2007

Performance and security measure of clustering protocols for sensor networks

Pubali Banerjee
Iowa State University

Performance and security measure of clustering protocols for sensor networks

by

Pubali Banerjee

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Doug Jacobson, Major Professor
Tom Daniels
Daji Qiao
Joseph Zambreno
Sarah Nusser

Iowa State University

Ames, Iowa

2007

DEDICATION

I would like to dedicate this thesis to my parents and to my husband Soumen without whose support I would not have been able to complete this work. I would also like to thank my children Shounak and Sanhita for their patience and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. Introduction to sensor networks

1.1 Sensor Networks

Sensor networks are autonomous, self-organizing systems consisting of a large number of small, inexpensive, battery-powered devices deployed densely in the deployment area.These nodes systematically gather information from the environment and transmit the collected information to the sink or base station. In contrast to traditional wireless devices, sensor nodes are characterized by severe power, memory and computational constraints. In contrast to ad hoc network nodes, sensor network nodes are less mobile, are deployed in larger numbers and are in general, more limited in capabilities (1). Sensor networks are mainly used for gathering environmental information like temperature, pressure,humidity, seismic and acoustic data. A wide range of real-world applications use sensor networks namely habitat monitoring, structural monitoring, surveillance, disaster management, inventory management, target tracking, intruder detection etc. (16).

The main challenge in sensor network design is energy conservation. The sensor nodes are battery powered. Also, the nodes need to be scattered in the deployment area and they have to be left unattended for a considerable amount of time. During this time any maintenance or recharging is not an option. Therefore the main focus of sensor network design is to get the job done using the least amount of energy. One of the main areas of research in sensor networks focuses on lightweight protocols for clustering, routing, scheduling and communicating.

1.2 A typical MICA mote

A MICA mote is a commercially available sensor node that has been used widely by researchers and developers. It has all of the typical features of a mote. MICA motes are available

to the general public through a company called Crossbow. These motes come in two form factors: Rectangular, measuring 2.25 x 1.25 by 0.25 inches (5.7 x 3.18 x.64 centimeters), it is sized to fit on top of two AA batteries that provide it with power. Circular, measuring 1.0 by 0.25 inches (2.5 x .64 centimeters), it is sized to fit on top of a 3 volt button cell battery.

The MICA mote uses an Atmel ATmega 128L processor running at 4 megahertz. The 128L is an 8-bit microcontroller that has 128 kilobytes of onboard flash memory to store the mote's program. This CPU is about as powerful as the 8088 CPU found in the original IBM PC (circa 1982). The big difference is that the ATmega consumes only 8 milliamps when it is running, and only 15 microamps in sleep mode. This low power consumption allows a MICA mote to run for more than a year with two AA batteries. A typical AA battery can produce about 1,000 milliamp-hours. At 8 milliamps, the ATmega would operate for about 120 hours if it operated constantly. However, the programmer will typically write his/her code so that the CPU is asleep much of the time, allowing it to extend battery life considerably. For example, the mote might sleep for 10 seconds, wake up and check status for a few microseconds, and then go back to sleep.

MICA motes come with 512 kilobytes of flash memory to hold data. They also have a 10-bit A/D converter so that sensor data can be digitized. Separate sensors on a daughter card can connect to the mote. Sensors available include temperature, acceleration, light, sound and magnetic. Advanced sensors for things like GPS signals are under development.

The final component of a MICA mote is the radio. It has a range of several hundred feet and can transmit approximately 40,000 bits per second. When it is off, the radio consumes less than one microamp. When receiving data, it consumes 10 milliamps. When transmitting, it consumes 25 milliamps. Conserving radio power is key to long battery life.

All of these hardware components together create a MICA mote. A programmer writes software to control the mote and make it perform a certain way. Software on MICA motes is built on an operating system called TinyOS. TinyOS is helpful because it deals with the radio and power management systems for the user and makes it much easier to write software for the mote (21).

1.3 Clustering in sensor networks

As mentioned earlier, sensor nodes are now deployed in large numbers as they are very affordable. The goal of a sensor network is to somehow collect data from all the nodes and send the aggregated data to the powerful base-station. An obvious solution to this problem is clustering. Researchers have tried to come up with clustering protocols so that the inter nodal communication is minimized. Another approach is to alternate active and sleep times and to use different power levels for inter and intra cluster transmissions to conserve energy. The energy required to transmit a packet over a distance d is larger for a greater value of d. Communicating with neighbors is less expensive than communicating with nodes that are situated at a greater distance.Therefore, if the nodes communicate with the neighbors within a cluster only, the communications will be less expensive.In [1] to [6] we see different descriptions of clustering protocols that gather data from all the nodes while conserving energy. In all these approaches, it is claimed that clustering approaches are superior to non-clustering approaches in sensor network scenarios. In (16) it is shown that clustering approaches always outperforms non-clustering approaches when there is a high degree of in-cluster data aggregation. However, under low levels of in-cluster data aggregation, non-clustered approaches perform better. Thus, in general, clustering approaches are considered to be superior, however, there are a few scenarios where non-clustered approaches might be better.

In clustering approaches, all the nodes are efficiently divided into disjoint subsets and within each such subset a leader is elected. The leader communicates with other clusters or the base-station. Clustering can be done for several reasons; in (2) the protocol performs data aggregation to reduce communication overhead. In (19) clustering is done to facilitate queries; in (18)(17) clustering is done to form an infrastructure for scalable routing. Clustering can also be used for efficient network-wide broadcast. As mentioned in (15) single-level clustering protocol is sufficient for many applications; for others multi-level hierarchical clustering may be needed. Cluster heads can establish multiple-hop communications to neighboring cluster heads of the same hierarchy level. It is possible to generalize any single-level clustering protocol to multi-level hierarchical clustering by repeatedly executing the clustering protocol on the

cluster-heads of each level to generate the cluster-heads of the next level and so on.

1.4 Security in sensor networks

Security in sensor networks is an emerging research area. As the technology becomes more mature, security concerns for sensor networks is becoming a key concern. Like other wireless devices in ad hoc networks, sensor networks are vulnerable to many attacks.The main constraint is the limitation of resources and the small amount of energy that can be spared for implementing security protocols. Public key schemes and Diffie-Hellman scheme cannot be implemented in sensor networks for this reason, although there is some research going on in this area at this time (13). The key is to use light-weight security schemes that will provide data confidentiality, integrity and authentication. Sensor nodes typically use unprotected hardware and no physical shield that would stop access to the sensor's memory, processing, sensing and communication components. Because one of the goals of this technology is to keep the cost low, such protection is not likely to be provided in the future. Thus sensor networks are very vulnerable to attacks and security is very important in this regard. In chapter 3 we discuss some of the common techniques used to achieve security in sensor networks.

1.5 Problem Statement and Organization

In this thesis, we provide a detailed study of the performance measure of clustering protocols for sensor networks.In Chapter 2, we introduce some performance metrics that provide a wholesome measure of the performance of clustering protocols. We then use these metrics to measure the performance of a commonly used clustering protocol and provide our results. In Chapter 3, we implement security for the clustering protocol and we observe how the the values of the performance metrics change. In Chapter 4, we introduce our secure clustering protocol and provide our simulation results which show that our protocol outperforms the existing secure clustering protocols.

CHAPTER 2. Optimal configuration of clustering protocols for sensor networks

2.1 Introduction

Sensor networks are often used to monitor a remote environment. Sensor nodes or motes are now readily available and are cheap. There are many sensor nodes that work together to achieve the monitoring function. Often, these sensors are scattered in remote locations where gathering information is otherwise impossible. This requires that the nodes be able to communicate with each other even in the absence of an established network infrastructure. This also means that the locations of the remote sensors may not be known. The other aspect of the sensor network scenario is that the sensor nodes are situated close to each other and there are several of these in number. The base station, which is the control unit for the whole network, is usually situated much farther away in an easily accessible area. The information collected by each node is usually not of great importance; rather the base station needs to know the overall condition of the variable or environment being monitored. Also, it takes far less energy for a sensor node to transmit data to a neighbor than to transmit data to the base station. All these factors together make clustering an obvious choice in sensor networks.

2.2 Clustering challenges in sensor networks

Clustering in sensor networks has its unique challenges. The main goal of a clustering protocol is to minimize the amount of energy spent in order to collect and fuse the data and send it to the base station. The optimal clustering configuration for minimal energy dissipation has to be determined. But the delay in the whole data aggregation process has to be minimized

also. In many sensor network applications, delayed data maybe of no use. Although the data aggregation process has to be energy efficient, it is also desirable not to lose too much information. Thus we need to find the optimal clustering configuration that will yield the minimum energy dissipation, minimum delay and a good quality of the aggregated data.

