cluster members start to follow the new schedule. There are two conditions that can trigger the rescheduling.

- A cluster has a change. The change can be a new node joining, or a node leaving or dead.
- The CH has lower energy than the threshold $Th(\text{energy})$. When cluster members send data to the CH, they also send their current residual energy. Based on the collected information, CH can calculate the average residual energy, $R(\text{energy})$,

within the cluster. Once the residual energy of the CH becomes lower than the $Th(\text{energy})$, the CH informs every cluster member of the new schedule. The new schedule is the sequence of the nodes in order of decreasing $R(\text{energy})$. Equation 4.2 presents the calculation of the threshold $Th(\text{energy})$ by using a predefined percentage of average residual energy, $P(\text{energy})$.

$$Th(\text{energy}) = P(\text{energy}) \times \frac{1}{n} \sum_{i=1}^n R(\text{energy})_i \quad (4.2)$$

Since the CH needs extra energy to inform cluster members of a new schedule, we need to find the balance between uniform energy dissipation and reducing cluster maintenance cost. In following Section 4.5, we will discuss how to set $P(\text{energy})$.

4.5 Evaluation of Schemes in DHAC

In previous sections, we presented some options or variants at different parts, i.e., variation points of the HAC algorithm. These options support more flexibility to accommodate different situations or technologies. Based on different environments, traffic patterns, and hardware, DHAC can choose appropriate methods and adjust predefined parameters to apply the HAC algorithm to WSNs. Fig. 4.7 presents possible variants for the different variation points for the DHAC approach.

This section includes more detailed investigation of DHAC and evaluates the effect of several parameters and methods that are critical to DHAC. We separate the evaluation into two parts, cluster formation and cluster maintenance.

Input data and resemblance coefficient calculation method	Quantitative		Location	
			RSS	
	Qualitative		Connectivity	Jaccard
				Sorenson
Simple Matching				
Clustering method	SLINK	CLINK	UPGMA	WPGMA
Cluster tree cut method	Number of clusters (C-NUMBER)		Height of cluster (C-HEIGHT)	
CH rotation method	Automatic rotation		Rescheduling	

Fig. 4.7: Variation points and possible variants of applying DHAC to WSNs

4.5.1 Evaluation of Cluster Formation

The investigation in the area of cluster formation includes cluster size and cluster range. The parameters considered in this section focus on quantitative versus qualitative data, and clustering methods.

We applied DHAC to a network of 100 nodes uniformly distributed in a $100m \times 100 m$ field. The DHAC merged nodes to form five clusters by using the UPGMA method. The several clusters are represented in Fig. 4.8 as star clusters with the CH (filled circle) leading the group. Fig. 4.8(a) and Fig. 4.8(b) show the clustering results with quantitative and qualitative data respectively. Obviously, the result of qualitative data is quite different from the result of quantitative data. However, different results are not important. What matters are the quality of clusters, clustering energy efficiency and other objectives.

To deeply discuss DHAC with different input data and methods, we use different

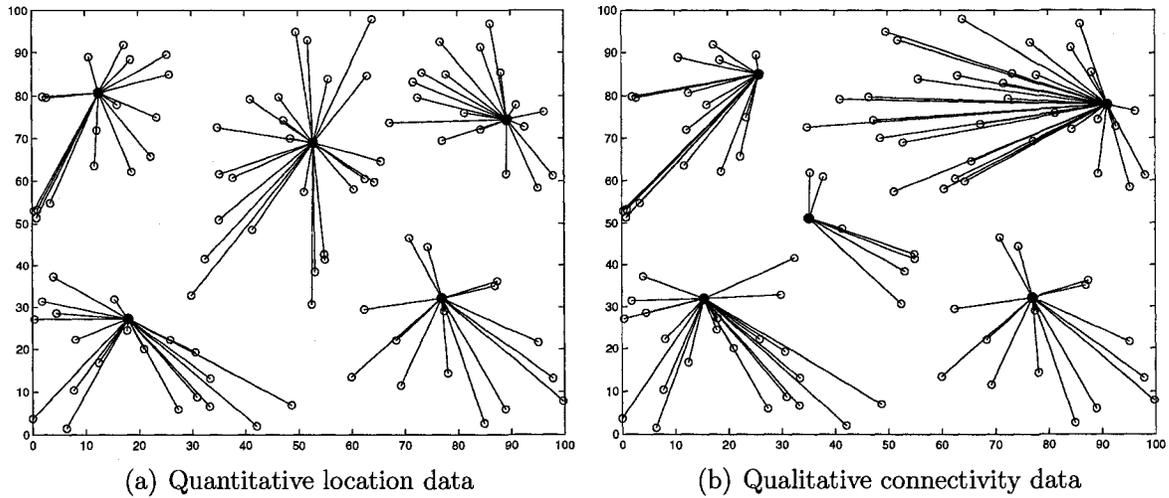


Fig. 4.8: DHAC using UPGMA for 100-node network with different data types

topologies with 100 nodes, uniformly distributed in a $100 \times 100 \text{ m}^2$ field with different node degrees (different transmission radius). The average node degree is used for all scenarios, and it will be referred to as node degree throughout this section. To ensure fair comparison of various HAC methods and reduce random effects caused by the simulation, we execute 30 simulation runs for each scenario and calculate the mean results.

DHAC with Quantitative Data

There are many the HAC algorithm methods. We demonstrate a comparison of four commonly used methods, SLINK, CLINK, WPGMA and UPGMA. For each method, DHAC is carried out with node location information. The clustering tree is then divided into groups using a predefined threshold of height of cluster, C-HEIGHT, as depicted in Fig. 4.7. The number of clusters decreases when the node degree increases. The cluster size can affect the node and network life. The first evaluation then is to investigate the effect of various clustering methods on the number of clusters for each node degree.

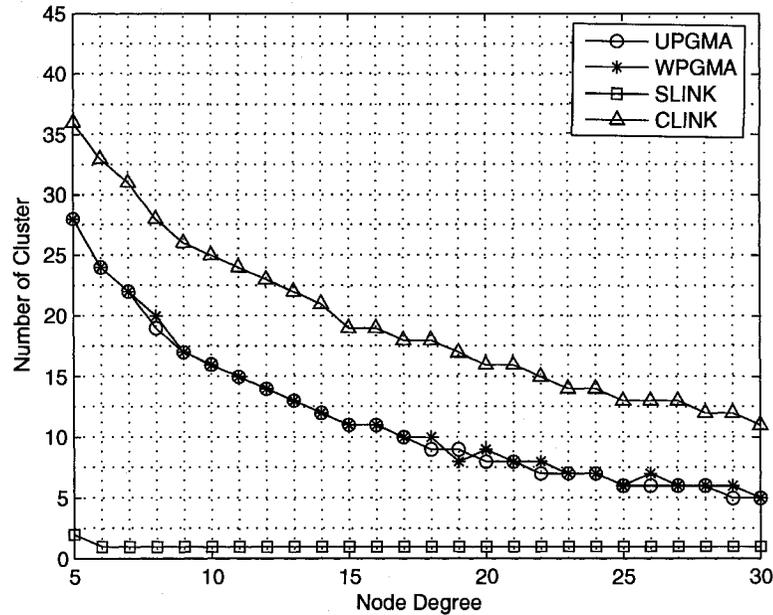


Fig. 4.9: Number of clusters versus node degree using different DHAC methods and quantitative location data

Fig. 4.9 shows the number of clusters against the node degree. SLINK and CLINK result in two extreme cases. SLINK tends to produce compact clusters. Fig. 4.9 indicates that only one cluster is generated when the node degree reaches 5. The result also reveals that the size of the clusters generated by SLINK is widely spread, ranging from 2 to 98 when the node degree is below 5. Unbalanced cluster size can cause some nodes to die quickly, while some nodes cannot report data in time due to longer delays in scheduling. On the other hand, CLINK produces many clusters. More clusters provide shorter network reaction time at the expense of heavy inter-cluster communications. Compared to SLINK and CLINK, UPGMA and WPGMA generate similar and more balanced numbers of clusters.

