國立交通大學

資訊科學與工程研究所

碩士論文

可用於大型無線感測網路之節能式空間關聯性資料收集演算法

Energy Efficient In-Network Spatial-Correlated Data Gathering Algorithm in Large Scale Wireless Sensor Networks

研 究 生：顏志安

指導教授：黃俊龍　教授

中華民國九十八年九月

可用於大型無線感測網路之節能式空間關聯性資料收集演算法

Energy Efficient In-Network Spatial-Correlated Data Gathering Algorithm in Large Scale Wireless Sensor Networks

研究生：顏志安　　Student：Yan, Jhih-An

指導教授：黃俊龍　　Advisor：Huang, Jiun-Long

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Department of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

September 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年九月

摘要

由於無線感測網路具有輕巧以及可大量佈建的特質, 應用無線感測網路來做環境監測以及災害預防已成爲一種可預見的趨勢。目前的無線感測器受限於能源的容量, 以及大型無線感測網路中能源替換的困難度, 其網路的使用時間相當有限, 因此如何減少感測器的能源消耗延長無線感測網路的使用時間是許多研究人員相當關心的問題。先前有許多研究觀察到溫度, 濕度等自然讀數都具有空間關聯性, 即在空間上擁有相近位置的無線感測器其感測讀數亦會相近, 根據此一特性可以把讀數相近的集合成一群集, 每一群集僅由一個感測器負責回報讀數降低網路中所需傳輸的訊息數量進而達到延長網路使用時間的效果。

在建立群集的過程中, 感測器之間必須互相交換訊息或是由一主機負責收集網路中的資料。由於此一過程會增加感測器額外的負擔, 因此如何降低建立群集時的能源消耗以及有效率的調整群集的結構, 爲這類群集演算法的重要問題。一些典型的分散式或集中式演算法針對資料之間的空間關聯性提出了建立群集的方式, 但是在這些演算法中, 對於群集的結構調整以及演算法的可擴充性都沒有辦法有效的解決。有鑑於此, 我們分析群集的特性並且設計了一個 In-Network的群集建立演算法 (ISCDG), 首先定義回傳訊息和接收者之間的空間關聯性, 並且對兩個具有高度空間關聯性的群集進行合併的檢查, 由於我們的演算法僅在網路中局部進行且可以有效率調整群集的結構, 因此可以用於大型的無線感測網路。在實驗的章節, 我們比較了我們的演算法與典型的分散式以及集中式演算法之間的效能差異, 也說明了 In-Network 的做法的確可以增加演算法的可擴充性。

Abstract

Many popular applications are developed on wireless sensor networks such as environment monitoring and disaster detection. The first characteristic of these applications is that sink periodically gather data from entire wireless sensor network, it makes the battery-powered sensors run of their energy fast. The other characteristic is that some error is accepted by user. According to these characteristics, an energy efficient way to gather approximately result can be a solution to extend lifetime of wireless sensor networks.

Some clustering algorithm were published to cluster sensor nodes, and let cluster heads are in charge of reporting sensing data to reduce the total number of reporting messages. These clustering algorithm consider the spatial correlation of sensing data to provide an user-tolerable approximately result. But the energy cost on the process of clustering and cluster maintenance may add some overhead to entire WSN. The other problem is that spatial correlation can not be used effectively since energy of sensor device is limited, they can not communicate with close nodes frequantly. In order to solve this problem, we analysed the spatial correlation between clusters, and design an in-network algorithm(ISCDG) to provide a better clustering result. A simple mechanism to do cluster maintenance is also provided in our algorithm.

Finally, we evaluate our work with different size of network, and show that the scalability of ISCDG is good. That means ISCDG can used on large scale wireless sensor network.

誌謝

首先要感謝的是我的指導教授黃俊龍老師, 在就讀碩士的這段時間, 老師給了我許多論文閱讀. 報告以及思考方式的建議, 開始進行實驗和論文撰寫之後, 也經常透過和老師的互相討論得到許多新的啓發, 能夠順利的完成這篇論文, 我非常感謝老師給我的幫助和指導。已經畢業的壬禾學長. 國禾學長. 建平學長和欣怡學姊在我碩一時給我很多的幫助, 特別感謝壬禾學長在研究和學業上給我的幫助, 以及欣怡學姊常常跟我聊天, 爲我的研究所生涯增色不少。也感謝小一屆的信翰. 濬仲. 建益和金亮, 和大家相處非常愉快, 其中常常麻煩信翰幫忙, 很感謝他總是熱心的伸出援手。博士班學長裡, 感謝士銓學長常常在聊天中給我很多想法, 最重要的就是同屆的文添. 育任及建彬, 很高興能夠跟大家一起度過兩年的實驗室生活。

感謝我的父母親在生活上和精神上給我的支援, 讓我可以順利完成學業, 這段時間沒有常常陪伴在你們身邊, 感謝你們的體諒。也感謝在新竹時的室友三福和正傑, 有你們在日常生活中互相扶持互相鼓勵, 是非常幸運的事。另外要感謝我的女友惠娟和貓咪錢錢, 謝謝你們給我很多精神上的安慰, 紓解我的煩躁和忍受我的無聊。最後, 非常感謝大門口的交大土地公, 總是帶給我心靈上的平靜並且保佑我能夠順利的畢業。感謝每一個在我就讀研究所期間幫助過我的人, 無法一一列舉, 祝福各位平安順利。

目錄

List of Tables

List of Figures

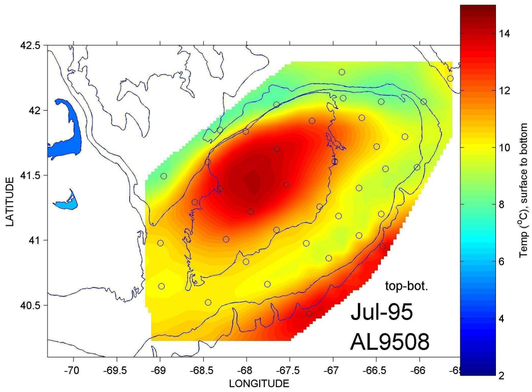


Figure 1: GLOBEC AL9508 - mean temperature

1 Introduction

For the reason of getting enough information, some applications of wireless sensor network (WSN) must periodically report data to sink– Environmental monitoring and disaster detection, for example. But the routine work is a heavy overhead for WSN since frequently transmitting reporting packet will shorten lifetime of battery-powered sensor. In order to solve this problem, query processing such as Directed Diffusion[3] and TAG[6] is a common way to reduce number of communications in WSN. Directed Diffusion selected better path to saving energy based on data-centric routing, while an in-network aggregation scheme was designed for TAG under tree-based routing structure. Besides, the spatial correlation between sensor nodes has been used on some previous research studies to provide an efficient data routing path, or to report a precision constrained data which sent by a represented node. Spatial correlation is a natural characteristic of wireless sensor network. If two sensor nodes are close to each other, the spatial correlation between them is also stronger. Based on this principle, Author in [8] analyzed different types of routing scheme and design a location-based clustering algorithm to report data more efficiently. Location of sensor on network topology has been considered to provide an efficient path for reporting data in the location-based algorithm, but the total number of reporting packets on entire network can't be significantly reduce specially on applications which report data periodically. In fact, each two sensors which positions are close, there exists a bigger probability to sense similar reading by them, as shown in Figure 1.

Figure 1 is image of GLOBEC AL9508[1] project, which show up the mean temperature of a part of ocean on July, 1995. We can notice that temperatures are similar in a close area, or we can say that the sensing result will be similar if two sensing devices have close location on topology. According to the spatial correlation of sensing readings, some previous research studies cluster related sensor nodes, then choose a represented node to be in charge of reporting sensing result. Instead of returning the real data to sink, those schemes return an approximate result to sink since in applications such as environmental monitoring, a general image is good enough. To make sure the return data are tolerable by user, a user specified threshold should be set first, then algorithm check whether two sensors can be cluster together or not based on their locations and sensing readings.

Algorithms such as CAG[12] and Snapshot[4] are clustering schemes based on spatial correlation of sensor reading. CAG start a clustering process from sink and cluster sensor nodes with a tree structure. But the clustering result of CAG is not good enough, since each CH is boundary node of cluster cause the reading of CH may not be representative. Another problem is that a skew CH reading probably makes one essential cluster to become separate on final result. Snapshot use linear regression to model correlation between sensors. Unlike CAG, Snapshot is a fully distributed clustering algorithm, and focus on the correlation among neighboring nodes. The amount of communications between sensors within clustering process and clustering maintenance may be a heavy overhead for WSN, and one hop communication also limit cluster size. DCglobal[2] is a centralized clustering algorithm. Powerful base station collect information of entire WSN, then cluster sensors based on partial order relation which defined by authors. Integral information can be considered on DCglobal, but a centralized scheme will cause clustering messages must though entire network, it may be a heavy overhead for a large WSN. According to these research studies, consider spatial correlation of sensor reading can help user to obtain approximate result and prolong network lifetime, but still some problems when clustering sensors. We list directions that have to be improve as follows:

- Avoiding produce too much control messages in the process of clustering sensors : Those messages which only for clustering are called control messages. Control messages can help sensors to exchange information for clustering, but they are not useful for the result of application, and too much control messages also consume energy.

- Providing clustering result as good as possible based on their spatial correlation : The less number of clusters means a better clustering result, since that means reporting messages of each round will also be less. How to improve clustering result based on spatial correlation between sensors is important.

- Providing a easy way for cluster maintenance : As time goes on, sensor reading will change. Once a sensor is not belong to its own cluster anymore, a cluster maintenance process must start. If cluster maintenance is too complicated, the correctness may be affect and the number of control messages is also larger than a easy way.

In order to achieve these goals, we design a in-network spatial-correlated data gathering algorithm(ISCDG) based on CAG, since CAG can spend fewer control messages to provide a basic clustering result. The way we merge two spatial-correlated clusters was called regroup, which will be introduced in Section 3.2. We list our contributions as follows:

- Our in-network algorithm cost less energy than centralized method, but a good clustering result can be obtained. We also provide a low delayed time version to actively regroup spatial-correlated clusters, it makes that user can keep gathering spatial-correlated data efficiently

- The scalability of our proposed algorithm is good, that means our in-network algorithm is suitable for a large scale wireless sensor networks

- A mechanism to handle low energy node is provided, it can effectively extend the network lifetime of wireless sensor networks

We introduce some related works and briefly describe some methods in Section 2. In Section 3, we particularly describe our algorithm. The important idea of regroup is introduced in Section 3.2. The total process of ISCDG is in Section 3.3. First, we briefly describe how CAG providing an initial clustering result and its advantages for our algorithm in Section 3.3.1. In the Section 3.3.2, we describe how to re-electing a new CH. Phase 1 of ISCDG is described in Section 3.3.3, and the phase 2 is in Section 3.3.4. The cluster maintenance method is in Section 3.4. Mechanism to handle low energy node is described in Section 3.5. Finally, we evaluate our algorithm with existed researches in Section 4.

2 Related Works

In this Section, we survey some research studies based on spatial correlation of sensor in WSN. Spatial correlation of sensor has been considered from two kinds of aspect.