2.3 Clustering protocols for sensor networks

In the last five years many researchers have tried to come up with clustering protocols that provide solutions to the above mentioned unique challenges. One of the most frequently used distributed clustering protocols for sensor networks is LEACH (2). This protocol uses the radio model to estimate energy dissipated in transmissions. It is assumed that the nodes are all uniformly distributed and the base station is very far away from the deployment area. Each node determines whether it will be a cluster head or not with a certain probability. The cluster heads then collect data from the other nodes, aggregate the data and then send the data to the base station. The cluster-heads are rotated randomly so that the load is balanced, as the most expensive operation is sending data to the base station. The nodes are assigned time slots by the cluster heads and the cluster heads receive data from their members according to the assigned schedules. In this model the nodes are assumed to be at most one hop away from the base station.

Another proposed cluster protocol is PEGASIS (4). This protocol uses the same first order radio model as LEACH. The nodes are deployed uniformly in the deployment area. The BS is fixed at a far distance from the sensor nodes. In this approach, the nodes form a chain among themselves using the greedy algorithm according to their proximity to the base station. The nodes then send the data to their nearest neighbor. The data is aggregated at every step. Thus the key idea is that a node will transmit to and receive data from a very close neighbor. For every round, a head is elected. When the aggregated data reaches the head, it is send over to the base station. Thus at every round only one node performs the most expensive operation, i.e. sending data to the base station. According to the simulation results, PEGASIS outperforms

LEACH in energy consumption as it saves energy by eliminating overhead for dynamic cluster formation, by minimizing the distance of transmission for ordinary nodes and by transmitting to the BS only once per round. However the one node at a time approach introduces a large delay factor as in this case there are no simultaneous transmissions.

Yet another approach is the one used in the TEEN protocol(5). In this protocol the clusters are formed using some criterion maybe distance, like in LEACH. The nodes sense data but they only send data to the cluster head when the sensed value is greater than a certain preset threshold value. Thus there is no scheduled communication between the heads and members. This approach is most useful for time critical applications. By limiting the number of transmissions of the nodes, energy is saved. However this approach may not be suitable for applications where constant monitoring and reporting of data is necessary. A modification of TEEN is suggested in (6). In this protocol called APTEEN, a hybrid approach that combines proactive and reactive networks is used.

Another more recent modification of LEACH is suggested in (7). It is called the EECS protocol. According to this protocol, the nodes form self organizing clusters like LEACH. The BS broadcasts a HELLO message. Nodes estimate the distance from the BS from the strength of the message received. Each node becomes a 'CANDIDATE' to be a head with a certain probability. Within a certain radius, the CANDIDATE with the most residual energy becomes the head. An ordinary node decides which head to join depending on the value of a function that involves the distance to the head and the head's distance to the base station. This scheme claims yo achieve a 135% improvement on LEACH.

2.4 LEACH

2.4.1 The LEACH model

LEACH(2) uses a simplified network model. The sensor nodes are uniformly spread over a rectangular area. The base station is assumed to be very far away from this square area. The nodes organize themselves into local clusters. The data is locally sent to the cluster heads. The cluster heads do the data aggregation and then transmit the aggregated data to the base

station. A cluster head does more work than a non cluster head node. When a cluster head dies, a chunk of the nodes lose communication, i.e. they effectively die. To reduce this, within a cluster, cluster heads are rotated, and all nodes get a chance to be a cluster head equal number of times. This ensures that the energy of all the nodes are balanced. The algorithm works in rounds, one round is defined as setting up the clusters, getting data from all the nodes once, fusing all that data in the cluster heads and sending that data to the base station.

2.4.2 The LEACH Parameters

In LEACH, there are some parameters whose values determine the performance of the algorithm. One such parameter is the number of clusters in a round. The total energy consumed in a round depends on the number of clusters. If the number of clusters is more, the coverage area of each cluster is small, thereby the energy required for communication between the cluster head and the members is less. On the other hand, a fewer clusters would mean less overhead to set the clusters up and fewer nodes will have to serve as cluster heads which is an energy expensive event. Thus there exists an optimal number of clusters for which the energy consumption will be minimal. In LEACH, an expression for the optimal number of clusters, k, is derived.

The analysis of LEACH in(2) does not include the MAC layer time scheduling policy nor does it include the number of routing hops in the networks layer. But the optimal number of clusters will depend on these factors as the energy consumed depends on these. Therefore k will change with the MAC protocol scheduling times and the number of routing hops. Again, in LEACH, the cluster heads set up a schedule for the non cluster heads to transmit. The scheme used is TDMA. The value of k will also affect the delay or latency of the algorithm. It will also affect the quality of the data aggregated. Thus there exists an optimal value of k for which the energy consumed per round, the delay incurred and the quality of data aggregated will all be optimal.

In this chapter, we find an expression for the optimal number of clusters k in the clustering protocol for which the combination of the energy consumed, the delay incurred and the information loss will be optimal.

2.5 Derivation of the optimal number of clusters for minimum energy spend

The problem is to analyze and come up with an expression for the optimal number of clusters in a sensor network that will consume the minimal amount of energy in one round, given the nodes are distributed uniformly and using TDMA in the MAC layer.

We use the simple radio model to calculate the energy spend at the receiver and the transmitter. To transmit a l bit message for d distance, the energy spend is

$E_{TX}(l,d) = lE_{elec} + lE_{fs}d^2$

(using free space model if d is small)

Or

$E_{TX}(l,d) = lE_{elec} + lE_{mp}d^4$

(using multipath model if d is large)

where E_{elec} is the energy used by the transmitter electronics and E_{mp} is the amplifier energy. To receive a l bit message the energy spend is

$E_{RX}(l) = lE_{elec}$

where E_{elec} is the energy used by the receiver electronics. The expression for the energy spend by a cluster head is

$$E_{CH} = lE_{elec}(\frac{N}{k} - 1) + lE_{DA}\frac{N}{k} + lE_{elec} + lE_{mp}d_{BS}^4 \tag{2.1}$$

where $lE_{elec}(\frac{N}{k} - 1)$ is the energy consumed by the cluster head to receive l bits information from $(\frac{N}{k} - 1)$ non cluster heads, E_{elec} is the energy used by the receiver electronics and E_{mp} is the amplifier energy..

The expression for the energy spend by a non cluster head is given by

$$E_{NCH} = lE_{elec} + lE_{fs}d_{toCH}^2 \tag{2.2}$$

where E_{elec} is the energy used by the receiver electronics. In LEACH the MAC protocol used is TDMA. Let T_{CHRX} be the number of seconds for which the cluster head listens to one non cluster head node, let T_{CHTX} be the number of seconds for which the cluster head transmits

to the base station and let T_{NCHTX} be the transmitting time for each non cluster head node. Hence the total time for which each cluster heads listens to $(\frac{N}{k}-1)$ non cluster heads is $(\frac{N}{k}-1)(T_{CHRX})$ seconds. The time for which all cluster heads transmit to base station is T_{CHTX}.Plugging in these time values from the MAC layer in equation(1) we get

$$E_{CH} = lE_{elec}(\frac{N}{k}-1)\mathrm{T_{CHRX}} + lE_{DA}\frac{N}{k} + (lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX} \tag{2.3}$$

where E_{DA} is the energy used for data aggregation. Plugging in the time values from the MAC layer in equation (2) we get

$$E_{nonCH} = (lE_{elec} + lE_{fs}d_{CH}^2)T_{NCHTX} \tag{2.4}$$

The total energy spend in one round is

$$\begin{aligned} E_{total} &= kE_{CH} + (N-k)E_{NCH} \\ &= [lE_{elec}(N-1)\mathrm{T_{CHRX}} \\ &+lE_{DA}N + k\{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\}] \\ &+(N-k)[lE_{elec} + lE_{fs}d_{CH}^{2]}]T_{NCHTX} \end{aligned} \tag{2.5}$$

To minimize E_total,

$$\begin{aligned} &\frac{dE_{Total}}{dk} = 0 \Leftrightarrow \\ &-lE_{elec}T_{CHRX} + \{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\} \\ &-lE_{elec}T_{NCHTX} + lE_{fs}T_{NCHTX}[-d_{CH} \\ &+(N-k)d_{CH}^{'}] = 0. \end{aligned} \tag{2.6}$$

Now, if we assume that each cluster is a circle,

$$d_{CH} = \frac{1}{2\pi}\frac{M^2}{k} \Rightarrow d_{CH}^{'} = -\frac{1}{2\pi}\frac{M^2}{k^2}.$$

where the N nodes are uniformly distributed in a MxM area.