Cluster range is another important criterion to evaluate clustering methods. The cluster range is defined as the maximum distance between any two cluster members. A cluster with a smaller range can consume less energy in intra-cluster communi-

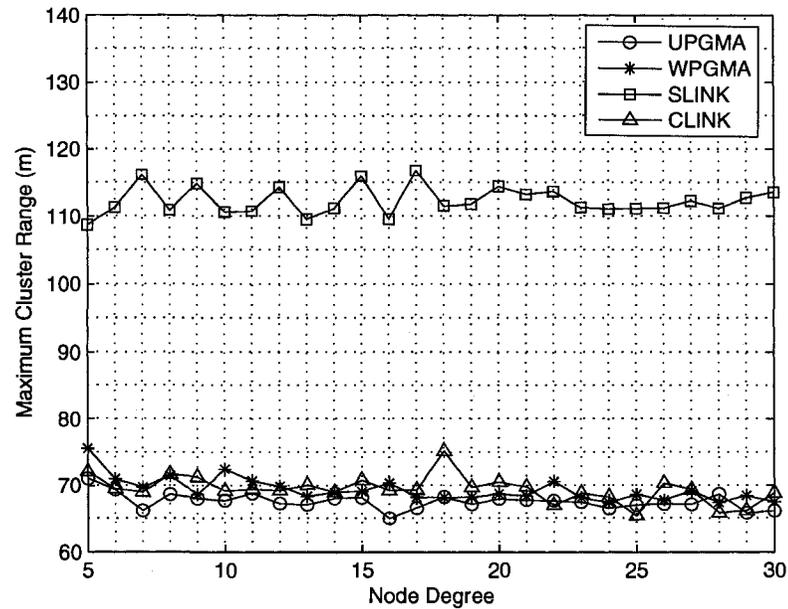


Fig. 4.10: Maximum cluster range versus node degree using different DHAC methods and quantitative location data

cations. Large cluster ranges should be avoided in a network, because the intra-communications cost can be high. Fig. 4.10 shows the maximum cluster range (in meters) versus the node degree for different algorithm methods. To compare the cluster range, we form 5 clusters in each network. It can be observed that the cluster range generally decreases when the node degree increases. Based on the experimental results, SLINK generates the largest cluster range while UPGMA produces the smallest cluster range.

DHAC with Qualitative Data

With the proposed DHAC, different types of network information, such as location, RSS and connectivity, can be used as input data. Using the node location information has two problems: higher device cost and more complicated localization protocols. RSS can replace the location information. RSS can be obtained from the antenna

without any extra processing; however, it is sensitive to the environment. Qualitative connectivity data presents another simple alternative to define the relationship among nodes. Fig. 4.11 compares DHAC with two different input data types, quantitative location data and qualitative connectivity data, with a fixed number of clusters. For qualitative connectivity data, we used the SORENSON method. Based on the experiment, quantitative data provides slightly smaller cluster ranges than qualitative data. While using qualitative data, UPGMA does not noticeably outperform CLINK and WPGMA as it does when quantitative data are used.

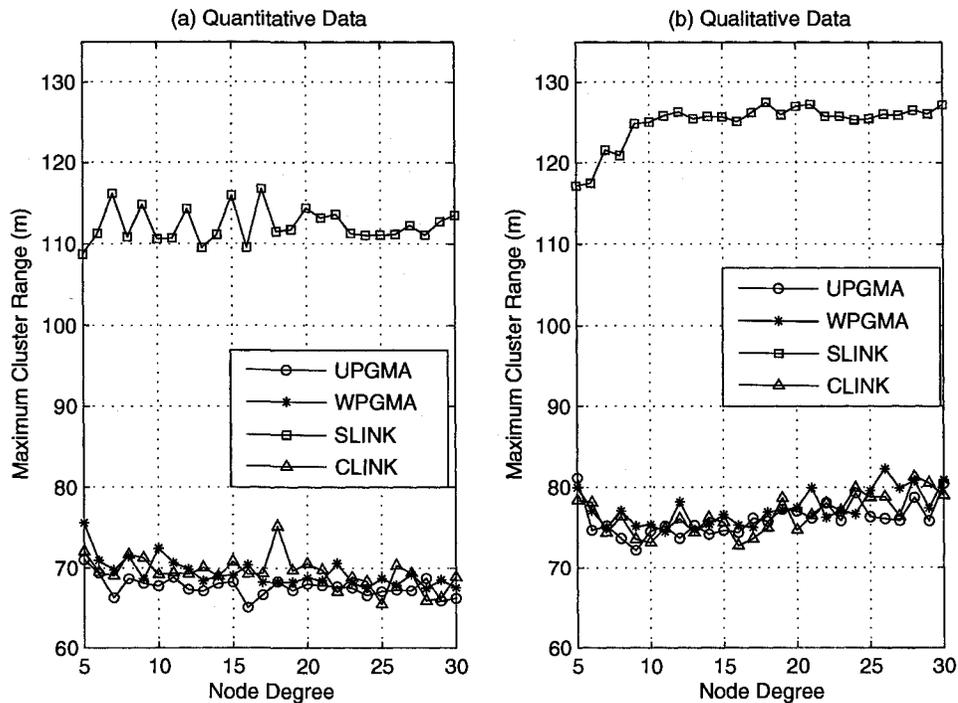


Fig. 4.11: Maximum cluster range versus node degree using quantitative and qualitative data

We further investigated the maximum cluster range using different resemblance calculation methods together with the qualitative connectivity data, as shown in Fig. 4.12. In addition, a comparison of those methods with the quantitative loca-

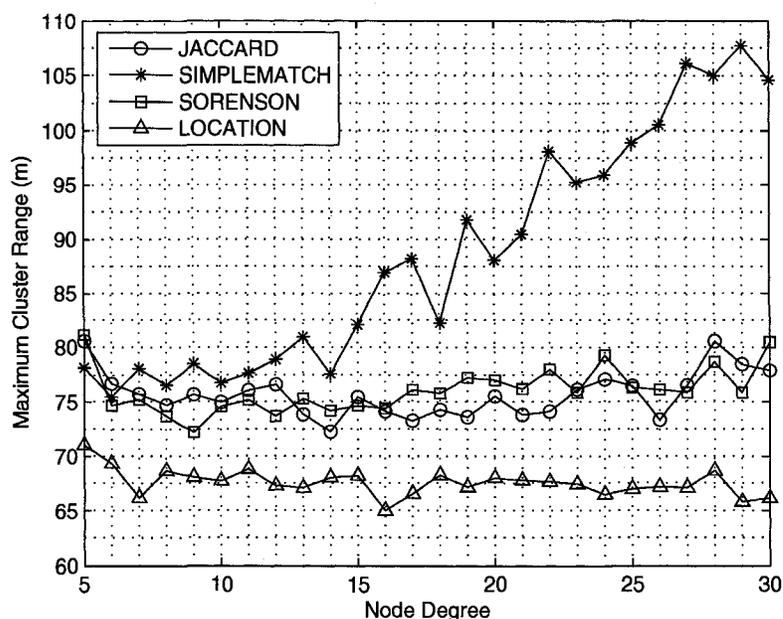


Fig. 4.12: Cluster range versus node degree with different resemblance calculation methods

tion data was conducted for evaluation. Many methods are available to calculate the resemblance coefficients. There are three well-known methods, JACCARD, SORENSON and simple matching. The UPGMA method was used due to the better results it produced in previous scenarios. The cluster tree is cut with the threshold of C-NUMBER. The cluster range drops with different rates when the node degree increases. All three methods using qualitative data generate a larger maximum cluster ranges than when using quantitative location data. The simple matching method generates the largest cluster range. When the node degree increases, the cluster range of simple matching method remains almost unchanged while other methods decrease with different rates. The result obtained from the SORENSON calculation is closer to that of quantitative location data than other methods.

4.5.2 Evaluation of Cluster Maintenance

As presented in Section 4.4.2, DHAC uses two schemes to rotate CH within clusters. Due to the rescheduling energy cost, DHAC needs to carefully choose the threshold, $Th(\text{energy})$, to avoid frequent rescheduling while achieving more uniform energy dissipation. In other words, more uniform energy dissipation and reduced energy dissipation can both be critical to the lifetime of network. Thus, the network lifetime is used to help determine the $Th(\text{energy})$. In Equation 4.2, DHAC needs to specify the percentage of average residual energy, $P(\text{energy})$, to calculate $Th(\text{energy})$.