2.1 Spatial Correlation of Sensor Location

First, spatial correlation of sensor location was used to improve routing[8] and link quality[11]. In Patterm's research[8], a location-based clustering algorithm has been published after analyzing different routing scheme such as Distributed Source Coding (DSC), Routing Driven Compression (RDC) and Compression Driven Routing (CDR). Sensor node can route data efficiently through cluster and improve data reporting performance in WSN. Because of the irregularity in quality of wireless communication between sensors, Xu.[11] published a weighted regression algorithm to efficient and accurate estimation of link quality in WSN. A sensor communicates with its neighbors and captures the spatial correlation between them, then estimate quality of links to its neighbors. These information can help sensor node to make a better routing decision and improve the performance of WSN further. Another research such as [9] also cluster sensors according to their locations, the different is that Qian[9] design a pre-selected scheme to cluster sensors at each round. It also provide a structure to report data only by represented node, but it's not suitable for applications which report data periodically. Spatial correlation of sensor location is used by [7] to improve coverage in WSN.

2.2 Spatial Correlation of Sensor Reading

Consider the spatial correlation of sensor reading is another aspect of spatial correlation of sensors. Most of research studies which focus on it cluster sensors based on their sensing readings, therefore exchange information between sensors must more frequently than based on sensor locations, since the sensing readings will change as time goes on. In order to provide enough information for clustering, distributed, centralized and tree-structured scheme has been published in previous researches. In distributed scheme (e.g. Snapshot[4], every sensor communicate with its neighbors directly. However, in centralized scheme (e.g. DCglobal[2], each sensor send their information to sink or base station, the powerful base station will be in charge of clustering sensors according to their returning information. A tree-structure scheme (e.g. CAG[12] and DEDAC[13] start to cluster sensors from sink, and the reading of boundary node will be the standard when clustering sensors in a small region. DEDAC improve CAG on cluster head selection, not only reading but also energy has been consider when choosing a new cluster head to represent entire cluster. All of these approaches are lossy algorithm, they focus on a representative values rather than a large number of redundant data. We briefly sketch one of each category and illustrate their process particularly.

2.2.1 Snapshot

Snapshot defined a model to calculate spatial correlation between two sensors. Multiple pairs of values for measurements $x_i(t)$ and $x_j(t)$ have to be cached, the format of them as follows:

$$(x_i(t_1), x_j(t_1)), ...(x_i(t_n), x_j(t_n))$$

For sensor i, the values of $x_j(t)$ was model as a linear projection of $x_i(t)$. That means sensor i will calculate a approximation result $\hat{x}_j(t) = a_{i,j} \times x_i(t) + b_{i,j}$. Snapshot used sum-squared error metric to measure approximation result, and a threshold T was defined to judge if sensor j can be represent by sensor i or not. Each sensor record the number of represented node by itself and their id. A example which display the representative relation between sensor nodes as show in Figure 2(a), Snapshot then start a elective process based on these information, the final result of Snapshot in our example is shown in Figure 2(b). If a node can represent most sensors, it should be choose to be CH first

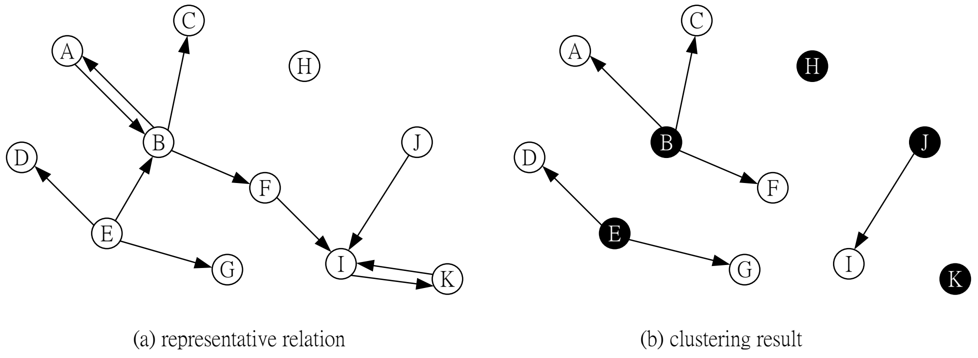


Figure 2: Example of Snapshot

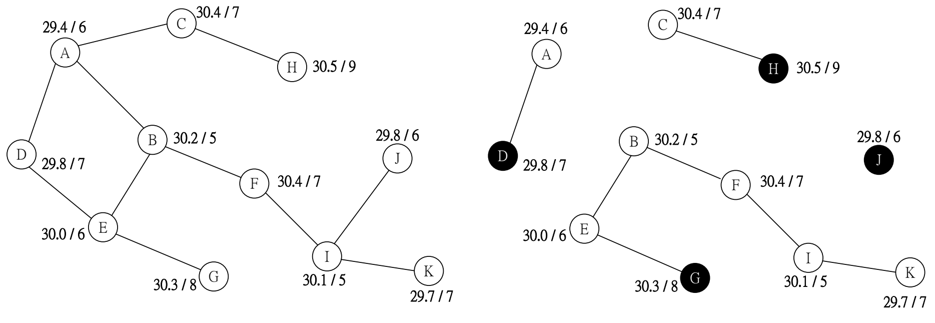
to minimize the number of CH.

Snapshot provide a model to measure the spatial correlation, multiple pairs of values definitely help sensor to check the strength of spatial correlation with its neighbors. In order to manage the cached values, a cached management also be discussed in Snapshot. The biggest problem in Snapshot is that too much control messages in the process of clustering and cluster maintenance. The other question is communication range in Snapshot is bound for on one-hop, that also means quality of clustering result must limit by same reason.

2.2.2 DCglobal

DCglobal perform clustering process as the sink. Each sensor nodes should first report their readings and energy levels to the sink, and sink will obtain global information of entire WSN after reporting. Since CH have to report their readings, choosing a node with lower energy to be CH will quickly drain out its energy. In order to avoid this situation, DCglobal define a partial order relation to order sensor nodes according to their energy levels first, secondly is their data coverage range. The data coverage range of sensor node s_i, denoted as C_i is a set of sensor nodes. For each sensor s_j in C_i there exists a sequence of sensor nodes $< s_i = s_0, s_1, ..., s_k = s_j >$ for $k \geq 0$, s_{t-1} directly communicates with s_t and the difference between s_i and s_t must less or equal than user-specified threshold for $1 \leq t \leq k$. Following this rules, DCglobal can create a ordered list which indicates the priority of selecting CH. Figure 3(a) illustrate the sensing reading energy level of each sensors, if there exists a link between two sensors means they can communicate directly. Figure 3(b) show the final result of DCglobal.

DCglobal use the global information of entire WSN to provide a better clustering



(a) reading / energy level

(b) final result

Figure 3: Example of DCglobal with threshold = 0.8

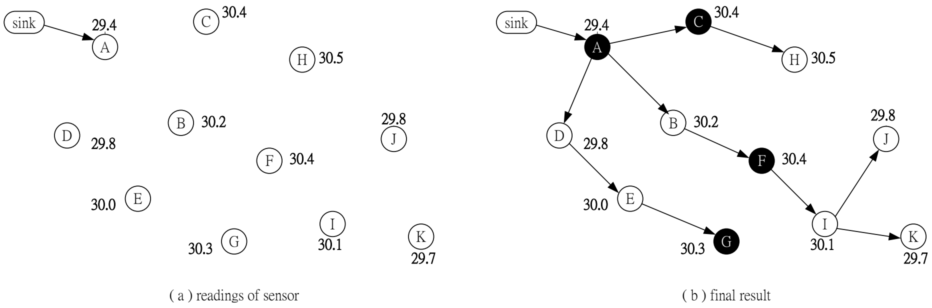
result and extend network lifetime. Since DCglobal is a centralized approach, the number of control messages should increase quickly in a large WSN. The other problem is that if the readings of sensors change frequently, control messages of cluster maintenance also increase work loads of WSN.

2.2.3 CAG

Yoon et. considered different applications, and published two modes for CAG in accordance with the process of applications. In interactive mode, the CAG algorithm operates in two phase: query and response. During the query phase, sink start to send query message which include CH sensor reading (CR) and user-specified threshold τ. Each sensors that receive query message have to check if my local sensor reading (MR) can be cover by CR under the threshold or not, that means each node should check if

$$MR < CR \pm CR \times \tau$$

or not. Sensor nodes which satisfy this condition are members of cluster that CR represented, the other ones will become new CHs and use their MRs to be CRs in the following query messages. After the query phase, a routing tree must be constructed and the clustering process also be done. Only CH have to report data at the response phase. For those applications which seldom query WSN to answer sensing readings, CAG with interactive mode can provide a reporting tree and a clustering result based on instant readings. Streaming mode has been designed for applications that need to periodically report data. After query phase, CHs start to answer sensing data, and the cluster adjustment scheme were in charge of adjusting cluster members during the reporting process. Figure 4 is an example of CAG, Figure 4(a) illustrate readings of sensor, and 4(b) show the clustering result of CAG.



(a) readings of sensor

(b) final result

Figure 4: Example of CAG with threshold = 0.8

The result of clustering is limited since the approach that selecting CH can not provide a representative node to be CH. In fact, Routing tree constructed and cluster sensors were combined on CAG can provide a further chance to improve the clustering result, two clusters close to each other also have a bigger chance to further cluster. We will use this characteristic to design an algorithm to fix the drawbacks of CAG.

3 In-Network Spatial-Correlated Data Gathering Algorithm(ISCDG)

3.1 Overview

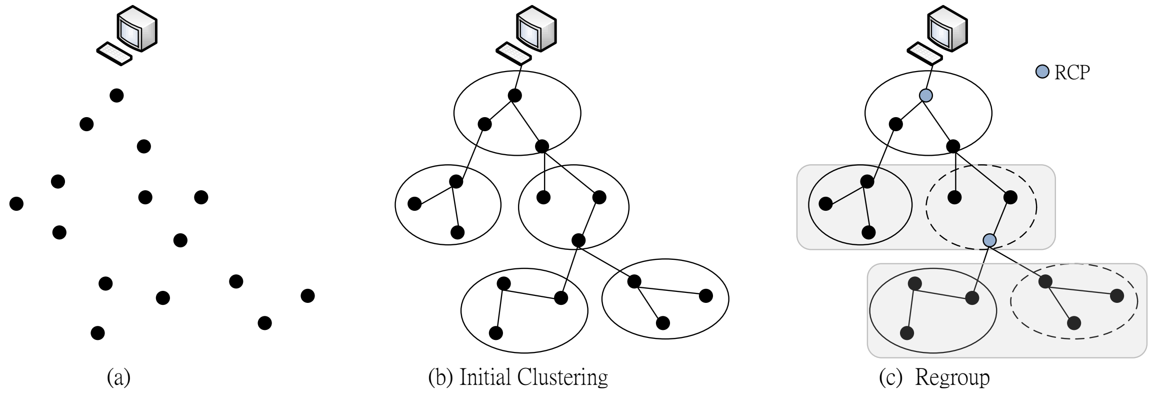
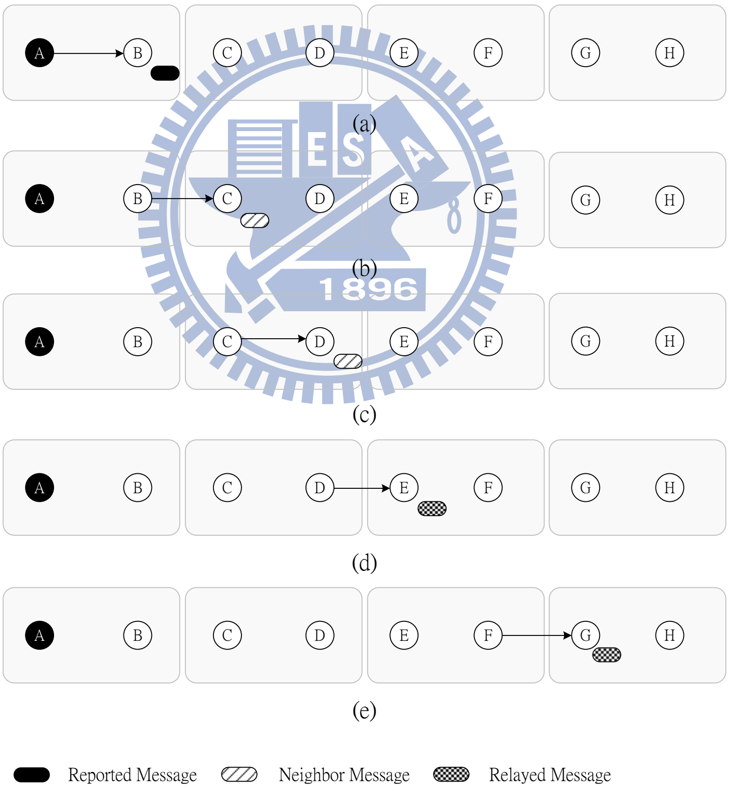