Plugging these in equation (6)

$$\begin{aligned} &-lE_{elec}T_{CHRX} + \{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\} \\ &-lE_{elec}T_{NCHTX} + lE_{fs}T_{NCHTX}\Big[-\frac{1}{2\pi}\frac{M^2}{k} \\ &-(N-k)\frac{1}{2\pi}\frac{M^2}{k^2}\Big] = 0. \end{aligned} \tag{2.7}$$

Let $A = -lE_{elec}T_{CHRX} + \{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\} - lE_{elec}T_{NCHTX}$ and let $B = lE_{fs}T_{NCHTX}M^2/2\pi$. Therefore,

$$\begin{aligned} (A-B)\{\frac{1}{k} + \frac{N-k}{k^2}\} &= 0 \\ \Rightarrow A &= B\frac{N}{k^2} \\ \Rightarrow k &= \sqrt{\frac{BN}{A}}, \end{aligned} \tag{2.8}$$

i.e.,

$$\begin{aligned} k \;=\; &\Big[E_{FS}T_{NCHTX}\frac{M^2N}{2\pi}\Big]^{\frac{1}{2}} \times \\ &\Big[(E_{ELEC} + E_{mp}d_{BS}^4)T_{CHTX} \\ &-E_{ELEC}\{T_{CHRX} + T_{NCHTX}\}\Big]^{-\frac{1}{2}}. \end{aligned} \tag{2.9}$$

2.6 Optimal number of clusters for *energyxdelay* metric

An important factor to consider in any data gathering application is the average delay per round. So far we have found the optimal number of clusters for minimum energy. But a low energy consumption is not the only measure of the performance of the protocol. We need to consider the delay in the protocol also. It is to be noted that in (2) the only metric of performance used is the minimal energy. In (4) the metric energy $\times$ delay is used. It seems to be a good choice because both energy and delay need to be minimized.

We now derive a formula for the optimal number of clusters that will yield the minimum energy $\times$ delay. The non cluster heads transmit to the cluster head according to a TDMA that is set up by the cluster head. Many cluster heads can listen simultaneously as they use CDMA. When a cluster head is done listening and data fusing, it transmits to the base station.

There is no time scheduling, the cluster heads use CSMA to gain access to the medium. The following formula is used to model the delay:

$$delay = (N/k - 1)T_{NCHTX} + k\Delta d$$

where Δ d is the time delay in CSMA.

The total energy times delay is given by

$$\begin{aligned} & \text{energy} \times \text{delay} \\ = \; & (kE_{CH} + (N-k)E_{NCH}) \times \\ & ((N/k-1)T_{NCHTX} + k\Delta d) \end{aligned} \tag{2.10}$$

i.e.,

$$\begin{aligned} & \text{energy} \times \text{delay} \\ = \; & ([lE_{elec}(N-1)\mathrm{T_{CHRX}} \\ & + lE_{DA}N + k\{(lE_{elec} + \\ & lE_{mp}d_{BS}^4)T_{CHTX}\}] \\ & + (N-k)[lE_{elec} + lE_{fs}d_{CH}^2]T_{NCHTX}) \times \\ & ((N/k-1)T_{NCHTX} + k\Delta d). \end{aligned} \tag{2.11}$$

This function can be minimized to obtain the optimal k for minimal energy X delay metric.

2.7 The loss of information metric

The cluster heads fuse data from the member nodes and then transmit the fused data to the base station.If there are fewer cluster heads, more energy is saved. But energy is saved at the expense of information. The base station only knows the aggregate information fused at the cluster heads. Thus, if there are more cluster heads, then the base station has more information about the local regions and vice versa. Therefore, the information loss is an important parameter that is a measure of the efficiency of the algorithm. The amount of information loss is dependent on the number of clusters in the network. It is important to

understand that in some scenarios it is fine to lose some amount of information. For example, if the network is monitoring the maximum temperature of the whole area, then the base station need not know the temperature sensed by each node. However, if at a future time the network needs to monitor the maximum temperature, say at the top left 50 square feet square, it might not be able to do so. Next, we formally determine an expression for the total loss of information.

In Shannon's Information Theory, one of the ways of measuring the information content of a signal is by using the entropy function. In this paper, we choose to use the expected entropy as a measure of the information content of the data.

In LEACH, each node senses and then transmits l bits of data. The cluster head then fuses the $\frac{N}{k}$ l bits of data into m bits of data where m is less than $\frac{N}{k}l$.

At each ordinary node, entropy of l bit signal sensed is given by

$$E_l = -\sum_{n=1}^{l} p_n log(p_n) \tag{2.12}$$

where p_n is the probability with which the nth bit will be 0 or 1 value.

Therefore, entropy of all the l bit signals sensed at the N nodes can be denoted by NE_l.

At a cluster head, aggregated data is m bits in length. The entropy of this data is given by

$$E_m = -\sum_{n=1}^{m} p_n log(p_n) \tag{2.13}$$

We define information loss ratio as the ratio of the information loss for data aggregation and the total information content of the sensed data. This can be formulated by the following expression

$$\Delta i = \frac{NE_l - kE_m}{NE_l} \tag{2.14}$$

The value of Δi will be between 0 and 1. A smaller value will mean a good quality of data. This is a decreasing function with the number of clusters. Ideally if every node reports its data to the base station then there will be no information loss. Thus the more the number of clusters the smaller will be the value of Δi. It is best to choose an acceptable range of for the value of Δi.

2.8 The variance of estimation

In many scenarios the sensor nodes are used to monitor a variable in the deployment area. The variable monitored can be modeled using a function of the location of the sensors. Let this function be represented by $f(x, y)$ where (x_i, y_i) is the location of the ith sensor in the deployment area. Let there be i sensor nodes in the MxM deployment area. The sensors use the clustering protocol to report their readings to the CHs. The CHs perform data aggregation using a data aggregation function and send values to the base station for the clusters. In a given cluster, the value that the CH sends to the base station is the value of the estimated signal for all the members. Let the estimated signal be denoted by $f'(x, y)$. We define the variance of estimation, v as follows:

$v = \sum_i (f(x, y) - f'(x, y))^2$ for all i.

A larger value of this metric indicates a poor quality of signal estimation and vice versa.

2.9 Optimal number of clusters for energy×delay ×information loss metric

We now combine the energy, delay and information loss metrics to obtain a good measure of the performance of the algorithm. The energy spend, the delay and the loss of information all have to minimized to obtain the optimal configuration of the network. Hence we minimize the product of the three. From (10) and (17) we obtain,

$$\begin{aligned} & \text{Energy} \times \text{delay} \times \text{info. loss} \\ = \; & \Big([lE_{elec}(N-1)\mathrm{T}_{\mathrm{CHRX}} + lE_{DA}N \\ & + k\{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\}] \\ & + (N-k)[lE_{elec} + lE_{fs}d_{CH}^2]T_{NCHTX} \Big) \times \\ & \Big((N/k)T_{NCHTX} + k\Delta d \Big) \times \left(\frac{NE_l - kE_m}{NE_l}\right). \end{aligned} \tag{2.15}$$

By minimizing this equation, we can obtain the optimal number of clusters for minimal energy spending and minimal delay and minimal information loss.

2.10 Simulation results

We use a 100m × 100m deployment area and 100 nodes to simulate the sensor network environment. The values of the different parameters used are the following: $E_{fs} = 10 * 10^{-12} Joules$; $Emp = 0.0013 * 10^{-12} Joules$; M=100;N=100; $E_{elec} = 50 * 10^{-9} Joules$; $E_{da} = 5 * 10^{-9} Joules$; $T_{nchtx} = 0.5$; $T_{chtx} = 1$;$T_{chrx} = 1$;d=100;l=6; Δ d=0.05 sec; $d_{fs} = 0.5 sec$.

We obtain the different values of the energy consumed per round, the delay and the information loss for different configurations of the clusters. The metrics are plotted against the number of clusters. The number of clusters are plotted along the x axes and the metrics are plotted along the y axes.