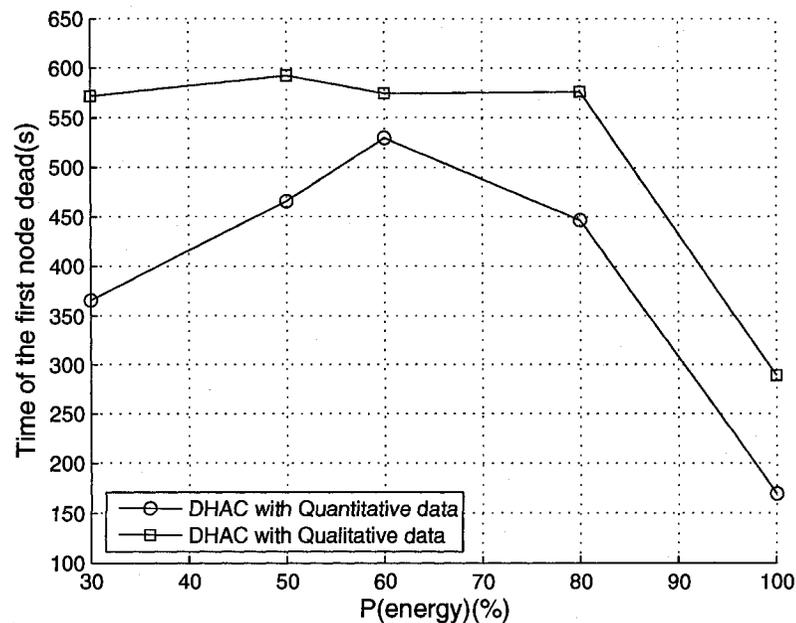


Fig. 4.13: Network lifetime of the first node dead versus $P(\text{energy})(\%)$

In Fig. 4.13, when $P(\text{energy})$ is changed from 30% to 80%, the DHAC with qualitative data maintain almost the same performance, whereas the DHAC with quantitative data suffers a large variation, reaching the best performance at $P(\text{energy})=60\%$. Once $P(\text{energy})$ is larger than 80%, the performance of both input data types quickly deteriorates. With both quantitative and qualitative data, $P(\text{energy})=60\%$ offers

good balance between re-scheduling energy cost and uniform energy dissipation. In the later experiments of Chapter 5, $P(\text{energy})$ is fixed at 60%.

Chapter 5

Performance Evaluation and Simulation Experiments

This chapter presents the performance comparison among the proposed DHAC, LEACH, and LEACH-C protocols. As presented in Chapter 2, LEACH is a typical fully-distributed protocol and LEACH-C is a centralized protocol; both are important hierarchical routing protocols. Simulation experiments are carried out in the network simulator NS-2 (version ns-2.29). To minimize the influence of random network deployment, protocols are executed in 10 different network topologies and the mean results are used for comparison.

5.1 Simulation Environment

The simulated WSNs consist of 100 homogeneous sensor nodes randomly deployed within the sensing field. Simulation parameters are shown in Table 5.1. Most of the parameters are similar to those in [11]. Using these parameters, a simple energy dissipation model of radio and processor hardware is presented as follows.

Table 5.1: Simulation parameters

Parameter	Value
Node number	100
Sensing field range	(0,0) to (100, 100)
Channel bandwidth	1 Mbps
Threshold distance(d_0)	75 m
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m ²
ϵ_{mp}	0.0013 pJ/bit/m ⁴
E_{fusion}	5 nJ/bit/signal
Data rate	5 TDMA frames per 10s
E_{initial}	1 J/node
Exchanging message size	
Data packet size	500 bytes
HELLO	
INVITE	25 bytes
SCHEDULE	
CONFIRM	225 bytes

- **Receiving energy dissipation.** The energy consumed by receiving 1 bit data, E_{elec} , depends on coding and modulation. The energy consumed by receiving an L bit message is thus given by

$$E_{\text{Rx}} = L \times E_{\text{elec}} \quad (5.1)$$

- **Transmitting energy dissipation.** Both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models are used [43, 44]. When the distance between the transmitter and receiver, D , is larger than a specified

threshold distance, d_0 , the channel switches to the multi-path fading model; otherwise the channel follows the free space model [42, page 84].

$$E_{Tx}(L, D) = L \times E_{elec} + \begin{cases} L \times \epsilon_{fs} \times D^2, & D < d_0 \\ L \times \epsilon_{mp} \times D^4, & D \geq d_0 \end{cases} \quad (5.2)$$

where $\epsilon_{fs} = 10\text{pJ/bit/m}^2$, $\epsilon_{mp} = 0.0013\text{pJ/bit/m}^4$, $d_0 = 75\text{m}$, and L is the number of bits in the transmitted message.

- **Computation energy dissipation.** Data aggregation and resemblance matrix updating cause computation energy dissipation [42, page 86]. E_{fusion} defines the computational costs of performing data calculation.

$$E_{com} = E_{fusion} \times Size_{signal} \times Number_{signal} \quad (5.3)$$

5.2 Experiments and Simulation Results

Three metrics are used to analyze and compare the performance of protocols: network lifetime, energy dissipation and the number of data packets received at the sink.

- **Network lifetime.** Network lifetime in the experiments is defined as the time between the beginning of a sensor network and the death of the last node. A node is considered as a dead node if it runs out of its initial energy, $E_{initial}$. There is no uniformly accepted definition of “network lifetime”. Usually researchers set the end time according to the objective of an application. In the following

section, we use $T_{n\%}$ to indicate the network lifetime when there are $n\%$ nodes dead.

- **Energy dissipation.** The energy dissipation of nodes relates to transmitting, receiving, listening and data aggregating. The main interest is the energy dissipation during clustering, which is used as the major criterion to evaluate hierarchical routing protocols. Since our focus is the network routing, not the network coverage, the energy dissipation during sensing is not included in the simulations.
- **Number of data packets received at the sink.** Given fixed E_{initial} , nodes exploit their resources to send data out to the sink. The number of data packets received at the sink is an important metric to compare the routing performance among different protocols. More data packets received at the sink is an indication that the protocol can provide better connections among nodes and the sink.

Our simulation results for clustering and energy saving are reported below. The above three metrics are evaluated for LEACH, LEACH-C and DHAC with different input data types. For simplification, following abbreviations are used to denote DHAC with different input data types.

- DHAC-LOC: the DHAC with LOCation quantitative data
- DHAC-RSS: the DHAC with RSS quantitative data
- DHAC-CON: the DHAC with CONnectivity qualitative data

Based on the results of clustering quality analysis in Chapter 4, the UPGMA method is applied to DHAC-CON and DHAC-RSS, and the CLINK method is applied

to DHAC-CON. To build the resemblance matrix using qualitative data, SORENSON is used to calculate the coefficients.

5.2.1 Sink Inside of the Network

In this experiment, the sink locates at the center of network, (50,50). For a network with a large number of uniformly distributed nodes, the center is a good estimation of the centroid. Simple calculus can show the sink at the centroid achieves the minimum mean squared Euclidean distance between nodes and the sink. Therefore, the main interests are consumed resources during clustering.