Figure 5: System Overview

Figure 5 illustrates the procedure of our in-network algorithm, ISCDG. Number of sensors are deployed in a sensing field to monitor environment. An interactive query system is required in our scheme, and user must accept approximate answers which reported from sensor nodes. At first a query with one user-tolerable threshold broadcasted from sink to whole WSN, and sensor nodes will group into several clusters based on the threshold. User specify the query as $< F, \varepsilon >$, where F is the feature of sensing region that user required, such as temperature, humidity, or luminosity. This step is called "Initial Clustering", a lot of clustering method based on spatial correlation of sensing reading can be used in the step. Since CAG provides some benefits for merging algorithm, we use CAG to cluster sensors in the "Initial Clustering" phase. An initial clustering result as shown in Figure 5(b).

We name the procedure of merging spatial-correlated clusters "Regroup". After initial clustering, some close clusters still have potential to merge into one under threshold constrained. We call these clusters spatial-correlated clusters. In order to in-network check spatial-correlation between two clusters, a special tag was added to message reported by cluster head at each reporting round. This tag will change state when message received by nodes that belong to different clusters. Figure 6 is an example to illustrate the classification of messages. Node A is the head of the cluster which contains A and B. When A starts to report its sensing data at reporting time, it sets this message tag to *Reported*

Message, and sends it to its parent on the routing tree. Since node B belongs to same cluster with A, so it keeps the tag of message that reported by A as shown in Figure 6(a). While node C received the message from B, it noticed that this message comes from different cluster and its tag is *Reported Message*, and changed tag to *Neighbor Message* then keep route message to sink as shown in Figure 6(b) and 6(c). Node D kept the tag since it was from the same cluster, but while node E received this message from D, it changed the tag to *Relayed Message* as shown in Figure 6(d) since this message came from different cluster and the tag is *Neighbor Message*. Following routing nodes will never change the tag anymore since this message had already crossed many clusters, they will keep tag to be *Relayed Message*. Node G in Figure 6(e) received a message which is from different cluster, but tag of this message is *Relayed Message*, therefore G did not change tag of this message.



(a)

(b)

(c)

(d)

(e)

Figure 6: Classification of Messages

Based on the tags of messages, nodes can know that whether two messages which reported by different clusters heads have potential to be regrouped into one. If their tags are *Reported Message* or *Neighbor Message* , it means that they came from close clusters and have a larger chance to be regroup. We only care about clusters which have close positions since even two distant clusters have similar sensing data, their relation is weak due to the different environment factors such as landform. We particularly describe the condition of regroup in Section 3.2.1 and which nodes should check condition of regroup on the routing tree in Section 3.2.2.

We separate our algorithm into four steps. The second step after initial clustering, nodes re-elect a new cluster head intra each cluster. This step can choose the most representative node to be the cluster head, that means its reading is the one that close to average reading most. The other goal of re-electing CH is to obtain necessary information for our in-network merging algorithm. Unlike centralized or distributed clustering algorithm, we let CHs start to report their sensing data after re-electing new CH so that we can in-network merging spatial-correlated clusters. There are two phases in our in-network spatial-correlated data gathering algorithm(ISCDG) The goal of first phase is to obtain best regroup result, and the second phase is to provide a low delay way to regroup clusters at each reporting round.

3.2 Regroup

In this Section, we introduce a characteristic of spatial-correlated clusters. In order to further reduce the number of clusters, spending some control messages can help WSN to analyse the spatial correlation between clusters and merge spatial-correlated clusters. We called this process ”Regroup”. Suppose that user-specified threshold = ϵ, for any cluster C_i, the following information are required for regrouping :

1. The current CH reading of C_i, represented as V_i

2. The absolute value of maximum difference between CH and its members in C_i, represented as $|MaxDif_i|$

3.2.1 Condition of Regroup

In our in-network scheme, we check the condition of regroup on the way that CH report its sensing data to sink. The routing tree constructed by CAG will be used in our in-network scheme. CAG provide a simple way to construct a routing tree based on spatial correlation, closed clusters have a great chance to report their data through a same node. Those nodes in charge of routing reporting data of closed clusters are called ”Regroup

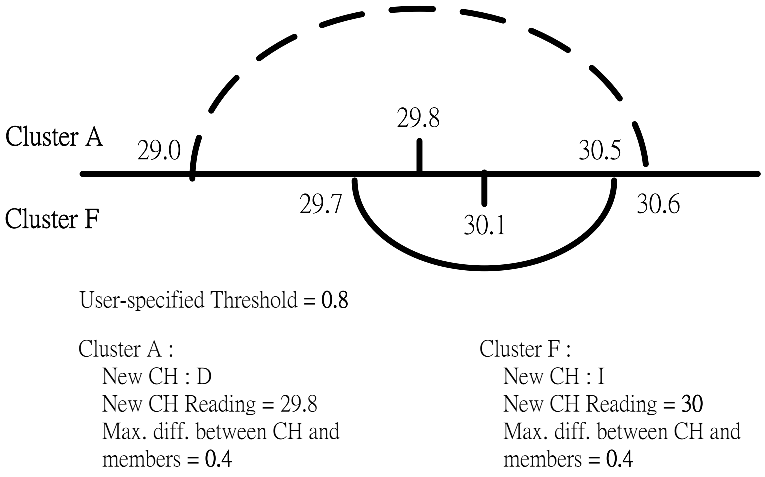


Figure 7: Example of condition of regroup

Checking Point (RCP)". We will give a detailed definition of RCP in the next Section. The following conditions must be checked by RCP locally.

Definition 1. Given clusters C_i and C_j, C_i can represent C_j in in-network approach if

1. The messages they reported will pass through a same regroup checking point
2. Both of their messages mark are not **relayed message** when two messages meet at a regroup checking point
3. $V_j \pm |MaxDif_i| \subseteq V_j \pm \epsilon$

The basic idea in our in-network approach is that no base station such as sink owns the global information of entire WSN. Thus regrouping clusters must be done locally. Figure 7 is a simple example of condition of regroup. Followed the clustering result in Figure 4(b), cluster A means the cluster whose original CH is sensor A, and cluster F means the cluster whose original CH is sensor F. These two clusters have a chance to be regrouped due to their highly spatial correlation. After re-select a sensor that most close to average reading of cluster to be new CH, the new CH ID and its reading are shown in Figure 7. For cluster A, its CH reading now is 29.8 and the range of value it can cover is $[29.0, 30.6]$ with $\epsilon = 0.8$. Considered cluster F, since its new CH reading is 30.1 and the $|MaxDif_i|$ is 0.4. So the range of value on cluster F is $[29.7, 30.5]$.We can easily notice that cluster F can be totally covered by cluster A, or C_A represent C_F in other words.

3.2.2 Regroup Checking Point

Some nodes in the WSN are in charge of checking the condition of regroup between clusters, and we call these nodes ”Regroup Checking Point” or RCP. Only cluster heads and those nodes having more than one child in the routing tree are RCP. Since branch node has a large chance to receive messages which reported from different clusters, and CH owns the representative data of its cluster by nature, CH can check the condition of regroup between received message and itself. In our algorithm, RCP examines tags of received messages. RCP buffers messages which are *Reported Message* or *Neighbor Message* in ISCDG - phase 1, and checks the condition of regroup between them to choose a representative cluster after a period of time. In order to reduce the delay time and keep regrouping spatial-correlated clusters at each reporting round, RCP routes received messages to its parent after recording their information. We particularly describe the scheme in Section 3.3.4.

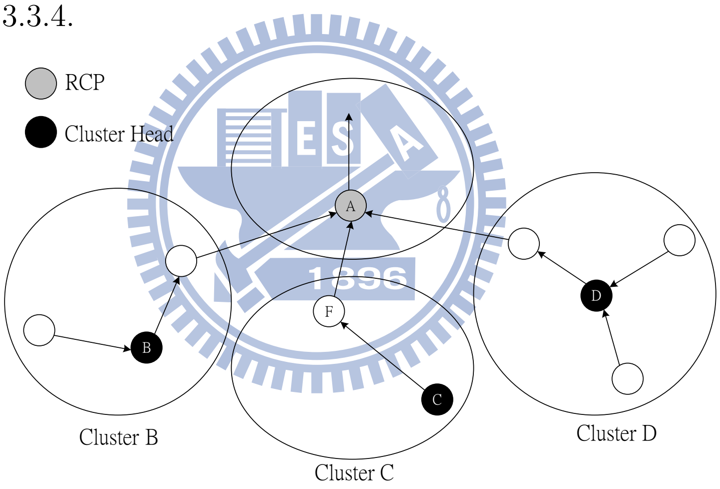


Figure 8: Example of RCP

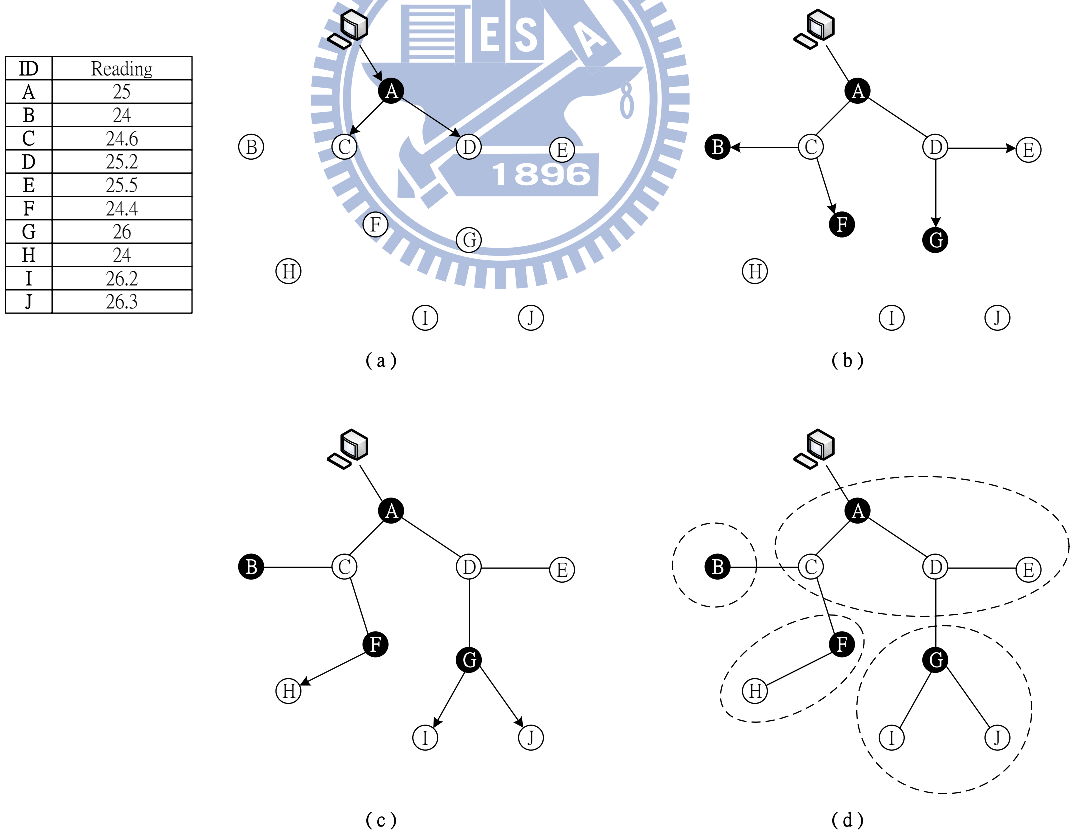
Figure 8 illustrate an example of RCP. Sensor nodes B, C, and D are cluster heads, so they are also RCP. The procedure of regroup will be execute once they receive a message from close clusters. Node A is not a CH, but also a RCP since it has three children in routing tree. The messages which reported by B, C, and D will pass through node A on their reporting way, so A is the RCP that be in charge of regrouping them.