In Figure 1 the energy consumed in each round of LEACH is plotted against the number of clusters from a physical layer standpoint. We see that the optimal number of clusters here is 3. Figure 2 is a plot of energy × delay versus number of clusters from a physical layer and MAC layer perspective. We see that in this case the optimal number of clusters k is 8. Figure 3 is a plot of energy × delay × information loss versus the number of clusters. For this the optimal number of clusters obtained is 20. We get a different plot in this case and the energy × delay × info loss function is minimum for k=20; ie 20 clusters gives the optimal configuration.

The plots show that when MAC layer time scheduling is used to determine the optimal number of clusters, the formula for optimal clusters obtained is different from the physical layer formula. When we use energy × delay×information loss as our measure, we see that the optimal number of clusters k is very different.

It is also possible to adjust the performance measure according to requirements. For example, if in a certain scenario, information loss is less important compared to energy dissipated and delay, then we could use $(informationloss)^c$ instead of $informationloss$ where c is a small constant between 0 and 1. We could choose the value of c depending on the scenario. This can be done for energy or delay also.

In Table 1 we show the different values of the performance metrics when the simulation of LEACH is run in network simulator using 100 nodes and a 100X100 square deployment area with the base station located at (75,150). For variance, we calculate variance between the true

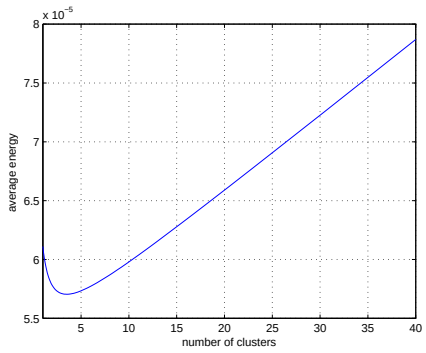

Figure 2.1 Average Energy per round

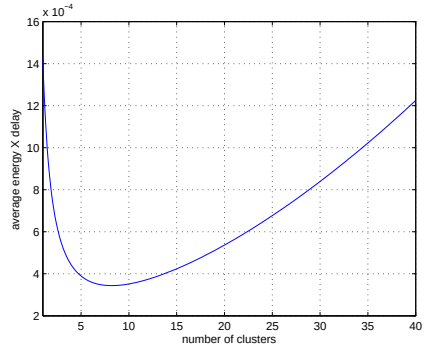

Figure 2.2 Average Energy X Delay per round

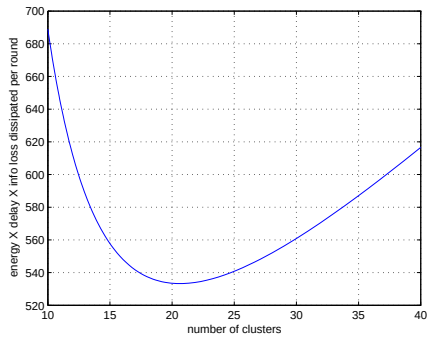

Figure 2.3 Average Energy x Delay x info per round

Table 2.1 LEACH simulation results

no. of CHs	Life Time	CHs alive	First node dies	data	delay	info loss	variance
4	379.4	3	205.5	29300	48.2	0.9401	18.38
6	293	5	299.19	40403	47.9	0.9260	12.74
7	283.19	6	152.38	21363	47.3	0.9137	11.06
8	248.8	7	185.9	19143	46.4	0.9014	9.50
20	201.5	12	43.53	5694	40.4	0.7534	3.99

and the sensed average of the sensed data. From this table we can see how the values of the different parameters change with the cluster numbers.

2.11 Conclusions

In this chapter, we have designed a method to determine the optimal configuration of a clustering protocol for sensor networks. We use the clustering protocol (2)LEACH in this paper. Our method can also be used to determine the optimal configuration for other clustering protocols. We define and formulate three metrics that evaluate the performance of the network with regard to energy spent, delay and information loss. We then optimize the product of these three metrics to obtain the optimal configuration. This analysis should serve as a guideline to users for deciding on the number of clusters when initializing the protocol. As the analysis covers the joint effect of more than one performance measure of a sensor network, the optimal configuration thus obtained will give an overall better performance. As we have discussed in this chapter, a combination of different metrics, weighted or un weighted can be used to determine the optimal clustering configuration for a given scenario.

CHAPTER 3. Security and optimal configuration of clustering protocols

3.1 Introduction

In the last chapter we introduced the different parameters to measure the performance of clustering protocols, namely, energy dissipated, delay and quality of aggregated data. In this chapter, we would like to study how these metrics are affected when we try to secure these protocols.

We have used energy dissipation as the main metric to determine the optimal configuration. We also used latency or delay in the whole data aggregation process as a performance measure. In many sensor network applications, delayed data maybe of no use. Although the data aggregation process has to be energy efficient, it is also desirable not to lose too much information. Therefore we proposed a method to find the optimal clustering configuration that will yield the minimum energy dissipation, minimum delay and a good quality (limited information loss) of the aggregated data. In chapter 1 we have shown how the optimal clustering configuration can be obtained for the LEACH protocol using the above mentioned metrics.

A key concern for clustering protocols in sensor networks is security. In a military battle field scenario, security of communication is a prime concern. In this chapter, we suggest a method for applying security to the LEACH protocol which provides integrity, confidentiality and authentication to the packet exchanges. We then study the performance of the secure clustering protocol using some metrics namely energy spent, delay and information entropy ratio.

3.1.1 Security in clustering protocols for sensor networks

In a typical sensor mote a small amount of resources are left for security and applications. This is insufficient to even hold the variables for asymmetric public key based cryptographic algorithms like RSA and Diffie-Hellman. Thus public key based systems do not work for sensor networks. Because of the resource constraints another solution is to use global keys. This is feasible but a global key based system does not provide the desired level of security. On the contrary, complete pair-wise keying between nodes provides the best possible security, but it is not a choice for sensor network because of the resource constraints. We briefly discuss three different approaches for sensor network security.

In 2001, a security protocol called SPINS(8) were proposed for sensors networks in general. The authors propose two protocols: SNEP protocol that provides data confidentiality, two-party data authentication, integrity and freshness and uTESLA, a key chain based protocol that provides authentication for broadcast data. In SNEP, data is encrypted according to the following format: $E = [D]_{(K_{encr},C)}$ where D is the data, K_{encr} is the encryption key and the counter is C. The MAC is $M = MAC(K_{mac}, C|E)$. The keys K_{encr} and K_{mac} are derived from the master key K using the one way key chain scheme. When node A sends a message to node B it sends

$A \rightarrow B : [D]_{(K_{encr},C)}, MAC(K_{mac}, C|[D]_{(K_{encr},C)})$

Since the counter value is incremented after each message, the same message is encrypted differently each time. Thus this scheme provides semantics security. The counter value is long enough such that there is no repetition during its lifetime. Authentication can be verified by verifying the MAC. The counter value in the MAC prevents replay. The communication overhead is low as the counter state is stored at the sender and receiver and need not be sent. The other protocol proposed in this paper is μ TESLA which is the authenticated broadcast protocol. In the setup phase of this protocol, the sender generates a sequence of keys from an initial key using a one way hash function. When broadcasting, the entire time is divided into slots and each time slot is associated with one key of the key chain. Packets broadcast in a time slot are encrypted using its corresponding key. The sender makes that key available after

a few time slots. This paper proposes security protocols for sensor networks in general, not for clustering architectures or LEACH.

In the paper by Ferreira et all in 2005 (9), an approach to provide security for LEACH has been described. This is one of the first solutions proposed to provide security for LEACH. In this approach, each node stores a key, X_x that it shares with the base station, and a group key, K_i that it shares with the group. From X_x, the key holder derives K_x using one way key-chain approach. The CHs broadcast advertisement messages. An advertisement message is a concatenation of its own id with a MAC value produced using K_x. The BS receives all these, verifies the MACs and sends a list of valid CHs. The ordinary nodes store the advertisement messages and send join requests to CHs which are on this list. Thus the BS authenticates the CHs. The ordinary nodes send data to the CH. They also send a MAC value produced with the key that they share with the base station. The CH aggregates all the data, concatenates all the MACs and sends it to the BS. The BS station verifies all the MACs and thus authenticates all the nodes. This solution makes the BS do most of the security verification, thus the resources of the sensor nodes are preserved. However although this solution provides CH authentication and some data confidentiality it does not provide two party data confidentiality.

A fairly new approach to key distribution in sensor networks is the random pre deployment key distribution scheme. One of the first papers to discuss this in the context of sensor networks was a paper by Eschenauer et all (10). According to this scheme, a large pool of keys is first generated. The nodes are then given a randomly chosen subset of keys from this large pool prior to deployment. Two nodes can find a common key with a certain probability. This scheme works well for sensor networks as the nodes do not need to store a large number of keys. Many other approaches based on random pre deployment key distribution have been proposed for sensor networks in general.