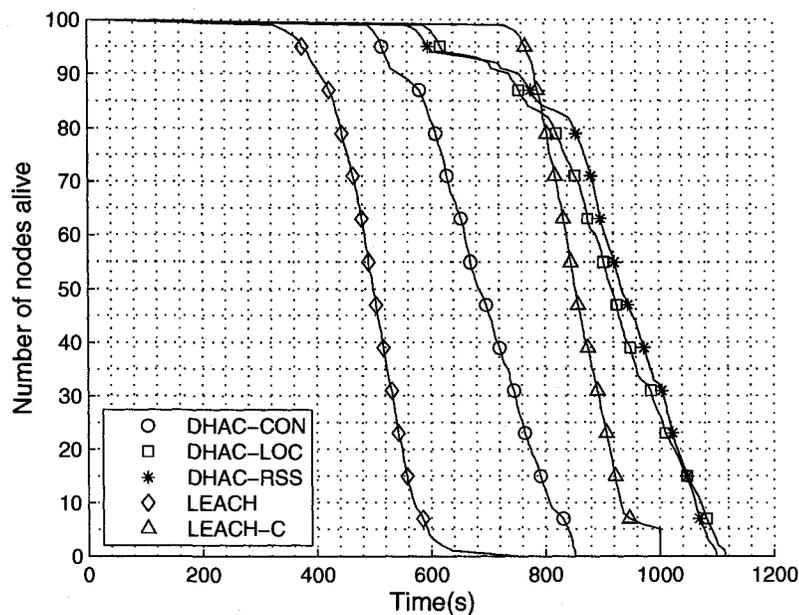


Fig. 5.1: Network lifetime, the sink at (50,50)

Fig. 5.1 shows the network lifetime of corresponding to different protocols. It can be observed that DHAC-LOC and DHAC-RSS have the longest $T_{100\%}$. LEACH has the shortest network lifetime: the $T_{1\%}$ is 364s and only 10% of the nodes are alive at 585s. Compared to LEACH, DHAC-CON prolongs $T_{1\%}$ by 13.5%. Compared to

DHAC, LEACH-C provides a longer lifetime than DHAC-CON but a shorter lifetime than DHAC-LOC and DHAC-RSS. Although LEACH-C has a smaller death rate at the beginning of network serving (until around 15% of the nodes are dead), DHAC-LOC and DHAC-RSS have more nodes alive the rest of the time. This indicates that during network serving, DHAC-LOC and DHAC-RSS provide better sensing coverage than LEACH-C. It can be observed in Fig. 5.1, DHAC-LOC and DHAC-RSS provide around 100s longer $T_{100\%}$ than LEACH-C.

In Fig. 5.1, DHAC-LOC and DHAC-RSS have shorter $T_{25\%}$ than LEACH-C and longer network life time for the rest. With DHAC, the node deaths are fast in the beginning because DHAC spends more energy on clustering at the initial phase. Although LEACH and LEACH-C spend less energy on clustering at the beginning, they require reclustered at the beginning of each round and spend more energy on clustering through the whole network lifetime.

For the different DHAC, the $T_{1\%}$ of DHAC-CON is 97s shorter than the other two, DHAC-LOC and DHAC-RSS. One reason is that DHAC-CON uses more energy during clustering to exchange neighbor lists. Another reason is that the quality of generated clusters in DHAC-CON is not as good as DHAC-LOC and DHAC-RSS, since the qualitative data contains less information than quantitative data. In Fig. 5.1, DHAC-LOC and DHAC-RSS have no noticeable difference in performance. Note that the performance of DHAC-LOC would be lower than in Fig. 5.1 once the energy dissipation of node localization is considered. To focus on network routing, throughout this thesis nodes are assumed to obtain their location information without extra energy dissipation. Although DHAC-RSS has good performance, RSS may not be precise enough for some applications. For example, if there are blocks in the network field, the signal strength no longer precisely indicates the distance between two nodes.

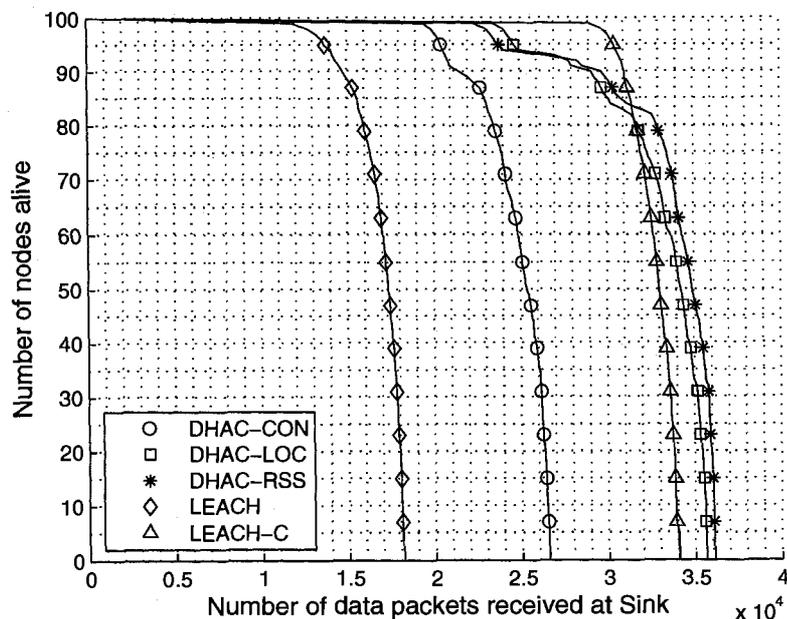


Fig. 5.2: Number of nodes alive versus the amount of data packets received at the sink(50,50)

Fig. 5.2 shows another perspective of the protocol performance: the number of data packets received at the sink during the network lifetime. More data from the network to the sink suggest the network better accomplishes its application duty. Similar to what is observed in Fig. 5.1, DHAC-LOC and DHAC-RSS outperform LEACH and slightly outperform LEACH-C. As shown in Fig. 5.2, when the last node died, the sink in DHAC-LOC and DHAC-RSS received approximately 36000 data packets, and DHAC-CON reached 26500 data packets; LEACH received 18000 packets and LEACH-C received 34000 packets.

The third metric is the energy dissipation during clustering. The less energy dissipation of clustering and cluster maintenance, the more energy can be used to perform network duty. In fact, the better performance of hierarchical routing protocols comes from less energy dissipation for building or managing clusters. Every hierarchical routing protocol strives for the least energy dissipation during clustering, reclustering

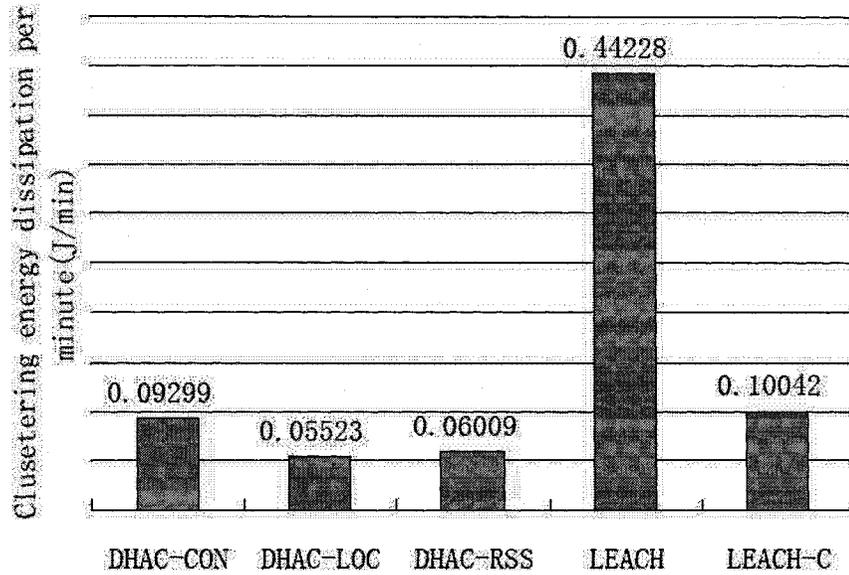


Fig. 5.3: Total clustering energy dissipation per minute, the sink at (50,50)

or rescheduling. Fig. 5.3 illustrates the average clustering (includes reclustering and rescheduling) energy dissipation over 100 nodes during the whole network lifetime. Total clustering energy dissipation per minute is given by

$$\frac{\text{Total dissipated energy during clustering}(J)}{\text{Time of the last node dead}(Min)}$$

LEACH spends 0.44228 J/min to execute clustering and reclustering, while DHAC uses at least 3 times less energy. LEACH-C has slightly more energy dissipation than DHAC. This explains why DHAC and LEACH-C offer longer network lifetime than LEACH.