3.3 Algorithm

We described our in-network spatial-correlated data gathering algorithm in this Section. Particular process of each step of ISCDG will be described in order. First, we briefly illustrate how CAG providing an initial clustering result. Second, we re-electing a new CH within each cluster. We then describe our two phases ISCDG in the following two subsection.

3.3.1 Initial Clustering

Although our in-network data gathering algorithm can be used on clusters which cluster based on spatial correlation, we chose CAG to provide an initial clustering result for further regroup. CAG is an uncomplicated clustering algorithm, clustering all nodes can be done after every nodes broadcast query once, that means the clustering cost is equal to number of nodes in WSN. The other benefit is that CAG also build a routing tree at the same time.



ID	Reading
A	25
B	24
C	24.6
D	25.2
E	25.5
F	24.4
G	26
H	24
I	26.2
J	26.3

Figure 9: Example of CAG with Threshold = 0.5

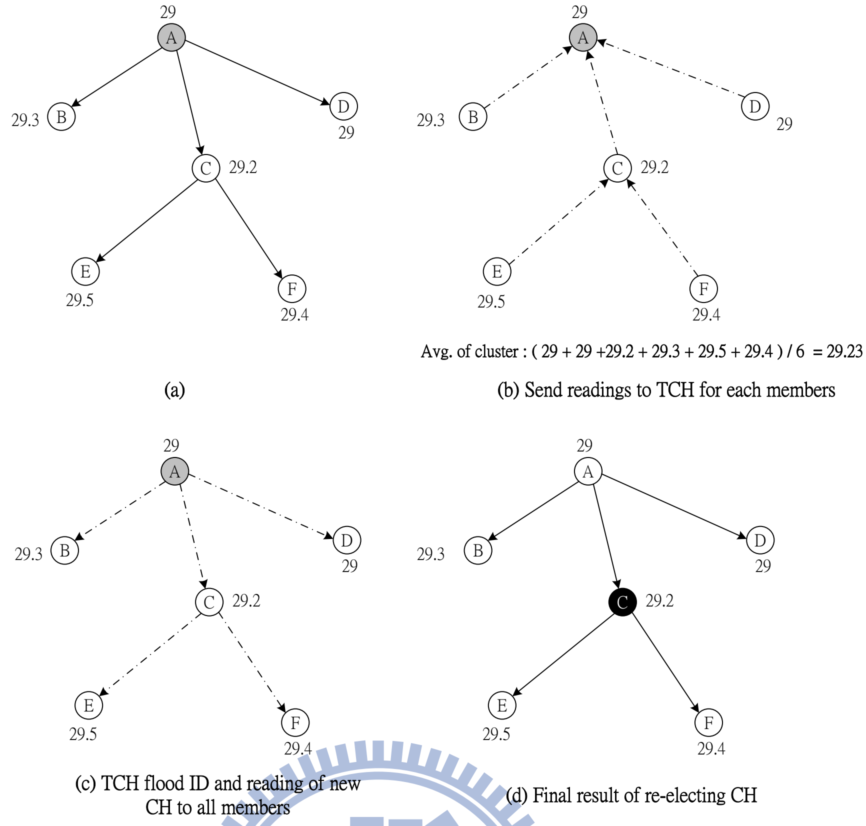
Figure 9 is an example of CAG. Sensor node A received query from sink, and started to cluster nodes. A became a CH and broadcast its reading to neighbors as shown in Figure 9(a). While node C and D received query from A, they tested that whether their reading lie in [25-0.5 , 25+0.5]. Since 24.6 and 25.2 both lain in the range, so C and D joined cluster of A, then kept broadcasting with reading of A. Node B, F, and G exceed [25-0.5 , 25+0.5], so they became a new CH. When these new CHs broadcast query, they use their own reading to be cluster base. Figure 9(d) show the final result of CAG in our example, and A, B, C, F, D, and G are RCPs as our definition.

3.3.2 Re-Elect Cluster Head

Observing the cluster which contains G, I, and J in Figure 9(d), their average reading is 26.16. According to the result, most representative node in this cluster should be node I. In order to shift CH to most representative one, and calculate $|MaxDif|$ of cluster, we design a procedure to re-elect CH. Figure 10(a) illustrate a cluster which is built after procedure of CAG, and sensor A is the CH. For the reason to differentiate from the new CH, we called sensor A is temporary CH (TCH). After the procedure of cluster sensors is done, each member starts to send their readings to their TCH as shown in Figure 10(b). TCH collects readings of cluster members, and calculates the average reading of entire cluster. In the example as shown in Figure 10, the average reading of cluster is 29.23. TCH will choose the node which reading is most close to average reading to be new CH. So sensor C is the new CH of cluster in our example. TCH floods the result to entire cluster, all the members will know the ID and reading of new CH. Figure 10(d) shows the final result of re-elect cluster head.

There are many ways to set the timing that sensors should start the procedure of re-elect cluster head. A simple method we chose is to start re-elect CH by boundary node of each cluster. Since a node will be a boundary of cluster if its child is not belong to the same cluster with it. So the scheme we designed is that if a node find out it is a boundary node, it starts to send its reading to its parent. When the parent nodes receive all readings of its children, it then sends to its own parent. Once TCH receives messages from all of its children, it can start to calculate the average reading. By this method, the procedure of re-elect cluster head can just start when a cluster is constructed, and do not have to wait for entire procedure finished.

Two advantages are provided by the process of re-elect CH. The first one is to obtain


29
A
29.3
B
D
29
C
29.2
E
29.5
F
29.4
Avg. of cluster : (29 + 29 +29.2 + 29.3 + 29.5 + 29.4) / 6 = 29.23

(a)

(b) Send readings to TCH for each members

(c) TCH flood ID and reading of new CH to all members

(d) Final result of re-electing CH

Figure 10: Example of Re-Elect Cluster Head

	CH reading	Sum of absolute value of difference with CH	Maximum difference with CH($\|MaxDif_i\|$)
Before re-electing CH	29	0.3+0.2+0.4+0.5=1.4	0.5
After re-electing CH	29.2	0.2+0.2+0.2+0.1+0.3=1.2	0.3

Table 1: Comparison of Re-Elect CH

value of $|MaxDif|$ for each cluster that is necessary information in our merging algorithm. The other one is to shift the value of CH to a more representative position. We list the sum of absolute value of difference between members and CH according to Figure 10 in Table 1. After re-electing new CH, the sum is lower than before. That means new CH provide a more representative value for user. Another benefit is that shifting the CH reading to a middle position can usually reduce the maximum difference between members and CH. In our example, the max. difference has became $|29.2 - 29.5| = 0.3$ from $|29 - 29.5| = 0.5$. It will improve the chance of further regroup spatial-correlated clusters.

3.3.3 ISCDG - Phase 1

Consider an illustrative example in Figure 11, Sensor A is a RCP which has three children, and Sensor B, C, D, and E are CH of their own cluster. RCP will start a timer when

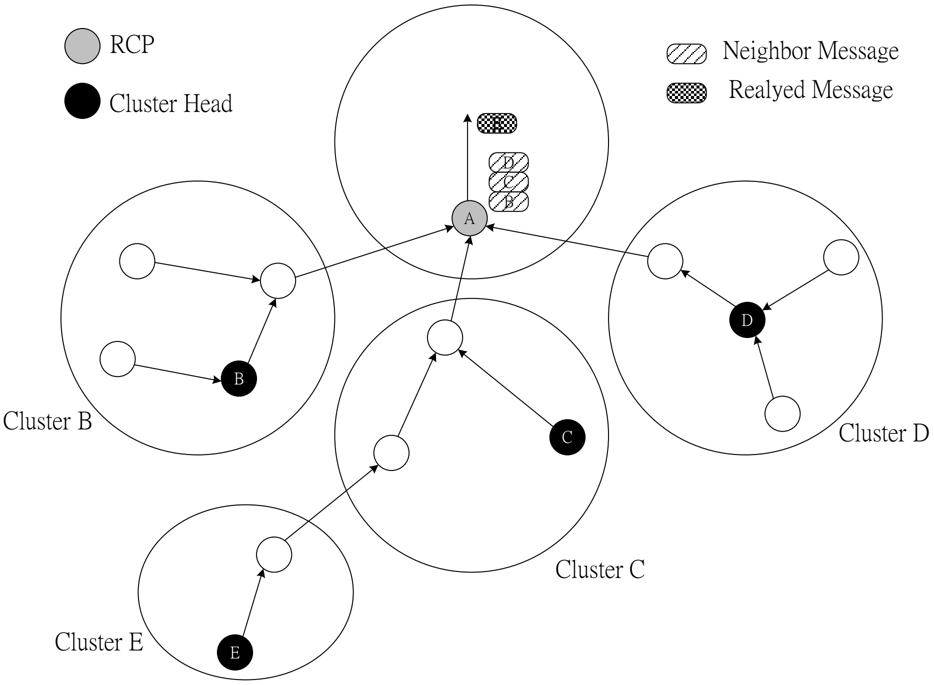


Figure 11: Example of ISCDG - phase 1

it receives first Report Message or Neighbor Message to wait for a period of time T. In Figure 11, RCP A started timer when it received Neighbor Message from B. After a period of time T expired, RCP A buffered three Neighbor Messages, and the Relayed Message which is from cluster E was kept routing by A since what we considered is only those spatial-correlated clusters, so RCP A only check the condition of regroup with Neighbor Messages which are from cluster B, C, and D.