In 2006, a solution based on random pre deployment key distribution was provided for the LEACH clustering protocol. The new security enabled LEACH is called Sec LEACH (11). Most of the key deployment methods before this paper were suited for general sensor networks. These schemes are designed for nodes that need to interact only with a static set of neighbors.

According to this protocol, LEACH has some unique requirements. Since the clusters are formed randomly and periodically, any node needs to be able to join any cluster head. For key distribution, first a large pool of S keys is generated. Then each node is given a key ring of size m drawn randomly and without replacement from S. For each node X a pseudo random function is used to generate its unique id id_x. id_x is then used to seed a pseudo random number generator to produce a sequence of m numbers. Rx, the set of key ids assigned to X, can be obtained by mapping each number in the sequence to its correspondent value modulus S. Also, each node is given a key that it shares with the base station. During the advertisement phase, the CHs broadcast their ids and a nonce along with their advertisement messages. The ordinary nodes can compute the key ids of the CHs using the pseudo random number generator and the CH's id as seed. An ordinary nodes chooses to join a cluster which is the closest to itself and with which it shares a common key. Thus the nodes will not always choose the closest CH. Also, some of the nodes may be orphans and will have to directly send packets to the base station which is an expensive operation. The main drawbacks of this solution are the following: the clustering configuration deviates from the optimal clustering configuration and there are orphan nodes which increase the energy consumption.

In this paper we propose to add security to LEACH using random pre deployment key distribution like (11). We want to study how the optimal configuration of the protocol will change when we add this security feature.

3.1.2 Key Distribution

Like in LEACH, a cluster head (CH) communicates with the member nodes in the cluster and the base station (BS) only and not with the other CHs and their members.

We suppose that there is a large pool S of keys available. For each of the n sensors, we select a set of m keys randomly, without replacement from S. Further, the selection of keys for all the sensors in a group are done independently. These keys will be used for authentication and secure communication between the member sensors and the CH and between the CHs and the base station.

Depending on the relative sizes of S and m, any two sensor nodes may share a common

(non-empty) set of keys with a certain probability. In addition, each sensor is also given one (or a small number of) key(s) for communication with the BS before deployment. These are to be used when a sensor becomes a CH (or is an orphan node) and sends data to the BS.

Note that depending on the key pools of two sensors in a group, they may or may not be able to communicate. The probability that two given sensors in a group communicate is given by $(1 - p_{\text{non}})$, where

$$p_{\text{non}} = \frac{\binom{s}{m}\binom{s-m}{m}}{\binom{s}{m}\binom{s}{m}}. \tag{3.1}$$

Here s, the size of the key pool and m is the key ring size.

In the above equation, the denominator gives the total number of possible key assignments to two sensors without any restrictions, and the numerator gives the total number possible choices of m keys for the first sensor multiplied by the number of possible choices for the second such that it has no (common) key from the key set selected for the first.

3.2 Performance metrics for the secure version of LEACH

3.2.1 Energy spent

The problem is to analyze and come up with an expression for the optimal number of clusters in a sensor network that will consume the minimal amount of energy in one round, given the nodes are distributed uniformly and using TDMA in the MAC layer.

We use the simple radio model to calculate the energy spend at the receiver and the transmitter. To transmit a l bit message for d distance, the energy spend is

$E_{TX}(l, d) = lE_{elec} + lE_{fs}d^2$

(using free space model if d is small)

Or

$E_{TX}(l, d) = lE_{elec} + lE_{mp}d^4$

(using multipath model if d is large)

where E_{elec} is the energy used by the transmitter electronics and E_{mp} is the amplifier energy. To receive a l bit message the energy spend is

$E_{RX}(l) = lE_{elec}$

where E_{elec} is the energy used by the receiver electronics.

The expression for the energy spend by a cluster head is

$$E_{CH} = lE_{elec}(\frac{N}{k} - 1) + lE_{DA}\frac{N}{k} + lE_{elec} + lE_{mp}d_{BS}^4 + lE_{cry} \tag{3.2}$$

where $lE_{elec}(\frac{N}{k} - 1)$ is the energy consumed by the cluster head to receive l bits information from $(\frac{N}{k} - 1)$ non cluster heads, E_{elec} is the energy used by the receiver electronics and E_{mp} is the amplifier energy and $lEcry$ is the energy spent to encrypt an l bit message.

The expression for the energy spend by a non cluster head is given by

$$E_{NCH} = lE_{elec} + lE_{fs}d_{toCH}^2 + lE_{cry} \tag{3.3}$$

In this secure protocol, the nodes that are not able to find common keys with any of the CHs cannot communicate with the CHs. We call these nodes the orphan nodes. The expression for the energy spent by an orphan node in given by

$$E_{orphan} = lE_{elec} + lE_{mp}d_{BS}^4 + lE_{cry} \tag{3.4}$$

Also the CHs that have no non cluster heads communicating with them are essentially orphan nodes.

In LEACH the MAC protocol used is TDMA. Let T_{CHRX} be the number of seconds for which the cluster head listens to one non cluster head node, let T_{CHTX} be the number of seconds for which the cluster heads and the orphan nodes transmit to the base station and let T_{NCHTX} be the transmitting time for each non cluster head node. Hence the total time for which each cluster head listens to $(\frac{N}{k} - 1)$ non cluster heads is $(\frac{N}{k} - 1)(T_{CHRX})$ seconds. The time for which all cluster heads transmit to base station is T_{CHTX}. Plugging in these time values from the MAC layer in equation(1) we get

$$E_{CH} = lE_{elec}(\frac{N}{k} - 1)p\mathrm{T_{CHRX}}$$
$$+lE_{DA}\frac{N}{k} + lE_{cry} + (lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX} \tag{3.5}$$

where E_{DA} is the energy used for data aggregation and E_{cry} is the energy used by the base station to encrypt the data before sending it to the base station and p is the probability that two nodes find a common key. Plugging in the time values from the MAC layer in equation (2) we get

$$E_{nonCH} = (lE_{elec} + lE_{cry} + lE_{fs}d_{CH}^2)T_{NCHTX} \tag{3.6}$$

The total energy spend in one round is

$$\begin{aligned} E_{total} &= kpE_{CH} + (N-k)pE_{NCH} + N(1-p)E_{orphan} \\ &= [lpE_{elec}(N-1)\mathrm{T_{CHRX}} \\ &+ lpE_{DA}N + kplE_{cry} + kp\{(lE_{elec} + lE_{mp}d_{BS}^4)T_{CHTX}\}] \\ &+ (N-k)p[lE_{elec} + lE_{cry} + lE_{fs}d_{CH}^2]T_{NCHTX} \\ &+ N(1-p)[lE_{elec} + lE_{cry} + lE_{mp}d_{BS}^4]T_{CHTX} \end{aligned} \tag{3.7}$$

We can minimize this expression to obtain the optimal k.

3.2.2 *energyxdelay* metric

An important factor to consider in any data gathering application is the average delay per round. A low energy consumption is not the only measure of the performance of the protocol. We need to consider the delay in the protocol also.

We now derive a formula for the optimal number of clusters that will yield the minimum energy $\times$ delay. The non cluster heads transmit to the cluster head according to a TDMA that is set up by the cluster head. Many cluster heads can listen simultaneously as they use CDMA. When a cluster head is done listening and data fusing, it transmits to the base station. There is no time scheduling, the cluster heads use CSMA to gain access to the medium. The following formula is used to model the delay:

$$delay = (\frac{N}{k} - 1)pT_{NCHTX} + kp\Delta d + (N)(1-p)\Delta d$$

where Δ d is the time delay in CSMA and T_{NCHTX} is the time taken by a non-cluster head to transmit to the head node..

The total energy times delay is given by

$$\begin{aligned} & \text{energy} \times \text{delay} \\ = \; & (kpE_{CH} + (N-k)pE_{NCH} + N(1-p)E_{orphan}) \times \\ & ((\frac{N}{k} - 1)pT_{NCHTX} + kp\Delta d + (N)(1-p)\Delta d) \end{aligned} \tag{3.8}$$

This function can be minimized to obtain the optimal k for minimal energy X delay metric.