To further investigate energy efficiency, we examined the relationship between total energy dissipation and the amount of data packets received at the sink. The total energy dissipation includes both clustering energy dissipation and the energy dissipation of transmitting data from network to the sink. As shown Fig. 5.4, higher slopes signify higher efficiency since nodes send out more data with less energy. DHAC-LOC

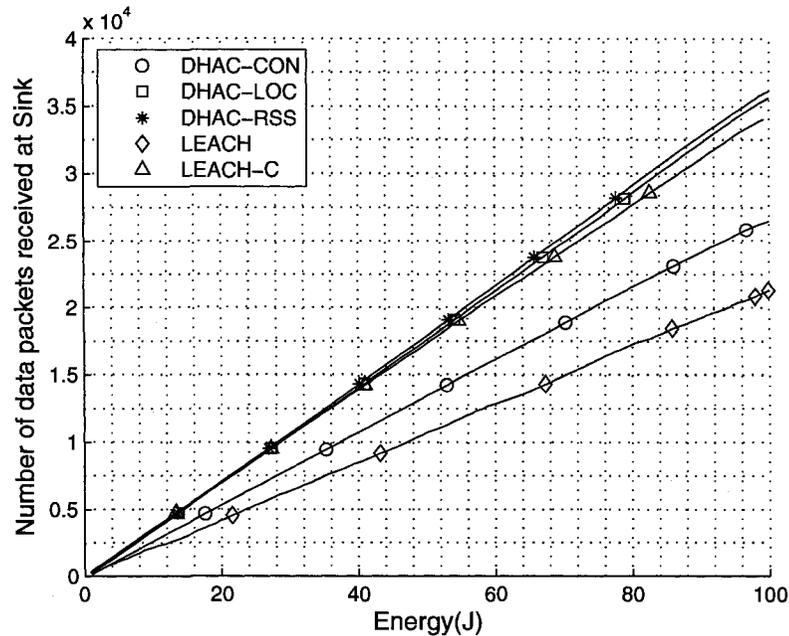


Fig. 5.4: Total energy dissipation versus the amount of data packets received to the sink(50,50)

and DHAC-RSS offer the best energy efficiency, while LEACH provides the worst efficiency. LEACH-C still performs better than DHAC-CON and slightly worse than DHAC-LOC and DHAC-RSS.

The above comparisons indicate that when the sink is located at the center of the network, DHAC performs much better than LEACH, and the performance of DHAC with quantitative data is better than LEACH-C. As expected, DHAC with quantitative data performs much better than the DHAC with qualitative data because using qualitative data requires more information exchanging to build the resemblance matrix and the qualitative data have less precise relationship descriptions.

5.2.2 Sink Outside of the Network

As introduced in Section 2.1.1, the sink is usually deployed far from the network field. In this experiment, the sink locates at (50, 300) while the network field spans from

(0,0) to (100,100). The distance between the sink and the center of network is 250m. When the sink is located far from the network, network nodes spend more energy to communicate with it.

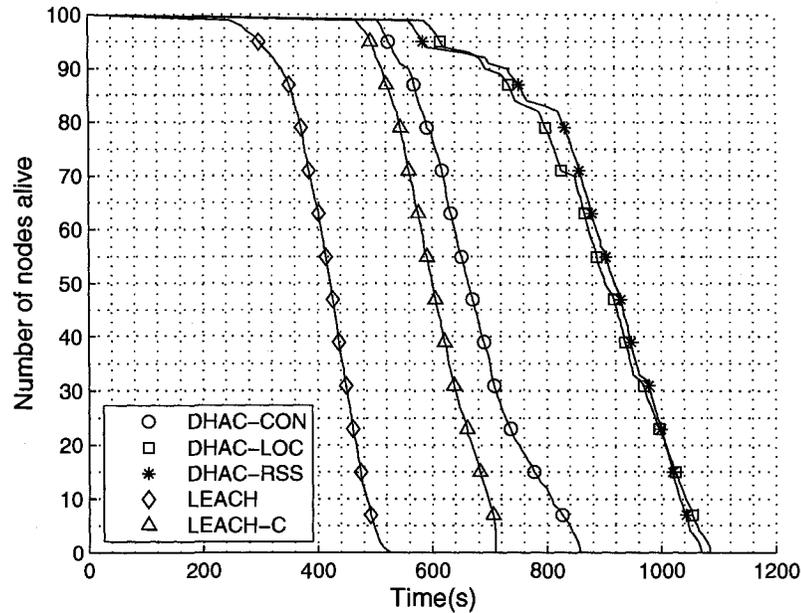


Fig. 5.5: Network lifetime, the sink at (50,300)

When the sink is moved outside of the network, the performance of LEACH-C quickly decreases because a centralized protocol has to depend on the communication between the network and the sink. When the nodes need to communicate with the far away sink, the centralized protocol has to offer lower performance than the distributed approach because the communication cost is much higher. It brings the WSNs deployment another constraint if the network application depends on the location or capability of the sink. In Fig. 5.5, DHAC provided longer network lifetime than either LEACH or LEACH-C. The $T_{100\%}$ of LEACH-C is reduced to 709s from 1000s when the sink is located at (50,50). In Fig. 5.6, LEACH-C has 29% reduction of the amount of data packets received at the sink than previous experiments, while

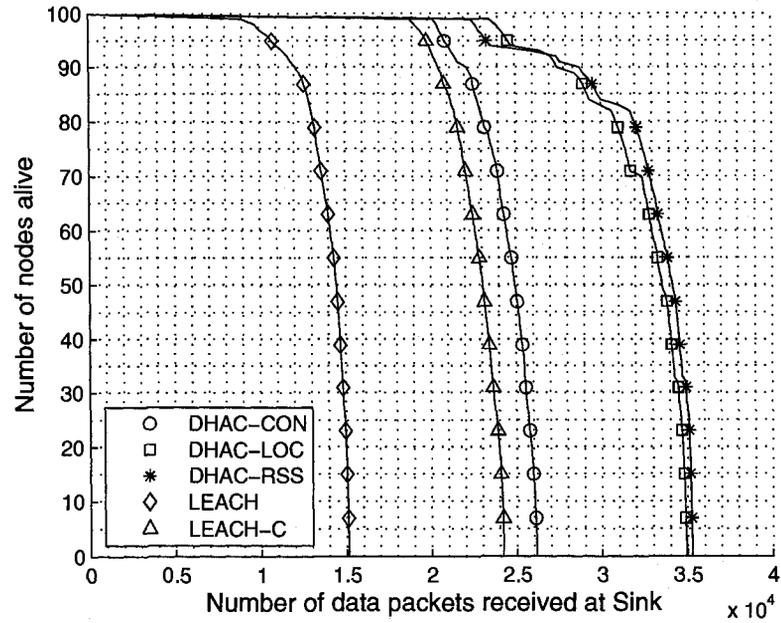


Fig. 5.6: Number of nodes alive versus the amount of data packets received at the sink(50,300)

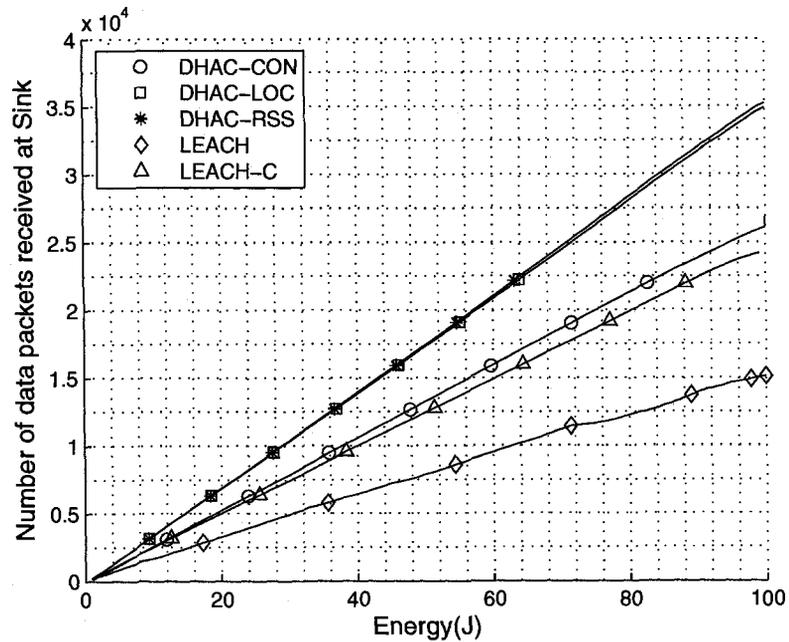


Fig. 5.7: Total energy dissipation versus the amount of data packets received to the sink(50,300)

other protocols only have slight reductions in the data received at the sink.

In Fig. 5.7, the energy efficiency of all protocols are slightly reduced except that LEACH-C become unusable when moving the sink far from the network. DHAC-LOC and DHAC-RSS still provide the highest efficiency, and the LEACH protocol still has the lowest efficiency. LEACH-C provides much lower energy efficiency than when the sink is located inside the network because LEACH-C has less energy left to perform network duty while spending more energy for cluster formation.