	CH Reading	$\lvert MaxDif\rvert$	User-tolerable Threshold
Cluster B	29	0.25	0.5
Cluster C	29.2	0.3	0.5
Cluster D	29.35	0.2	0.5

Table 2: Information of Neighbor Messages on RCP A

We list information contained on Neighbor Messages which received by RCP A on Table 2. Except CH reading, each message must include the maximum difference between members and CH, defined as $|MaxDif|$ in previous Section. RCP A checks the condition of regroup between each two messages. According to the definition of condition of regroup, for cluster B since $[29.2 - 0.3, 29.2 + 0.3] \subseteq [29 - 0.5, 29 + 0.5]$, cluster C can be covered by cluster B. Similarly for cluster C, $[29 - 0.25, 29 + 0.25] \subseteq [29.2 - 0.5, 29.2 + 0.5]$ and $[29.35 - 0.2, 29.35 + 0.2] \subseteq [29.2 - 0.5, 29.2 + 0.5]$, we knew that Cluster C can cover cluster B and D. Figure 12 illustrates the relation of regroup between messages, an arrow from

B to C means cluster B can cover cluster C. According to Figure 12, we greedy choose the message which has the most degree of outgoing arrows to be "Master Cluster" and those messages linked by Master Cluster are called "Collateral Cluster", so the cluster C is Master Cluster in our example, and cluster B and D are Collateral Cluster of C. The tie-breaker when two messages have the same degree of outgoing arrows are mark of messages and sensor id. Reported Message has higher priority than Neighbor Message when both of them have the same degree, if the mark of messages are also identical, bigger sensor id must be another way to break the tie.

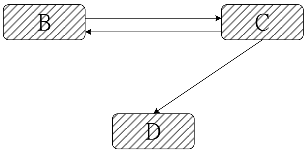


Figure 12: Relation of Regroup between Cluster B, C, and D

RCP has two jobs must to do after master clusters decided.

1. Adjust the $|MaxDif|$ of master cluster, then keep routing the message of master cluster to sink

2. Send control messages of regroup to collateral cluster include information for cluster maintenance

After we regroup spatial-correlated clusters into one, the master cluster is in charge of reporting data for its own cluster and collateral clusters it covered. To make sure the $|MaxDif|$ of master cluster can represent both of master and collateral clusters, RCP calculate the new $|MaxDif|$ of master cluster ($|MaxDif|_m$) as follows :

$|MaxDif|_o$ = Original $|MaxDif|$ of master cluster

$|MaxDif|_m$ = New $|MaxDif|$ of master cluster

MCH = Cluster Head of Master Cluster , CCH_i = Cluster Head of Collateral Cluster i

$ShiftDif_i$ = $|MCH$ reading - CCH_i reading$|$ + $|MaxDif|$ of Collateral Cluster i

$|MaxDif|_m$ = MAX($|MaxDif|_o$, $ShiftDif_i$) for all cluster i which is collateral cluster of this master cluster

The new value, $|MaxDif|_m$ will be became the $|MaxDif|$ of master cluster, and be included on the message. The $|MaxDif|_m$ of master cluster C is 0.45 in our example.

Figure 13 illustrates actions of RCP after regroup. Control messages of regroup sent by RCP to collateral clusters, cluster B and D, to notify them that they are collateral clusters of cluster C. The CH reading of their master cluster must be included on control messages, CH of collateral clusters uses this information to calculate safe region for cluster maintenance later. Since RCP regroups cluster B, C, and D, and cluster C is the master cluster, so only reported data send form cluster C will be keep route to sink, as shown in Figure 13. Collateral clusters re-starts to send their reported messages only if their CH reading exceed the safe region, otherwise they will not report data at following rounds.

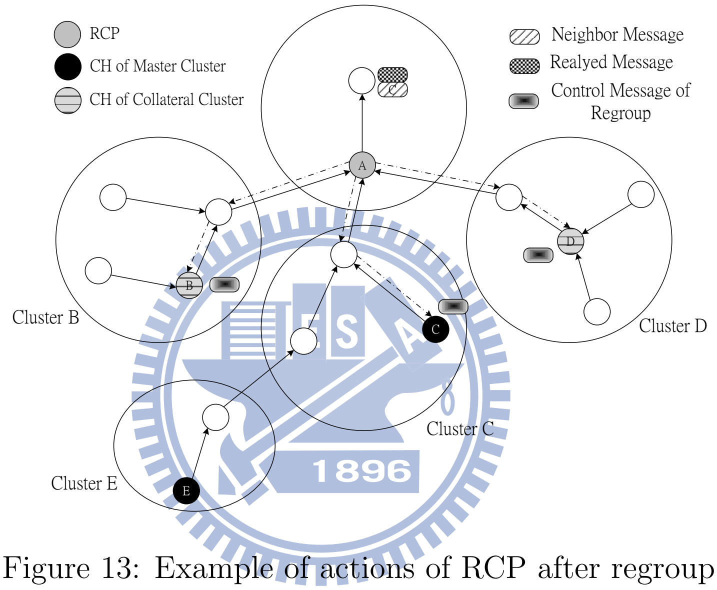


Figure 13: Example of actions of RCP after regroup

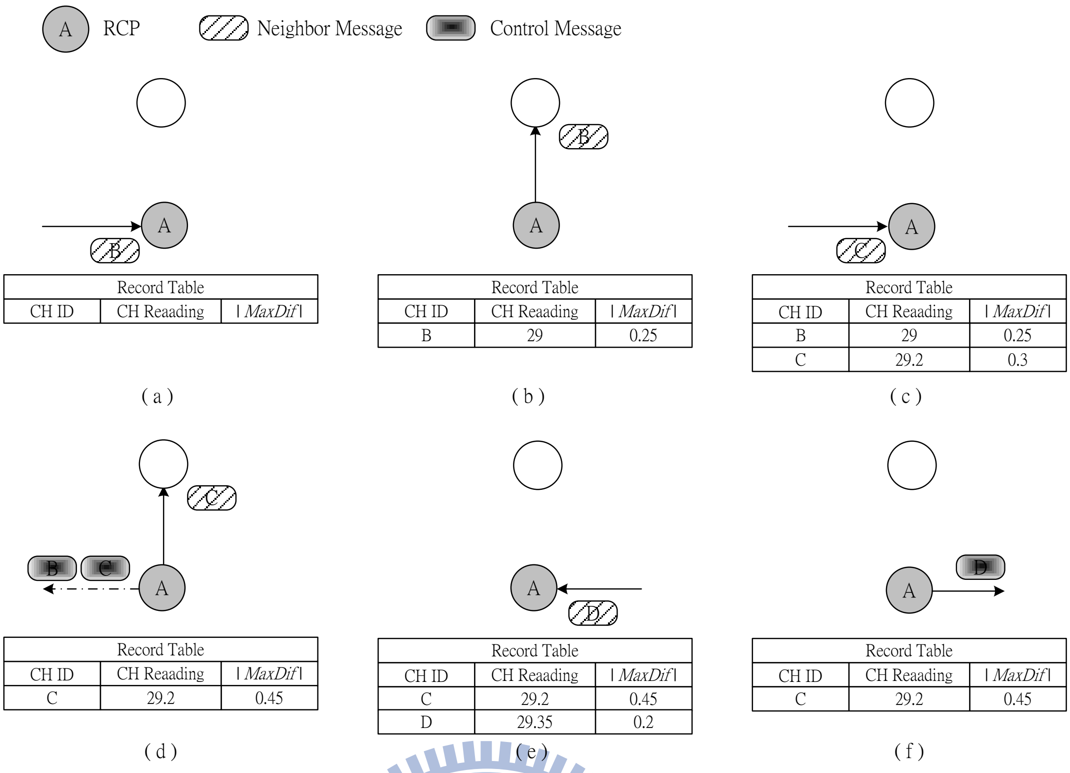
Algorithm 1 : ISCDG - Phase 1 for RCP

```
Input:
$\varepsilon$ : user-tolerable threshold
$C$ : set of all Reported Messages and Neighbor Messages received on a period of time $T$
$|MaxDif_i|$ : $|MaxDif|$ of cluster i
$v_i$ : CH reading of cluster i , $i \in C$
Output:
$R$ : set of master clusters, where $R \subseteq C$
for each cluster i, $i \in C$ do
    for each cluster j, $j \neq i$ do
        if ( $v_j \pm |MaxDif_j| \subseteq v_i \pm \varepsilon$ ) then
            cluster i can represent cluster j
            RCP add cluster j to $RepresntedList_i$
        end if
    end for
end for
while( $C \neq NULL$ ) do
    choose cluster i which has max size of $RepresentedList_i$, $i \in C$
    add i to $R$
    for each j represented by i do
        Remove j from $C$
    end for
end while
for each cluster i, $i \in R$ do
    Calculate new $|MaxDif|$ for i
    Check classification of message of i
    Keep route message of i to parent
    Notify i to adjust its $|MaxDif|$
    for each cluster j, j is represented by i do
        Notify j is a collateral cluster
    end for
end for
```

3.3.4 ISCDG - Phase 2

Follow the phase 1 of ISCDG, we can effectively regroup spatial-correlated clusters. But the buffered method which used on RCP may extend delay time of WSN. Specially on applications that report data periodically, a longer delay time will influence the time effect of reporting data, so we do not buffer spatial-correlated messages on RCP at each reporting round except the first round. In phase 2, RCP check condition of regroup between messages in their time order.

Figure 14 illustrates an example of phase 2. The network topology and routing tree is as the same as Figure 11, and the information of clusters is listed in Table 2. The Neighbor Message from CH B arrived at RCP A as Figure 14(a), since the recorded table of A is empty, RCP A recorded the information of B in its recorded table, then keep routing message from B to its parent on routing tree as shown in Figure 14(b). Neighbor Message from C arrived at RCP A later then Neighbor Message from B, RCP A recorded information of C and check the condition of regroup between C and other items in its



(a)

Record Table		
CH ID	CH Reaading	| *MaxDif* |

(b)

Record Table		
CH ID	CH Reaading	| *MaxDif* |
B	29	0.25

(c)

Record Table		
CH ID	CH Reaading	| *MaxDif* |
B	29	0.25
C	29.2	0.3

(d)

Record Table		
CH ID	CH Reaading	| *MaxDif* |
C	29.2	0.45

(e)

Record Table		
CH ID	CH Reaading	| *MaxDif* |
C	29.2	0.45
D	29.35	0.2

(f)

Record Table		
CH ID	CH Reaading	| *MaxDif* |
C	29.2	0.45

Figure 14: Example of ISCDG - Phase 2

recorded table. Because B can be regroup by C, RCP A adjusted $|MaxDif|$ of C and changed the item of recorded table, then kept routing message from C. RCP A also sent control messages to notify C for its new $|MaxDif|$, and B to be a collateral cluster as shown in Figure 14(d). Neighbor Message from D arrived RCP after B and C, RCP A follow the same process to check condition of regroup between D and items of recorded table. Since D can regroup by C, RCP A stopped routing message from D, and sent a control message to notify D to be a collateral cluster as shown in Figure 14(f). The information kept in recorded table of RCP A are information of master clusters.

In phase 2, regroup result must be affected by the order of time that spatial-correlated messages arrived at RCP. But almost no additional delay time from reporting CH to sink. In a application that must periodically report sensing data, we use phase 1 to regroup spatial-correlated clusters as more as possible, and use phase 2 to keep checking condition of regroup at following reporting round. Although regroup result of phase 2 will be affect by time order of spatial-correlated messages, it provide low delay time reporting way, and also actively regroup spatial-correlated clusters for applications that periodically report.

```
Algorithm 2 : ISCDG - Phase 2 for RCP
Input:
$\varepsilon$ : user-tolerable threshold
$A$ : new arrived non-Relayed Message
$R$ : set of all Reported Messages and Neighbor Messages on recorded table
$|MaxDif_i|$ : $|MaxDif|$ of cluster i
$v_i$ : CH reading of cluster i , $i \in R$
Output:
$m$ : a master cluster
add A to R
for each cluster i, $i \in R$ do
    if $i \neq A$ then
        if ( $v_A \pm |MaxDif_A| \subseteq v_i \pm \varepsilon$ ) then
            go to case 1
            break;
        else if ( $v_i \pm |MaxDif_i| \subseteq v_A \pm \varepsilon$ ) then
            go to case 2
            break;
        end if
    end if
end for
case 1 :
Calculate new $|MaxDif|$ for i
Notify A to be a collateral cluster
Notify i to update its $|MaxDif|$
Adjust recorded table
case 2 :
Calculate new $|MaxDif|$ for A
Keep routing information of A to parent
Notify A to update its $|MaxDif|$
Notify i to be a collateral cluster
Adjust recorded table
```

3.4 Cluster Maintenance

In the past, many in-network studies investigate how to effectively set up the routing tree and form the clusters for economizing the energy attrition. Unfortunately, seldom of these works center on maintaining the robustness of the clusters so that the sensors can further save power by avoiding re-establishing the clusters frequently. The longer a cluster lives, the more energy conserve for all the members in cluster. In this Section, we discuss how to maintain the structure of clusters as long as possible and provide a simple scheme to adjust clusters.