3.2.3 The information loss ratio metric

The cluster heads fuse data from the member nodes and then transmit the fused data to the base station. If there are fewer cluster heads, more energy is saved. But energy is saved at the expense of information. The base station only knows the aggregate information fused at the cluster heads. Thus, if there are more cluster heads, then the base station has more information about the local regions and vice versa. Therefore, the information loss is an important parameter that is a measure of the efficiency of the algorithm. The amount of information loss is dependent on the number of clusters in the network. Next, we formally determine an expression for the total loss of information.

In this chapter, we choose to use the expected entropy as a measure of the information content of the data, as in Chapter 1.

In LEACH, each node senses and then transmits l bits of data. The cluster head then fuses the $\frac{N}{k}$ l bits of data into m bits of data where m is less than $\frac{N}{k}l$.

At each ordinary node, entropy of l bit signal sensed is given by

$$E_l = -\sum_{n=1}^{l} p_n log(p_n) \tag{3.9}$$

where p_n is the probability with which the nth bit will take 0 or 1 value.

Therefore, entropy of all the l bit signals sensed at the N nodes can be denoted by NE_l.

At a cluster head, aggregated data is m bits in length. The entropy of this data is given by

$$E_m = -\sum_{n=1}^{m} p_n log(p_n) \tag{3.10}$$

We define information loss ratio, Δk as the ratio of the information loss for data aggregation and the total information content of the sensed data. A smaller value of Δk corresponds to a better quality of the aggregated data and vice versa.

For the secure flavor of LEACH discussed in this paper, we can formulate the information loss ratio using the following formula:

$$\Delta i = \frac{NE_l - (kpE_m + N(1-p)E_l)}{NE_l} \tag{3.11}$$

where kp is the total number of cluster heads that perform aggregation and N(1-p) is the total number of orphan nodes including the heads that do not have any members and p is the probability that two nodes find a common key.

3.3 Simulation results

We use a 100m $\times$ 100m deployment area and 100 nodes to simulate the sensor network environment. The values of the different parameters used are the following: $E_{fs} = 10 * 10^{-12} Joules$; $Emp = 0.0013 * 10^{-12} Joules$;$E_{cry} = 2 * 10^{-9} Joules$; M=100;N=100; $E_{elec} = 50 * 10^{-9} Joules$; $E_{da} = 5 * 10^{-9} Joules$; $T_{nchtx} = 0.5$; $T_{chtx} = 1$;$T_{chrx} = 1$;d=100;l=1; Δ d=0.05 sec; $d_{fs} = 0.5sec$;$d_1 = d_2 = 1$ $p = 0.9$ (probability of two nodes finding a common key).

We obtain the different values of the energy consumed per round, the delay and the information loss for different configurations of the clusters. The metrics are plotted against the number of clusters. The number of clusters are plotted along the x axes and the metrics are plotted along the y axes.

In Figure 1 the energy consumed in each round of LEACH is plotted against the number of clusters from a physical layer standpoint for a 100 nodes network. We see that the optimal number of clusters here is 3.

In Figure 2 we plot the energy consumed in each round of Secure LEACH against the number of clusters and we see that the energy function has a minimum value at 2, again for

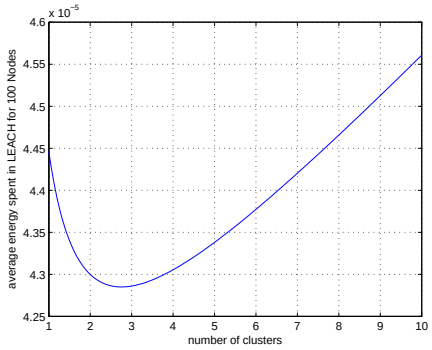

Figure 3.1 Average Energy per round in LEACH for 100 nodes

the 100 node network. When we plot the energy spent against the number of clusters for LEACH and Secure LEACH in figures 3 and 4, we see that the optimal number of clusters is 6 and 9 respectively.

In Figure 5 and 6 we plot the energy $\times$ delay versus number of clusters from a physical layer and MAC layer perspective. We see that in this case the optimal number of clusters k is 8 for LEACH and 10 for Secure LEACH. Figures 7 and 8 are is a plots of energy $\times$ information loss versus the number of clusters. For LEACH the optimal number of clusters obtained is 20 and 3 for Secure LEACH.

The plots show that when MAC layer time scheduling is used to determine the optimal number of clusters, the formula for optimal clusters obtained is different from the physical layer formula. When we use energy $\times$information loss as our measure, we see that the optimal number of clusters k is very different.

It is also possible to adjust the performance measure according to requirements. For example, if in a certain scenario, information loss is less important compared to energy dissipated and delay, then we could use $(informationloss)^c$ instead of $informationloss$ where c is a small constant between 0 and 1. We could choose the value of c depending on the scenario. This can be done for energy or delay also.