Wherever the sink is located, DHAC always provides stable and better performance in terms of network lifetime, energy dissipation and data reporting. As a centralized protocol, LEACH-C only follows the instructions from the sink by using the sink to execute the algorithm and build clusters. Obviously, a centralized scheme gives worse performance when the sink is far from the network nodes. On the contrary, a distributed protocol like DHAC shows more flexibility and higher efficiency by adapting to different network deployments. Compared to another distributed protocol, LEACH, DHAC provides 200% longer $T_{1\%}$.

5.2.3 Different Sink Locations

The location of the sink determines the communication cost of sending data out of the network to end users. Moving the sink can cause huge changes of energy dissipation. This section investigates the effects of sink locations by moving the sink from (50,50) to (50, 450) while fixing the X coordinate as 50. The results on network lifetime are shown from Fig. 5.8 to Fig. 5.11. The end points are used to define the $T_{20\%}$, $T_{50\%}$, $T_{80\%}$ and $T_{100\%}$ respectively.

From above figures, while the sink is moving further away, DHAC-CON and DHAC-RSS keep approximately a factor of 2 times the lifetime compared to LEACH,

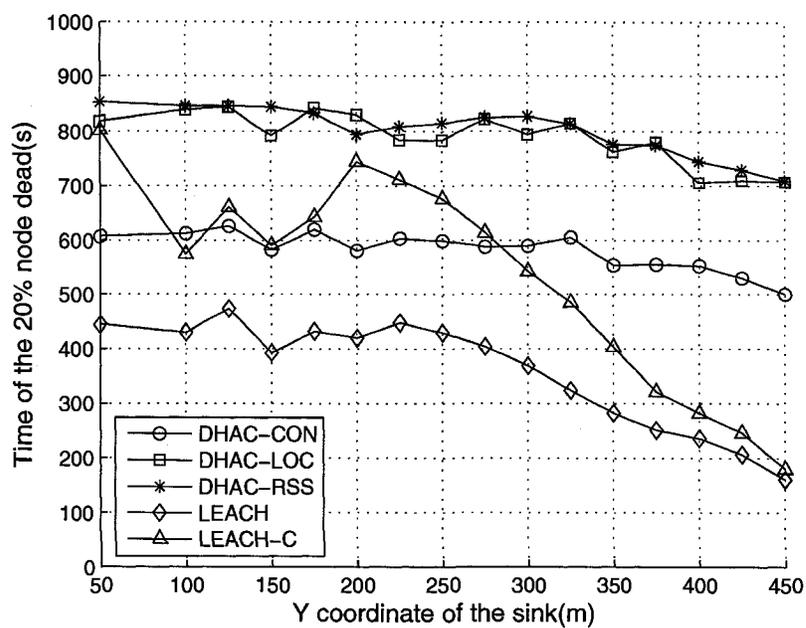


Fig. 5.8: Time of the 20% nodes dead against the sink location

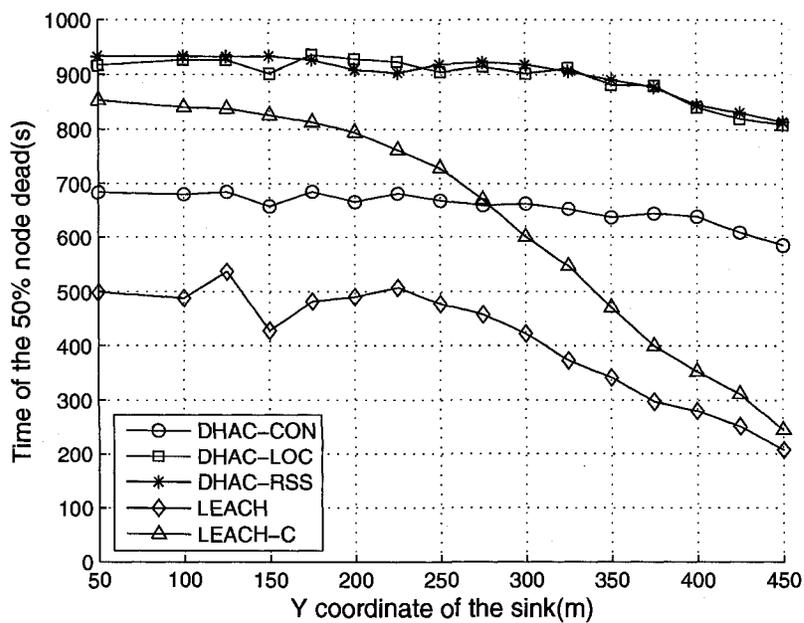


Fig. 5.9: Time of the 50% nodes dead against the sink location

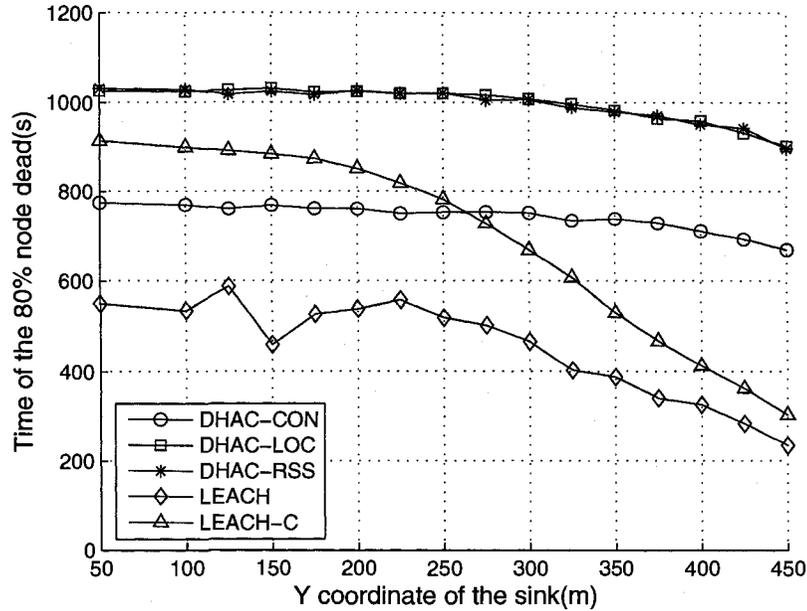
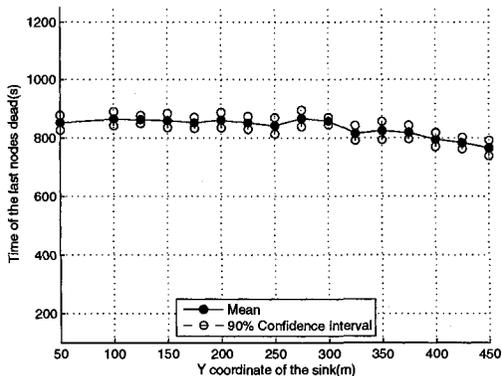


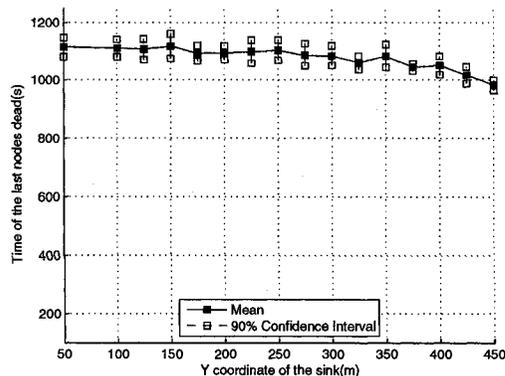
Fig. 5.10: Time of the 80% nodes dead against the sink location

and DHAC-CON has a factor of 1.5 improvement. While the sink moves further, the network lifetime of LEACH-C decreases very quickly. After the sink moves further than (50, 300), the performance of LEACH-C becomes lower than DHAC.