Since we focus our works on applications which periodic report data by CH of clusters, it’s important to make sure readings of cluster members are still covered by their CHs. While CHs report data at each round, readings of its members vary based on terrain or circumstances. It brings an inevitable problem that the changed readings may break the ambit of clusters. Once a sensor’s reading exceed coverage range of its CH, a process for cluster maintenance will be start to adjust cluster. A sensor which has to run a process of adjustment if it enters the ”orphan state”, we defined ”orphan state” as follows.

Definition 2. A sensor node enters ”orphan state” if

1. It’s not a CH and its reading can’t be covered by its CH anymore

2. It’s a CCH and this cluster can’t be covered by its master cluster

Once a node enter orphan state, a process of cluster maintenance will be executed to make sure each cluster of WSN still confirm spatial-correlation. To reduce the times of communication, each nodes will calculate their safe regions. If their new sensing readings still lie in the safe region, no additional actions are needed. Our ISCDG - phase 2 also provide a convenient way to maintain clusters. We describe the cluster maintenance scheme particularly as follows.

3.4.1 Cluster Member

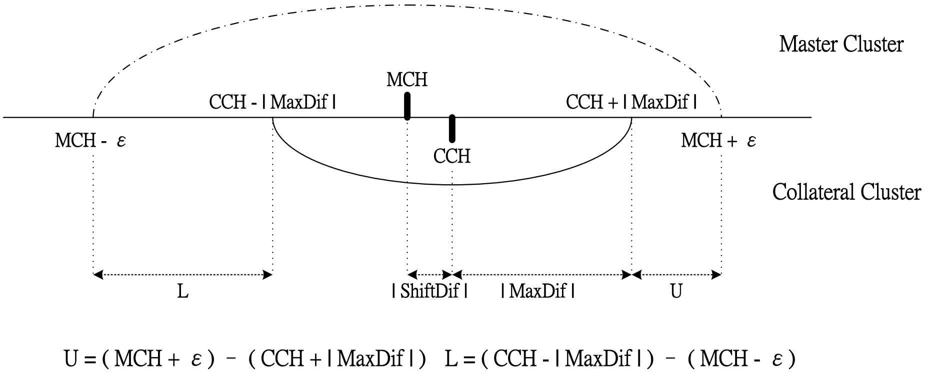


Figure 15: Safe Region of Cluster Members

We separate cluster maintenance into two parts according to their roles, and introduce ”safe region” to point out the region that sensors don’t have to execute cluster maintenance to reduce unnecessary control messages. First, for members of master cluster, since nothing changed after regroup, they just follow the user-tolerable threshold. If their readings lie in [MCH - ε , MCH + ε], they will keep staying in the same cluster, otherwise the process for cluster maintenance must be executed. For members of collateral cluster, since they have been covered by master cluster, their safe region must be affected by reading of MCH, as shown in Figure 15. The safe region of members of collateral clusters is [CCH - $|MaxDif|$ - MIN(U,L) , CCH + $|MaxDif|$ + MIN(U,L)]. If their new sensing readings still lie in the safe region means this collateral cluster still covered by its master cluster, otherwise cluster maintenance is needed.

We used a intuitional way to do cluster maintenance for members which exceed their sage region. when a member no longer belongs to its CH, it just becomes a new CH to represent itself and starts to report sensing data. Since our ISCDG - phase 2 can actively check the condition of regroup, if this new CH can be regroup by its original CH, they must meet in a RCP due to the structure of routing tree that made by CAG. Figure 16 is an example which node B exceed its safe region. Figure 16(a) illustrates the original cluster members with CH C. Since node B exceeded its safe region, it became a new CH of a one member cluster as shown in Figure 16(b). RCP A received Report Message from C and Neighbor Message from B at next reporting round. The $|MaxDif|$ of B is 0, because no other members in its cluster except itself. That means once node B still can be represented by node C, it will be regroup by RCP A as shown in Figure 16(c). Since ISCDG - Phase 2 provide a method to actively regroup nodes break away their clusters, no additional communication between members and CH is needed in our cluster maintenance method.

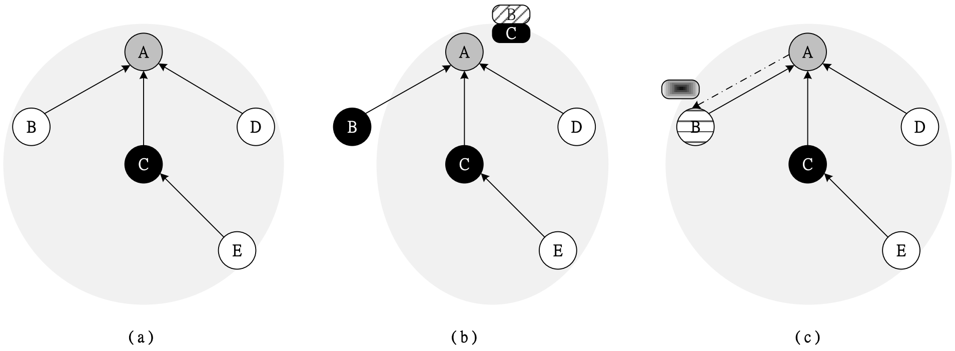


Figure 16: Example of Cluster Maintenance for Members

3.4.2 Collateral Cluster Head

To make sure that a collateral cluster is still covered by its master cluster, collateral cluster head(CCH) has to refresh $|MaxDif|$ and check the condition of regroup with master cluster to confirm the spatial correlation. Once the new reading of CCH break relation of regroup with master cluster, it will re-start to report sensing data at following rounds. Just like cluster members, if this cluster still exist spatial-correlation with previous master cluster, RCP will regroup this two clusters again.

3.5 Low Energy Node

Our goal is to prolong network lifetime of WSN, so reducing the loading of a low energy node is an important problem. A lot of researches noticed that CHs run out of energy much faster than other sensor nodes, so some replacing CH methods intra cluster were mentioned in those researches. Actually in most applications, nodes which close to sink or base station have the heaviest loading compared with entire WSN, since the more a node close to sink the bigger subtree it has, that means it has to route many reporting data from its subtree. Base on this reason, we design a mechanism to alleviate loading for a low energy node by change the structure of routing tree. We observed process of simulation, nodes which ran out their energy fastest were not absolutely a CH, but all of them close to sink on position, so our mechanism is for all nodes in WSN not only CHs.

User can define a energy threshold to estimate whether a node is a low energy node. Once energy of a node below to the energy threshold, it starts following process to reduce its loading. First it sends a message to its parent for notify that it almost runs out its energy, then the parent node will redo CAG[12] to rebuild the routing tree from itself. To prevent a low energy node still owns a big sub-tree, it delays received CAG message for a period of time T_E, that makes nearby nodes get a bigger chance to become new parent of nodes which are children in previous routing tree. A node that has higher energy will be in charge of routing reporting data to sink after this process, then follow our ISCDG - phase 1 can regroup spatial-correlated clusters again.

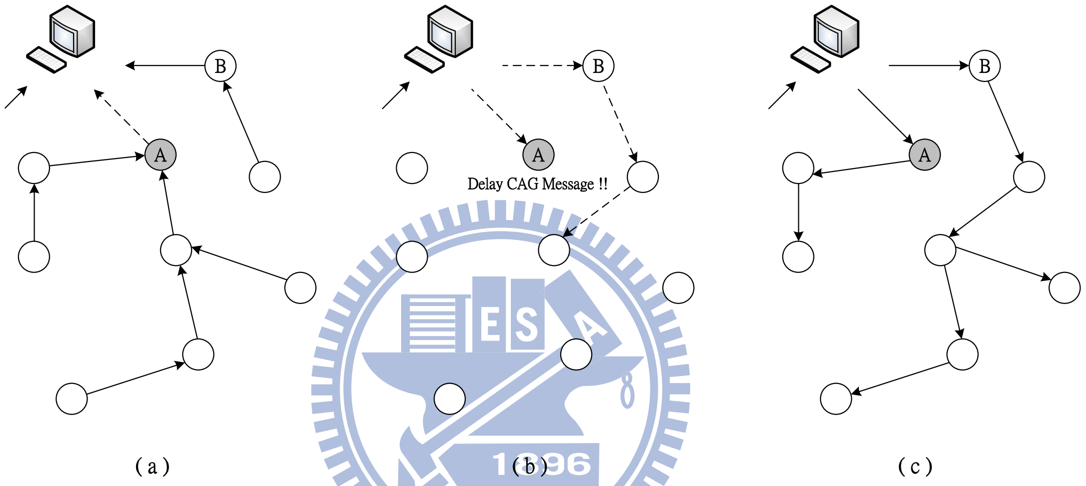


Figure 17: Example of Maintenance Mechanism for Low Energy Node

Figure 17 is an example to briefly illustrates the mechanism for low energy node. Sensor A in Figure 17(a) found out its energy below to user-specified energy threshold, so it notified its parent which is sink in this example. Sink redo CAG[12] in partial WSN, since A is a low energy node, it delayed received CAG message for a period of time T_E as shown in Figure 17(b). The nearby node B followed original mechanism of CAG to build routing and cluster nodes while node A delayed its CAG message, so node C, the original child of A changed its link to child of B since it received CAG message from it first. A new structure of routing tree is shown in Figure 17(c). Sensor A reduced its loading after our mechanism, that will prolong its lifetime effectively. The partial WSN will then use ISCDG - phase 1 to regroup spatial-correlated clusters, and use ISCDG - phase 2 to keep checking condition of regroup on RCP at following rounds. The other benefit of our mechanism is that redo CAG can get fresh result on routing tree based on new spatial correlation between sensor nodes, it can help ISCDG to get a better regroup result.

3.5.1 Node Fail

Many applications of WSN are used in a environment that full of uncertain factors, such as mine or forest. In this kind of environment, sensor nodes may lose function since some unforeknown reasons, like weather and destroyed by organism. That means it's hard to forecast the real lifetime of a sensor node in this kind of situation. In order to ensure sensor nodes still work, periodic beacon can be used to announce that it still work well. In applications which periodically report data, parents on routing tree can also track reporting data from its children to confirm that they are still alive. Once the node fail situation was detected, our mechanism for low energy node can be used to fix routing path. The difference is that only few hops execution is needed to handle the node fail situation.

4 Performance Evaluation

In this Section, we evaluate ISCDG with other existing algorithms. We use OMNeT++ 4.0 and Mobility Framework for OMNeT++ 4.0 to simulate all works. OMNeT++ is a c++ based network simulator, the Mobility Framework for OMNeT++ provide a networking stack, including application layer, network layer, MAC layer, and physical layer. We choose CSMA that is supply by Mobility Framework for MAC, Loss of messages due to collisions is taken to account. Complete details about OMNeT++ may be found at http://www.omnetpp.org, and details about Mobility Framework for OMNeT++ 4.0 is located at http://wiki.github.com/mobility-fw/mf-opp4. All experiments are simulated on real data set, the Intel Lab Data, and the synthesis data set, we made it to simulate large scale network.