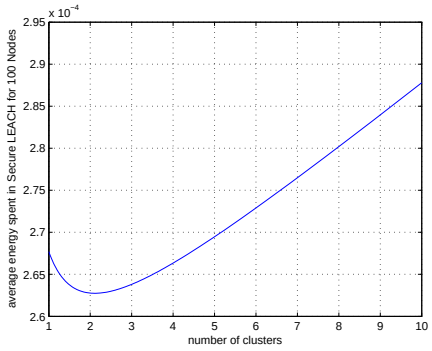

Figure 3.2 Average Energy per round in Secure LEACH for 100 nodes

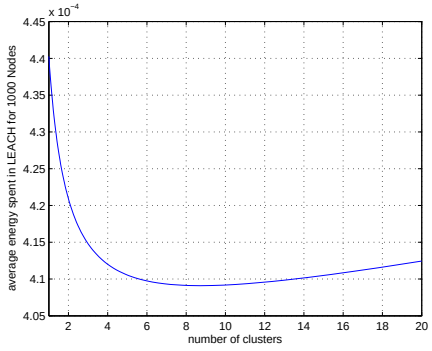

Figure 3.3 Average Energy per round in LEACH for 1000 nodes

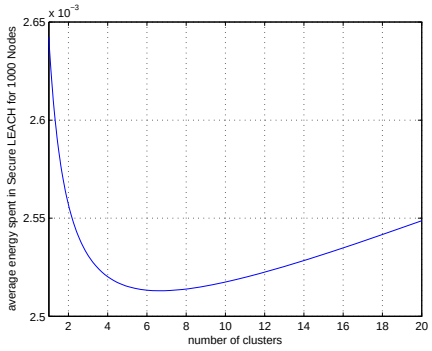

Figure 3.4 Average Energy per round in Secure LEACH for 1000 nodes

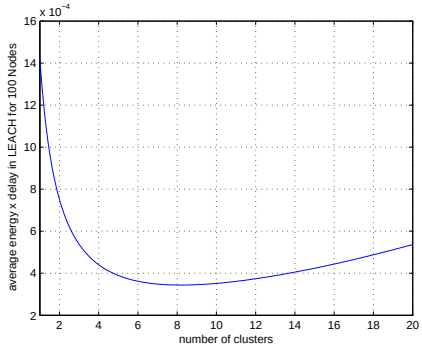

Figure 3.5 Average Energy X Delay per round in LEACH for 100 nodes

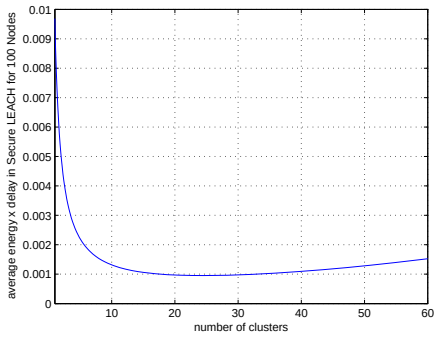

Figure 3.6 Average Energy X Delay per round in Secure LEACH for 100 nodes

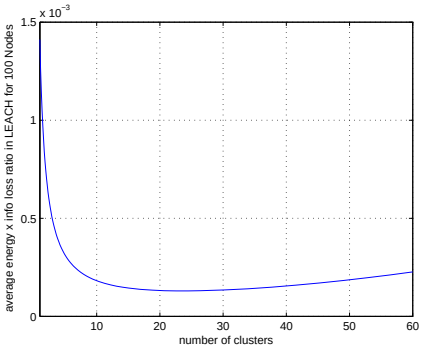

Figure 3.7 Average Energy x info per round in LEACH for 100 nodes

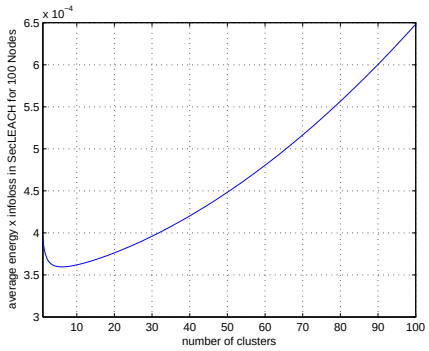

Figure 3.8 Average Energy x info per round in SecLEACH for 100 nodes

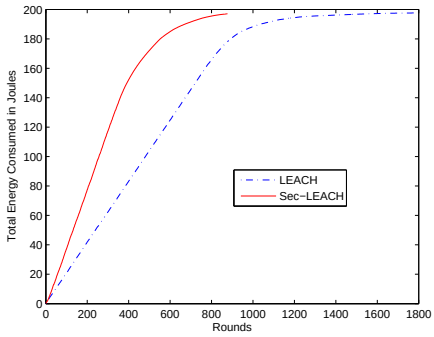

Figure 3.9 Total energy spent for LEACH and SecLEACH

3.4 Conclusions and further work

In this chapter, we have designed a method to determine the optimal configuration of a secure clustering protocol for sensor networks. We see how adding security to an existing protocol changes the optimal configuration. Our method can also be used to determine the optimal configuration for other clustering protocols. We define and formulate three metrics that evaluate the performance of the network with regard to energy spent, delay and information loss. We then optimize combinations of these three metrics depending on the application scenario to obtain the optimal configuration. This analysis should serve as a guideline to users for deciding on the number of clusters when initializing the protocol. As the analysis covers the joint effect of more than one performance measure of a sensor network, the optimal configuration thus obtained will give an overall better performance.

CHAPTER 4. Security and performance analysis of a secure clustering protocol for sensor networks

4.1 Introduction

In this chapter we introduce a new protocol for secure clustering and data transfer for sensor networks. This protocol is a secure solution for the LEACH protocol. In this protocol we use pre deployment key distribution. Before the nodes are deployed they are populated with a couple of keys. We also divide the deployment area into grids and use this knowledge for pre deployment key distribution. We do that so that the probability of two nodes sharing a key is increased. We compare our results with existing solutions and that of LEACH and show that our protocol yields better results.

4.2 The GS-LEACH protocol

4.2.1 Overview

The GS-LEACH protocol employs a grid based deployment of sensors where a certain number (say, n) of sensors are deployed randomly around each point of a regular (square) grid in a planar domain of interest, R. The sensors around a grid point form a cluster for local data compression and take turns as cluster heads (CHs) to communicate with the base station (BS). Sensors in the network are energy constrained while the BS is assumed to be more resourceful and is located outside the region R, possibly at a relatively large distance away from R. Local data compression and rotation of CHs to communicate with the BS prolong the life-time of the network (2). For secure communication, we suppose that prior to deployment of the sensors in a "group" (the set of sensors deployed around a grid point), each of the n sensors is given a set of m randomly selected keys from a large key pool, which it uses for communication with the

group members. An additional key is also given to each sensor for communication with the BS. At the set-up phase of the protocol, each sensor in a group decides to elect itself as CH with some probability, independently of the other members of the group, such that on the average, there is only one CH per grid point. Each CH then sends out a short message and its key information to the members in the group. The members in the group that have a common key with the CH then respond by sending in a join-request message and the id of the common key to the CH. The members who do not have a common key with the CH sit out the particular round and take part in the next round with a higher probability of becoming a CH. A suitable assignment of these probabilities balances the energy levels of the sensors in every n rounds. After receiving the join-requests, the CH sets up a TDMA schedule and sends a confirmation message to each member in the group with its time slot information. This allows the members to shut-off their transmitters at all times except the scheduled time intervals and save energy. The set-up stage is next followed by a much longer steady-state, where the members sense data from the environment and send encrypted data to the CH, which then decrypts these, performs data aggregation, and sends it to the BS using its unique BS-encryption key.

In the next few subsections, we describe the details of the various steps involved in the protocol. We conclude this subsection by listing some of the important features of the GS-LEACH protocol and their implications:

1] Secure communication of information between the clusters and the BS;

2] Use of disjoint key pools for different clusters ensures that capturing the keys of a single group will not affect the security of the rest of the network;

3] Localized communication of the CSs with the CHs makes it energy efficient. Compared to SecLEACH, where a non-CH sensor may communicate with any of the CHs, some of which may be at a great distance away, the GS-LEACH only communicates in a small neighborhood of the grid point;

4] Scalable to regions of different shapes and sizes, driven by the grid;

5] Uniform coverage due to the grid structure of the clusters, thereby guaranteeing a better quality of the information collected;

4.2.2 Deployment

Suppose there are N sensors to be deployed over a planar domain R. We use a grid based deployment and suppose that R is covered with points from a square-grid of increment h in each direction (East-West/ North-South, say). In practice the orientation of the grid may be adjusted depending on the orientation of the domain, if necessary. Let k denote the number of grid points falling in R. Here, k, the number of grid points in R can be adjusted by choosing the increment size h of the grid and is a parameter of the protocol. Efficiency/performance of the protocol depends on the choice of k.

At each grid node, $n = N/k$ sensors are deployed randomly, according to some two-dimensional probability distribution. Depending on the application and available deployment mechanisms, the distribution of the sensors may or may not be uniform around the grid point. For example, in case of an air drop from an airplane, it is reasonable to assume that the sensors have a two-dimensional bell-shaped distribution such as a truncated bivariate normal distribution with the center at the grid point and standard deviation proportional to the height from which the sensors are dropped. On the other hand, in case of a controlled manual deployment over a flat region, the distribution can be uniform over a square of sides h centered at the grid point. However, our protocol does not assume a specific deployment distribution.

4.2.3 Key Distribution

Like in the LEACH, a cluster head (CH) in GS-LEACH also communicates with the member nodes in the cluster and the base station (BS) only, but not with the other CHs and their members. We suppose that the sensors deployed around each grid point are to communicate within the group, and are the only eligible members to form a cluster at the grid. Thus, neither the CH at a grid-node nor any of other sensors in a group (deployed around the grid-node) would communicate with the rest of the sensors in the network. This ensures that if one or more of the sensors in a group are taken over by the enemy, security of the rest of the network will not be compromised.

We suppose that there is a large pool S of keys available that is divided into k disjoint

groups of equal size (say, $S_1, \ldots, S_k$); the keys for the sensors at the ith grid node are to be selected from the set S_i, $i = 1, \ldots, k$. For each of the n sensors at node i, we select a set of m keys randomly, without replacement from S_i. Further, the selection of keys for all the sensors in a group are done independently. These keys will be used for authentication and secure communication between the member sensors and the CH of a group deployed around a given grid point. Depending on the relative sizes of S_i and m, any two sensor nodes within a group may share a common (non-empty) set of keys with high probability. In addition to the 'within group' communication keys, each sensor is also given one (or a small number of) key(s) for communication with the BS before deployment. These are to be used when a sensor becomes a CH and sends data to the BS.

To add an additional layer of protection, the key distribution mechanism is modified slightly by incorporating a pseudo-random number generator (PRNG) as in the SecLEACH protocol (11). Here, instead of assigning the keys directly, first we use a pseudo-random function to generate an id for each of the sensors in a given group and use the id to seed the PRNG of a large period $|S| = |S_1| + \ldots + |S_k|$, where $|A|$ denotes the size, i.e., the number of elements of a finite set A. In the next step, we generate a set of m numbers for a member sensor in the ith group that lie in the ith segment S_i of S. A second set of id-s are generated for each sensor in a group for communication with the BS. The advantages of using the PRNG step are two-fold - (i) the enemy must be able to identify and use the PRNG algorithm before it can make use of the id information from a captured sensor node, and (ii) only the id, but not the full set of keys of a given sensor is announced to the member sensors in a group in the set-up phase, thereby reducing the energy spent for transmission and receiving. On the downside, the PRNG functionality adds to the energy and storage burden of a sensor as each sensor has to store this algorithm and run it to generate the keys at the beginning of every set-up round. However, in typical applications, the sensors have enough storage capacity to store a relatively small number of keys and the energy needed to run the algorithm is lesser than the energy needed to transmit and receive m keys for a moderately large m. Thus, the PRNG step will be added in the basic formulation of our protocol.