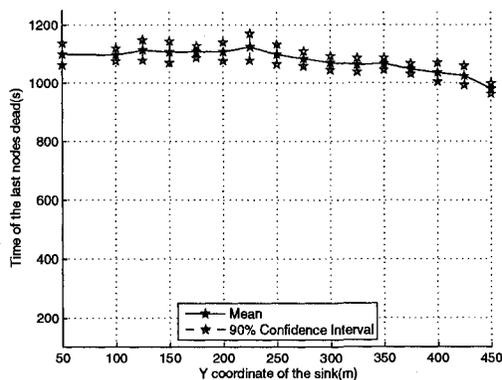
Fig. 5.11 shows the time of the last node dead against the location of the sink; each sub-figure corresponds to one clustering protocol. Solid marks indicate sample mean values μ , and hollow marks show two-sided 95% confidence interval, $\mu \pm \delta$ based on 10 different topologies. Since each sample mean μ only used 10 samples and we have no *a priori* knowledge on the standard deviation of the population, the confidence interval is related to Student's *t*-distribution rather than Gaussian distribution. Therefore δ is given by $t_{(1-\alpha)/2,v} \frac{\hat{\sigma}}{\sqrt{N_s}}$, where $N_s = 10$ is the number of samples, $\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{N_s} (T_i - \mu)^2}{n-1}}$ is the estimated standard deviation of the population, and $t_{(1-\alpha)/2,v} = t_{0.05,9} = 2.262$ is a two-sided critical value for a *t*-distribution (in our case, the confidence level $\alpha = 95\%$ and the freedom of *t*-distribution $v = n - 1 = 9$). Thus, the two-sided 95% confidence



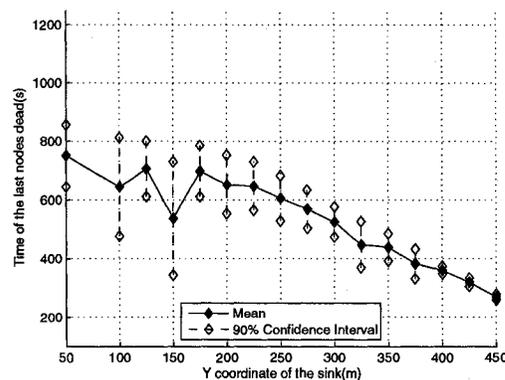
(a) DHAC-CON



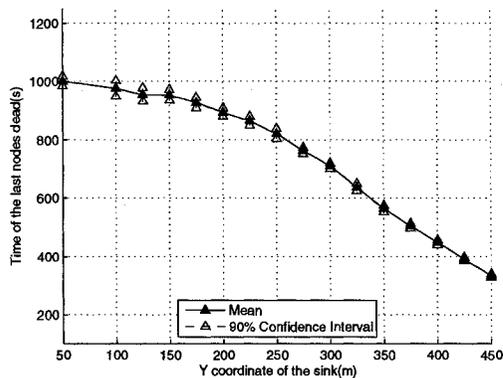
(b) DHAC-LOC



(c) DHAC-RSS



(d) LEACH



(e) LEACH-C

Fig. 5.11: Time of the last nodes dead against the sink location

interval is $\mu \pm 2.262 \frac{\hat{\sigma}}{\sqrt{10}}$. Note the population distribution of "Time of % nodes dead" is still assumed to be Gaussian. Refer to [45, page 146-155] for details.

Given the fixed confidence level $\alpha = 95\%$, Fig. 5.11(d) shows LEACH has the widest confidence intervals due to its randomness on CH selection, while LEACH-C, shown in Fig. 5.11(e), has the narrowest confidence intervals than others. When the sink moves, DHAC provides more stable performance than LEACH and LEACH-C.

5.2.4 Different Data Rates

In previous experiments, the data rate is fixed at 5 TDMA frames every 10 seconds: each node sends data 5 times to the CH or sink every 10 seconds. This section examines the effects of changing the data rate from the minimum, 1 frame per 10s, to the maximum, 15 frames per 10s. The length of data is fixed at 500 bytes.

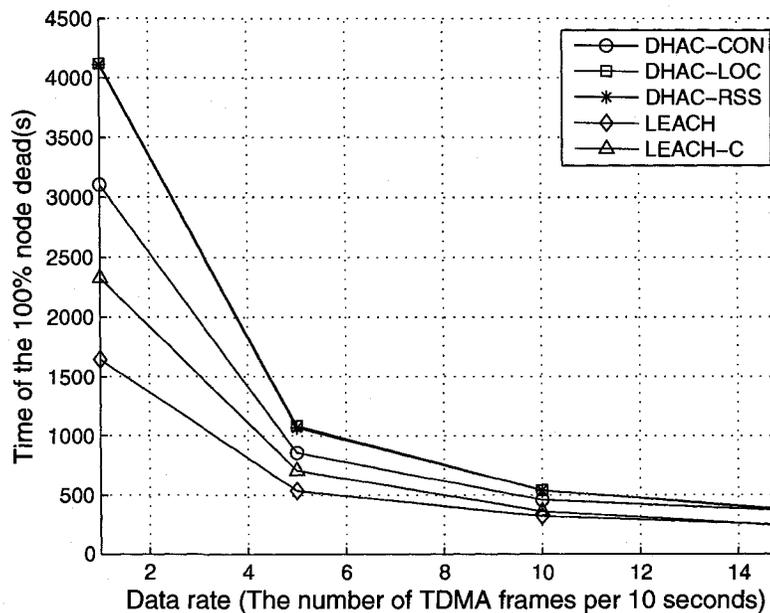


Fig. 5.12: Time of the last node dead with different data rate

Fig. 5.12 uses $T_{100\%}$ to indicate the network lifetime. Compared to LEACH and

LEACH-C, DHAC provides better performance when the data rate is low. When the data rate reaches 15 frames per 10s, DHAC has more than 150s longer $T_{100\%}$ than LEACH and LEACH-C. When the data rate decreases to only 1 frame per 10s, DHAC-LOC and DHAC-RSS have approximately 2500s longer $T_{100\%}$ than LEACH. Since LEACH and LEACH-C use a fixed period to control CH rotation, the reclustering does not consider if the CH has enough energy to coordinate its cluster. As a result, LEACH and LEACH-C spend the energy to do clustering at certain fixed times. The CH rotation scheme of DHAC considers the residual energy of the CH. When energy dissipation is not fast and residual energy of CH is enough, DHAC requires less rescheduling actions and consumes less energy to do rescheduling than LEACH and LEACH-C which do the reclustering in every round.

5.2.5 Different Node Initial Energy

Previous experiments use 1J/node as the E_{initial} . In this section, the E_{initial} is increased to 2J/node and 3J/node to study the influence of E_{initial} . The sink is located at (50,300).

Table 5.2 shows that DHAC provides similar performance regarding network life for different value of E_{initial} . DHAC shows slightly better performance than LEACH and LEACH-C when the first node dies, and approximately 200% longer $T_{100\%}$. As E_{initial} is increased, DHAC has longer and longer network lifetimes than LEACH and LEACH-C. When the E_{initial} is 1J, DHAC-RSS has 532s longer $T_{100\%}$ than LEACH; the difference increases to 1452s when the E_{initial} reaches 3J. Since LEACH and LEACH-C have to recluster periodically, the longer the network runs, the more energy is dissipated during clustering. DHAC only executes clustering once. And by combining automatic CH rotation and rescheduling, DHAC can achieve more uni-

Table 5.2: The time of 1%, 20%, 50%, 80% and 100% node dead with different E_{initial}

Energy (J/node)	Protocol	1%	20%	50%	80%	100%
1	LEACH	269	390	437	477	538
	LEACH-C	469	543	600	669	710
	DHAC-CON	506	590	663	752	858
	DHAC-LOC	587	794	902	1008	1085
	DHAC-RSS	560	828	919	1007	1070
2	LEACH	445	686	798	873	980
	LEACH-C	838	994	1108	1235	1287
	DHAC-CON	717	1050	1198	1344	1491
	DHAC-LOC	1001	1411	1606	1745	1988
	DHAC-RSS	979	1361	1561	1771	1996
3	LEACH	653	1003	1161	1279	1433
	LEACH-C	1224	1440	1610	1788	1886
	DHAC-CON	1156	1662	1776	1900	2250
	DHAC-LOC	1483	2078	2340	2607	2884
	DHAC-RSS	1417	2068	2366	2584	2885

form energy dissipation. Notice that the DHAC-CON has slightly shorter $T_{1\%}$ than LEACH-C because DHAC-CON needs to spend more energy on clustering at the beginning.