We ask WSN to periodic report their sensing data, the default data reporting interval is 120 time units, i.e., CHs must report their sensing data to the sink every 120 time units. The maximum number of messages that a sensor node can transmitted is set to 10000.

Three performance metrics are used in our experimental result, the network lifetime, the number of CHs, and the clustering cost. Network Lifetime measures the length of time until first sensor node runs out its battery, it mainly affect by energy cost in procedure of clustering sensor nodes and the number of reporting messages within the WSN. Therefore, number of CHs shows the number of nodes that should report their sensing data at reporting time. Clustering cost can be used to measure the total energy cost in clustering sensor nodes. The clustering cost of ISCDG indicate total number of control messages are used in Initial Clustering, Re-Elect CH, and ISCDG - phase 1.

4.1 Real Data Set

We use the public available Intel Lab Data[5] that consists 54 sensor nodes. Since the dataset comprises readings of different feature and multiple days, we use temperature on 2004-02-28 to simulate all works. However, a little part of readings are lost, we fix them by using the reading that most close to the losing time to fill losing epoch.

4.1.1 Impact of Threshold

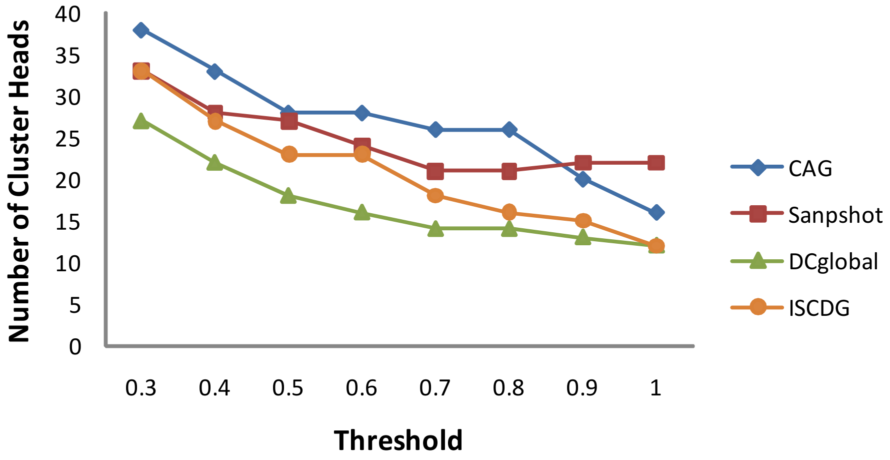


Figure 18: Number of CHs

Figure 18 shows the clustering result of each works. DCglobal collects global information, so it can obtain the best clustering result, that means least number of nodes have to report after DCglobal has been done. ISCDG get a better result while the user-tolerable threshold grows up, it almost as good as DCglobal while the threshold is 1. The reason is it will be easier to pass the condition of regroup between two clusters when the threshold become bigger. Snapshot is a one-hop clustering algorithm, so it can not get more benefit when the threshold is large enough. As shown in Figure 18, when threshold is larger than 0.8, the number of CHs of Snapshot stop decreasing. ISCDG chooses CAG to get a initial clustering result, but ISCDG still gets a better result than CAG even threshold grows up. According to Figure 18, regroup can help WSN obtain a better clustering result based on the spatial correlation.

Figure 19 shows the initial clustering cost, that is number of messages used in clustering sensor nodes by entire WSN. CAG is least one since only its initial clustering cost is equal to number of nodes, that means only 54 nodes is needed in CAG. In order to collect the global information, DCglobal spend most energy clustering sensor nodes. Our in-network algorithm spend less energy than DCglobal since we do not ask all nodes send their information back to sink, but the clustering result is similar if user accept a larger error percent. Temperature readings of Intel Lab Data on 2004-04-28 are around 19 to 25, so the error percent is almost 45initial clustering cost od ISCDG increase while threshold

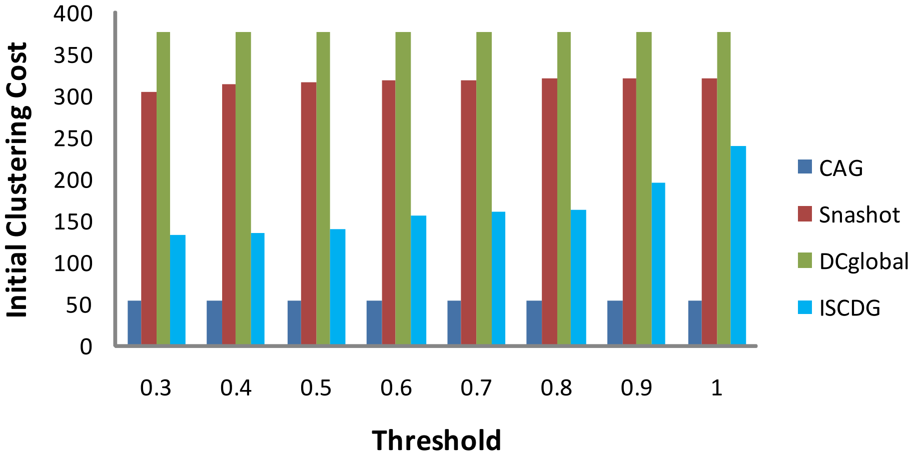


Figure 19: Initial Clustering Cost

increasing, the reason is regroup is easier to happen with a larger threshold, then more control messages must be used within entire WSN.

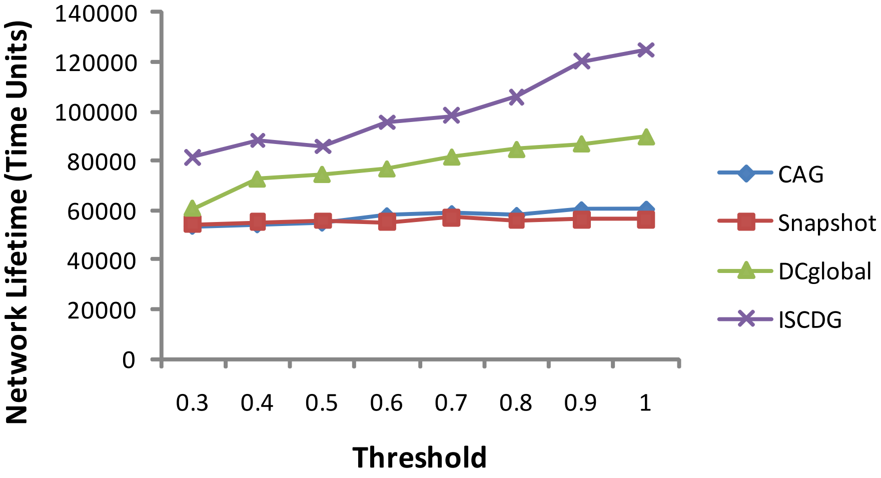


Figure 20: Network Lifetime

Figure 20 shows the network lifetime with different threshold. ISCDG has the longest network lifetime compared with other works. As shown in Figure 21, ISCDG - phase 2 effectively regroup clusters at each reporting rounds. Since ISCDG - phase 2 provide a scheme to naturally regroup CH which exceed its safe region, only light overload are needed during reporting process as shown in Figure 22(a). DCglobal also can adjust structure between clusters since its cluster maintenance is executed in sink, nodes that need to do cluster maintenance must report their current reading back to sink, it will spend many messages on cluster maintenance. The other problem of DCglobal is that messages of cluster maintenance may be loss on the road such that the work has be done again and again if node can not received maintenance message from sink successfully. Note that in applications which periodically report sensing data, too many control messages also make collision heavier, so the network lifetime of DCglobal is affect as shown in Figure 20. The maintenance mechanism of CAG is done locally, a local CAG will be triggered by node which need maintenance, so the number of control messages is low as shown in Figure 22(a). Figure 22(b) focuses the number of control messages on CAG and ISCDG, it shows that CAG even cost less energy on cluster maintenance than ISCDG. The challenge of CAG is that the clustering result is not good enough, that makes number of reporting messages keep high at each reporting round. Snapshot let every cluster members send test messages to ask their own CH for new reading once their sensing data changed, it makes a large number of control messages as shown in Figure 22(a). The other disadvantage is that it can not actively merge spatial-correlated cluster such that number of CHs keep increasing as time goes on. Because of such reasons, Snapshot has the lowest network lifetime.

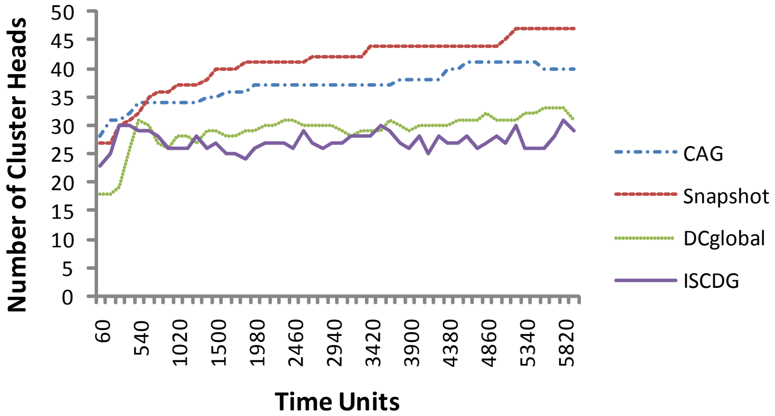
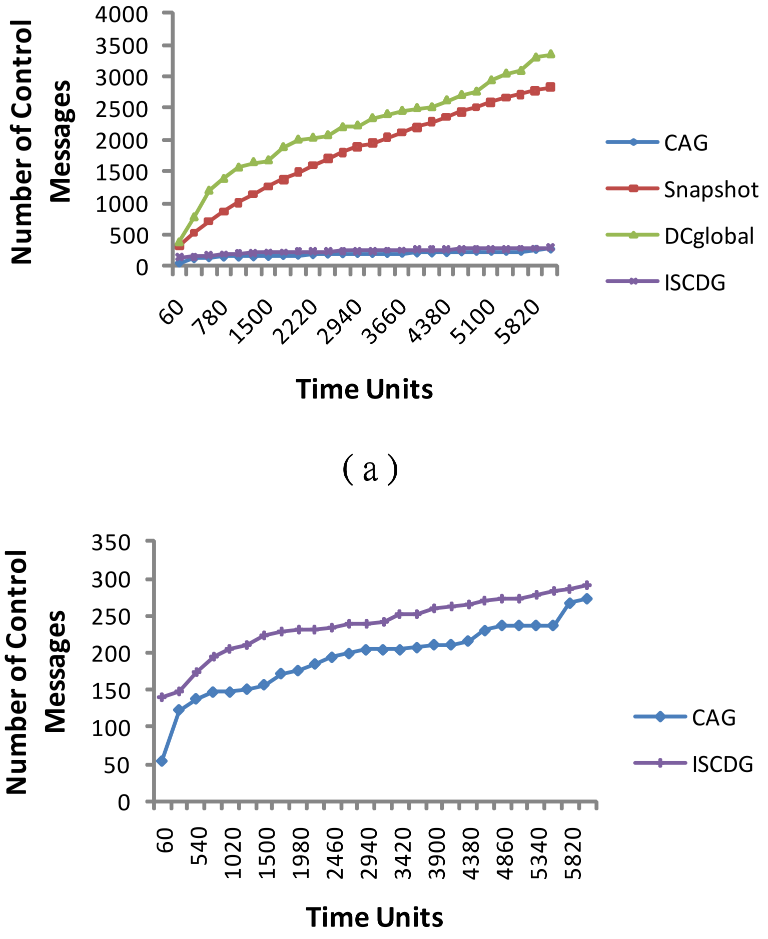