Chapter 6

Conclusions and Future Work

Clustering has been a topic of interest in many different disciplines for a long time. To adapt to the constraints of WSNs, clustering has generated a lot of discussion. This thesis advocated the application of well-known the HAC algorithm methods to WSNs and proposed a distributed approach, DHAC, to classify sensor nodes into appropriate groups instead of simply gathering nodes by their distance to some randomly selected CHs. We demonstrated the application and evaluation of four well-understood HAC algorithm methods, SLINK, CLINK, UPGMA and WPGMA. Based on the study of the HAC algorithm methods, we illustrated how to use the DHAC approach to mitigate the problems encountered with current protocols, and we supported the analysis by simulation.

The simulation model uses networks with a fixed number of nodes uniformly distributed in a square field. DHAC is compared to two well-known protocols, LEACH and LEACH-C, with three criteria, network lifetime, energy dissipation and the number of data packets received at the sink. Simulation scenarios use different sink locations, initial energy of nodes, and data rate. The results show that DHAC outperforms LEACH in all criteria. Compared to LEACH-C, DHAC with quantitative data pro-

vides better performance in all criteria, and DHAC with qualitative data provides better performance than LEACH-C when the sink is located far from the network area.

6.1 Future Work

We believe the following topics are interesting to explore.

- Improve the inter-cluster communication scheme. Select a subset of member nodes as gateways to serve inter-cluster communications.
- As a flexible clustering approach, DHAC can easily adapt other improved protocols based on LEACH. For example, DHAC can easily implement the cluster chain formation from PEGASIS by using the hierarchical tree from clustering.
- Investigate mobility issues for more complicated scenarios, such as moving sink and mobile sensor nodes.
- Develop cross-layer design. DHAC can be combined with MAC layer protocol to improve energy efficiency. DHAC parameters can also be adjusted to suit different WSN applications.
- Handle mixed data, quantitative and qualitative data. When some of network nodes have location knowledge and others only know their neighbors, DHAC can combine two different types of data to build optimized clusters.

References

- [1] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole, "Research challenges in environmental observation and forecasting systems," in *Proc. ACM/IEEE 6th Int. Conf. Mobile Computing and Networking (MobiCom)*, Boston, USA, Aug. 2000, pp. 292–299.
- [2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2 supplement, pp. 20–41, 2001.
- [3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM 1st Int. Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, 2002, pp. 88–97.
- [4] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao, "Target classification and localization in habitat monitoring," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, China, Apr. 2003, pp. 6–10.
- [5] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proc. ACM/IEEE 7th Int. Conf. Mobile Computing and Networking (MobiCom)*, Rome, Italy, 2001, pp. 151–165.
- [6] A. N. Knaian, "A wireless sensor network for smart roadbeds and intelligent transportation systems," Ph.D. dissertation, Massachusetts Institute Of Technology, Cambridge, USA, 2000.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [8] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun. Mag.*, vol. 11, no. 6, pp. 6–28, 2004.
- [9] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.
- [10] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Hawaii*

- Int. Conf. on System Sciences (HICSS)*, Wailea Maui, Hawaii, USA, Jan. 2000, p. 10.
- [11] —, “An application-specific protocol architecture for wireless microsensor networks,” vol. 1, no. 4, 2002, pp. 660–670.
- [12] M. R. Anderberg, *Cluster Analysis for Applications*, ser. Monographs and Textbooks on Probability and Mathematical Statistics. New York: Academic Press, Inc., 1973.
- [13] B. Everitt, *Cluster Analysis*. London: Heinemann Educational Books Ltd., 1980.
- [14] H. C. Romesburg, *Cluster Analysis for Researchers*. Malabar, Florida: Krieger Publishing Company, 1990.
- [15] 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,” in *Proc. ACM/IEEE 7th Int. Conf. Mobile Computing and Networking (MobiCom)*, Rome, Italy, 2001, pp. 272–287.
- [16] M. A. M. Vieira, J. C. N. Coelho, J. D. C. da Silva, and J. M. da Mata, “Survey on wireless sensor network devices,” in *Proc. IEEE 9th Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, Lisbon, Portugal, 2003, pp. 537–544.
- [17] V. S. Hsu, J. M. Kahn, and K. S. J. Pister, “Wireless communications for smart dust,” in *Electronics Research Laboratory Technical Memorandum Number M98/2*, Feb. 1998.
- [18] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next century challenges: Mobile networking for “smart dust”,” in *Proc. ACM/IEEE 5th Int. Conf. Mobile Computing and Networking (MobiCom)*, Seattle, Washington, USA, Aug. 1999, pp. 271–278.
- [19] 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,” in *Proc. 24th European Solid-State Circuits Conf. (ESSCIRC)*, 1998, pp. 9–16.
- [20] J. M. Rabaey, M. J. Ammer, J. J. L. da Silva, D. Patel, and S. Roundy, “Picoradio supports ad hoc ultra-low power wireless networking,” *IEEE Computer*, vol. 33, no. 7, pp. 42–48, 2000.
- [21] Crossbow Technology, Inc. (2007) 2007 wireless sensor networks product reference guide. [Online]. Available: <http://www.xbow.com>
- [22] Freescale Semiconductor, Inc. (2007) Freescale Sensor Products datasheet. [Online]. Available: <http://www.freescale.com>

- [23] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: A survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, 2006.
- [24] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, vol. 3, New York, USA, 2002, pp. 1567–1576.
- [25] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks," in *1st Int. Workshop on Networked Sensing Systems (INSS)*, Tokyo, Japan, 2004.
- [26] M. I. Brownfield, K. Mehrjoo, A. S. Fayez, and I. N. J. Davis, "Wireless sensor network energy-adaptive MAC protocol," in *Proc. IEEE 3rd Consumer Communications and Networking Conf. (CCNC)*, Las Vegas, Nevada, USA, Jan. 2006, pp. 778–782.
- [27] IEEE 802.15 Task Group 4. (2005) IEEE 802.15.4-part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). [Online]. Available: <http://www.ieee802.org/15>
- [28] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [29] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. ACM/IEEE 5th Int. Conf. Mobile Computing and Networking (MobiCom)*, Seattle, Washington, USA, Aug. 1999, pp. 174–185.
- [30] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proc. IEEE INFOCOM*, Hong Kong, China, 2004, pp. 629–640.
- [31] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM/IEEE 6th Int. Conf. Mobile Computing and Networking (MobiCom)*, Boston, USA, Aug. 2000, pp. 56–67.
- [32] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.
- [33] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Commun. Mag.*, vol. 7, no. 5, pp. 16–27, 2000.
- [34] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *ACM SIGMOD Record*, vol. 31, no. 3, pp. 9–18, 2002.

- [35] H. Chen, C. S. Wu, Y. S. Chu, C. C. Cheng, and L. K. Tsai, "Energy residue aware (ERA) clustering algorithm for leach-based wireless sensor networks," in *2nd Int. Conf. Systems and Networks Communications (ICSNC)*, Cap Esterel, French Riviera, France, Aug. 2007, p. 40.
- [36] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," in *Proc. 15th Int. Parallel and Distributed Processing Symp. (IPDPS)*, San Francisco, USA, Apr. 2001, pp. 2001–2008.
- [37] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerospace Conf.*, vol. 3, Big Sky, Montana, USA, 2002, pp. 1125–1130.
- [38] D.-H. Nam and H.-K. Min, "An efficient ad-hoc routing using a hybrid clustering method in a wireless sensor network," in *Proc. IEEE 3rd Int. Conf. Wireless and Mobile Computing, Networking and Communications (WiMOB)*, White Plains, New York, USA, Oct. 2007, p. 60.
- [39] I. Stojmenović, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 1, pp. 14–25, 2002.
- [40] Y. Yin, J. Shi, Y. Li, and P. Zhang, "Cluster head selection using analytical hierarchy process for wireless sensor networks," in *Proc. IEEE 17th Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, Helsinki, Finland, 2006, pp. 1–5.
- [41] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [42] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute Of Technology, Cambridge, USA, 2000.
- [43] T. Rappaport, *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA, 1996.
- [44] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 85–97, 1998.
- [45] S. Bernstein and R. Bernstein, *Schaum's Outline of Elements of Statistics I: Descriptive Statistics and Probability*. McGraw-Hill Professional, 1999.