Figure 21: Number of CHs as Time Goes On



(a)

(b)

Figure 22: Number of Control Messages as Time Goes On

4.2 Synthesis Data Set

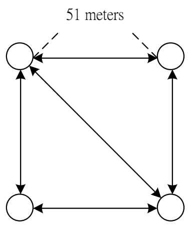
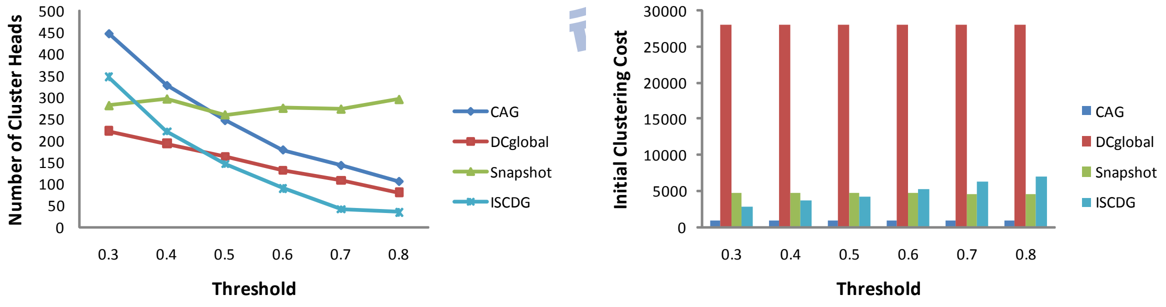


Figure 23: Network Topology

For the synthesis dataset, we generate large number of sensor nodes to evaluate ISCDG in large network size. We deploy sensors to be grid topology and the side of each grid is 51 meters as shown in Figure 23. The transmission range of one node is set to be 100 meters according to specification of MICAz 2.4GHz[10]. Each sensor nodes are set to measure the temperature of the monitored region. We follow the generation function of DCglobal[2], but the temperature value in the center of an event is randomly set between $[15, 40]$°C since the monitor region of our setting is much bigger. Therefore spatial correlation is more obvious in our synthesis data set.

4.2.1 Impact of Threshold



(a) (b)

Figure 24: Clustering Result in 900-Nodes Network

Figure 24 shows the clustering result of a 900 nodes network. As shown in Figure 24(a), ISCDG has a good clustering result when user-tolerable threshold exceed 0.5. According to Figure 24(b), the overhead in clustering of ISCDG is also light compared with DC-global. Based on the result, our in-network mechanism can spend small number of control messages to get a good clustering result even in large scale WSN.

4.2.2 Impact of Network Size

In this subsection, we evaluate our works in different network size, and the threshold is set to 0.8 in the following experiments. Figure 25 shows the clustering cost in different network size. DCglobal cost a lot of energy in clustering while the network size grows up since it is a centralized algorithm. Although Snapshot and CAG still spend less messages in large scale WSN, but their clustering result is not good enough as shown in Figure 24. ISCDG only increase little number of control messages while the network size grows up even through its clustering result is better than other works.

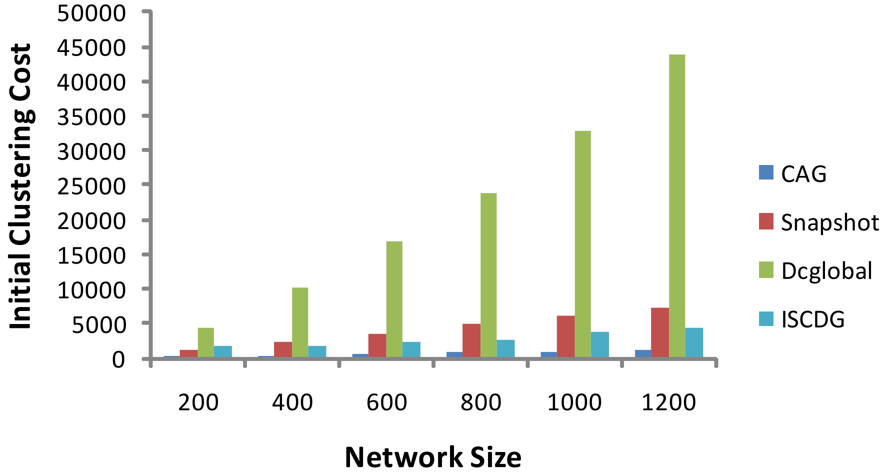


Figure 25: Clustering Cost in Different Network Size

Figure 26 shows the network lifetime in different network size. DCglobal is good in small network size, but the cost of clustering and cluster maintenance damage its network lifetime heavily in large scale WSN. As we discussed in how to handle a low energy node, doing cluster maintenance in sink increases huge loading to nodes which close to sink. The other problem is that maintenance messages are easy to lose due to collision specially on applications which periodically report sensing data. CAG and Snapshot provide a scheme to do cluster maintenance locally, but the number of CHs of them still increase slowly as time goes on since both of them can not actively merge spatial-correlated cluster at each round. Condition of regroup is a principle for nodes to find out spatial-correlated clusters, and ISCDG - phase 2 provide a mechanism to in-network regroup them on RCP. Based on our scheme, we can keep clustering sensor nodes, and only increase light loading to entire WSN. According to these reasons, ISCDG has the best result on network lifetime in our experimental result.

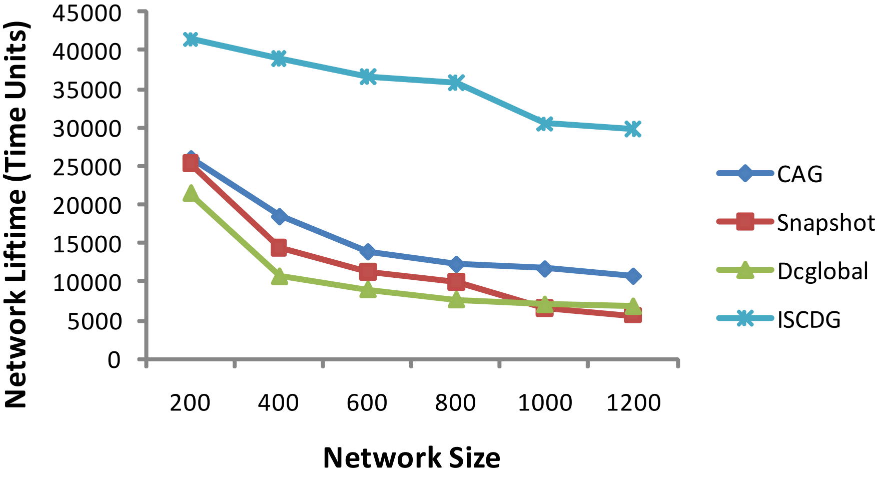


Figure 26: Network Lifetime in Different Network Size

4.3 Impact of Low Energy Node Mechanism

Since we use a different scheme to handle the low energy node, we evaluate ISCDG with and without the scheme in real data set. ISCDG-no is the ISCDG without low energy node mechanism. As shown in Figure 27, the network lifetime of ISCDG is longer than ISCDG-no, that means changing structure of routing tree can help WSN extend its network lifetime. The reason is that nodes which close to sink shoulder heavier loading than other nodes specially in large scale WSN, therefore our proposed mechanism can reduce the loading of low energy nodes.

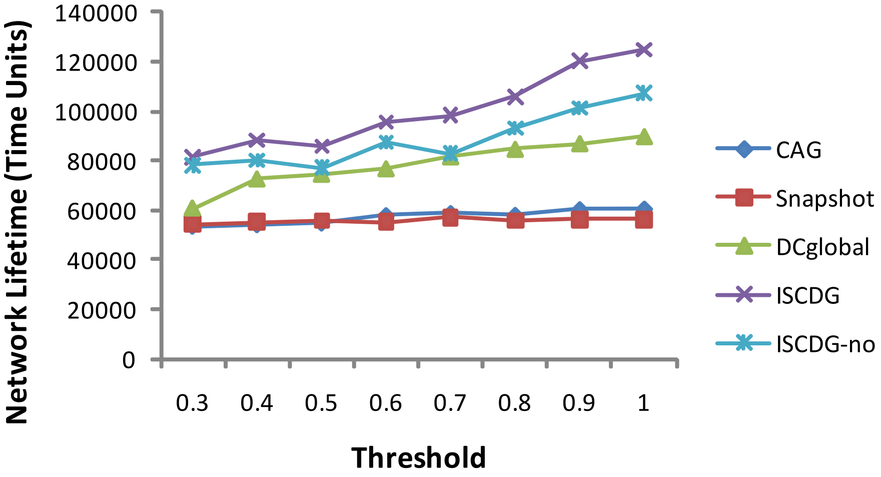


Figure 27: Network Lifetime without Low Energy Node Mechanism

5 Conclusion and Future Work

We design an in-network algorithm to merge clusters based on the spatial correlations between them, using in-network method to obtain a better clustering result can make a larger benefit in large scale network. We consider tough application which periodically gather sensing data, cluster maintenance must be an important problem in this kind of application. Our ISCDG-phase 2 provide a convenient way to keep clustering spatial-correlated clusters that can keep getting benefit from spatial correlation of sensing data. We also design a mechanism to handle low energy node since we notice that loading of entire WSN is skew, nodes more close to sink get heavier loading. As summary, ISCDG is an effective and scalable algorithm to provide good clustering result.

Since the benefit of ISCDG is affect by structure of routing tree, we can design an algorithm specially for ISCDG in the future. The other future work is that we will try to use the idea of regroup in other applications.

References

[1] U.S. GLOBEC Georges Bank. http://globec.whoi.edu/jg/serv/globec/gb/modeling/broadscale_summary.html0. 1995.

[2] C. C. Hung, W. C. Peng, Y. S. H. Tsai, and W. C. Lee. Exploiting spatial and data correlations for approximate data collection in wireless sensor networks. In *2nd International Workshop on Knowledge Discovery from Sensor Data*, 2008.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000.

[4] Y. Kotidis. Snapshot queries: towards data-centric sensor networks. In *Proceedings of 21st International Conference on Data Engineering*, 2005.

[5] S. Madden. http://db.csail.mit.edu/labdata/labdata.html. 2004.

[6] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 5th symposium on Operating systems design and implementation*, 2002.

[7] M.P. Michaelides and C.G. Panayiotou. Exploiting spatial correlation for improving coverage in sensor networks. In *IEEE Global Telecommunications Conference*, 2007.

[8] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. In *Third International Symposium on Information Processing in Sensor Networks*, 2004.

[9] J. Qian, Z. Yu, S. Liu, and J. W. Chong. A pre-selected clustering protocol based on spatial correlation. In *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008.

[10] Crossbow Technology. http://www.xbow.com.

[11] Y. Xu and W. C. Lee. Exploring spatial correlation for link quality estimation in wireless sensor networks. In *IEEE International Conference on Pervasive Computing and Communications*, 2006.

[12] S. Yoon and C. Shahabi. The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Transaction on Sensor Networks*, 2007.

[13] Y. Zhang, H. Wang, and L. Tian. Energy and data aware clustering for data aggregation in wireless sensor networks. In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007.