Note that depending on the key pools of two sensors in a group, they may or may not be able to communicate. The probability that two given sensors in a group communicate is given by $(1 - p_{\text{non}})$, where

$$p_{\text{non}} = \frac{\binom{s}{m}\binom{s-m}{m}}{\binom{s}{m}\binom{s}{m}}. \tag{4.1}$$

Here $s = |S_i|$, the size of the ith key pool S_i (assumed to be the same for all $i = 1, \ldots, k$). In the above equation, the denominator gives the total number possible key assignments to two sensors without any restrictions, and the numerator gives the total number possible choices of m keys for the first sensor multiplied by the number of possible choices for the second such that it has no (common) key from the key set selected for the first.

4.2.4 Cluster Head Selection and cluster formation

Clusters in GS-LEACH are formed using a distributed algorithm, as in LEACH, where individual nodes take decisions to become a CH independent of the rest of the nodes. In this step, the major difference between the two protocols lies in the assignment of probabilities to individual sensors to be CHs. More specifically, at the beginning of the $(r+1)$th set-up phase (at time t, say), sensor i performs a coin tossing experiment with a certain probability $p_i(t)$ of getting a head and decides to be a CH if the toss results in a head. To balance the energy load of the sensors in a group, the probability $p_i(t)$ is chosen such that each sensor becomes the CH approximately once in $N/k = n$ rounds. Since all sensors within a group may not be able to participate in all rounds, we further choose $p_i(t)$ to take into account the energy differentials among the different sensors that sit out one or more rounds. Let $C_i(t)$ be the indicator of the event that sensor i has not been a CH in one of the past r rounds and let r_{1i} be the number of times that sensor i has not been an ordinary member of the cluster. On any given round, the latter event occurs when sensor i does not have a common key with the key ring of the corresponding CH. We set the probability $p_i(t)$ as

$$p_i(t) = \begin{cases} 0 & \text{if } C_i(t) = 0 \\ \frac{\delta}{n-r} + \frac{(1-\delta)r_{1i}}{\sum_j r_{1j}} & \text{if } C_i(t) = 1, \end{cases} \tag{4.2}$$

where $\delta \in (0,1)$ is weighting factor that depends on the relative levels of energy spent in a round by a CH and an ordinary participating sensor and where the summation over j extends over all nodes that have not been a CH in the previous r rounds (and may or may not have sit out one or more rounds). The rationale behind the formulation of $p_i(t)$ is as follows: if a sensor has been a CH in one of the past r rounds, then it has already served the expected number of times (i.e., once) to be a CH in n rounds and therefore, it is ruled out from becoming a CH until the completion of the n rounds. On the other end, a sensor that has not been a CH in the past r rounds is assigned a nonzero probability to the CH in the next round. Note that at the end of the rth round, the expected number of CHs from a given group is r, and hence, on the average, the remaining $(n-r)$ sensors are elegibile to be the next CH. However, even within the set of sensors that has not been a CH yet, each may have a different number of sit-outs. The assigned probability $p_i(t)$ has a component, $\delta/(n-r)$ that is identical for all such sensors for not having been a CH up until round r and a second component that is higher for sensors that sat out more number of rounds.

A problem with the above probability assignment is that each sensor must know the values r_{1j} for all non-CHs in the group, which may not be available. As a result, we replace the denominator by the total expected number of sit outs up to round r. Note that each r_{1j} is a random variable with the Binomial (r, p_{non})-distribution, where p_{non} is given by (1). Further, the expected number of sensors that appear in the sum in $p_i(t)$ is $(n-r)$. Hence, the expected total number of sit-outs upto round r is $(n-r)rp_{\text{non}}$. Substituting this in the above formula for $p_i(t)$, we get

$$p_i(t) = \begin{cases} 0 & \text{if } C_i(t) = 0 \\ \frac{\delta}{n-r} + \frac{(1-\delta)r_{1i}}{(n-r)rp_{\text{non}}} & \text{if } C_i(t) = 1, \end{cases} \tag{4.3}$$

Note that the expected number of CHs for the $(r+1)$th round is

$$E\Big([no.ofCHs]\Big) = E\Big(\sum_{i:C_i(t)=1} p_i(t)\Big) = E\Big(i : C_i(t) = 1\Big)E\Big(p_{11}(t)\Big) = (n-r)\Big[\frac{\delta}{n-r} + \frac{(1-\delta)Er_{11}}{(n-r)rp_{\text{non}}}\Big] = 1.$$

4.2.5 An Algorithmic description of the GS-LEACH:

- *I. Deployment phase*

1. Fix the grid locations over the planar domain R.
2. Each sensor x in the ith *group* is given its id, id_x, the PRNG algorithm, and a separate key $\mathrm{id}_{x,BS}$ for communications with the BS.
3. The ith group members are deployed around the ith grid point randomly.

- *II. Set-up phase*

4. Available sensors in each group elect themselves as CHs using the probabilities $p_i(t)$ of (3.2).
5. Each CH broadcasts an advertisement message using its id, id_H and a nonce to avoid replay.

   $CH \rightarrow network: id_H|nonce$
6. Each non-CH in the group uses the PRNG with id_H to generate the m keys of the CHs from which it receives advertisements, and checks for a common key. If none, it shuts its communication devices for the given round, thereby saving its energy. Otherwise, each node chooses the CH from which it received the strongest advertisement among those CHs that it has at least one common key with. It sends out a join request message consisting of id_x, its own id, id_H, the index c of the common key and the 'nonce' that the CH originally sent out to that CH. The message is protected by a MAC that is generated using the common key between this node and the CH.

   $x \rightarrow CH: id_x|id_H|c|nonce|$

   $mac_{k_c}(id_x|id_H|c|nonce)$
7. CHs send out a time schedule to the responding member sensors protected by the common key; each of these member sensors transmit only during the given time slot and are in monitoring mode only during other times to save energy.

- *III. Steady-stage phase*

8. A communicating non-CH sensor x sends out a message consisting of id_x,id_H, signal, mac_{k_c}.

$x \rightarrow$ *CH:*

$id_x|id_H|nonce|signal$

$|mac_{k_c}(id_x|id_H|nonce|signal)$

9. A CH performs local data aggregation and sends to the BS the summary signal id_H, $\mathrm{id}_{H,BS}$, and $\mathrm{mac}_{k_{r_H}}$ where k_{r_H} is the key shared between the CH and the BS.

   CH→BS:

   $id_H|id_{BS}|fuseddata|nonce$

   $|mac_{k_{H,BS}}(id_H|id_{BS}|fuseddata|nonce)$

4.3 Security Analysis of GS-LEACH

4.3.1 Security Features

There are some security features build in the GS-LEACH protocol. The key pools for the different grids are disjoint subsets of the main key pool. Thus even if some keys of one grid are captured, these captured nodes would give out absolutely no information about any of the nodes in the other grids. This feature of the protocol makes it more secure.

The key distribution mechanism incorporates a pseudo-random number generator (PRNG). Here, instead of assigning the keys directly, first we use a pseudo-random function to generate an id for each of the sensors in a given group and use the id to seed the PRNG of a large period $|S| = |S_1| + \ldots + |S_k|$, where $|A|$ denotes the size, i.e., the number of elements of a finite set A. In the next step, we generate a set of m numbers for a member sensor in the ith group that lie in the ith segment S_i of S. A second set of id-s are generated for each sensor in a group for communication with the BS. Thus the key ids are transmitted and not the keys themselves. Unless the PRNG algorithm is known, the key ids are of no use to the eavesdropper.

The CHs broadcast the advertisement messages in the clear without any encryption but an eavesdropper can only get the key ids of the cluster heads which cannot be readily used. The join requests are encrypted and a MAC as well as a nonce is used. The use of the CH id and the sender id provides authentication; the use of the MAC provides integrity and the use of the common symmetric key between the CH and the sender node provides confidentiality. The use

of the nonce provides freshness of the packet and protects against replay attacks. Similarly, in the other packet exchanges, namely when a non-cluster head sends data to the CH and when a CH sends data to the BS, integrity, authentication, confidentiality and data freshness are provided by the protocol. Although confidentiality and authentication are provided, because some of the nodes within a grid share keys, the confidentiality and authentication provided are not absolute.

4.3.2 Resistance against attacks

The attacks on clustering protocols can be classified into two classes, outsider attacks and insider attacks. When a malicious node attacks the network, it is an outsider attack. The malicious node can be more powerful than the clustered nodes which can make the attack very powerful. When a cluster head or a node becomes a malicious node the attack is an insider attack. These can be very hard to detect.

Node Takeover: A node in the network may become compromised. This can happen in several different ways. The node maybe taken over by the enemy or the enemy may jam that node and pretend to be that node itself. All the keys of the node are known to the enemy. According to this protocol, the malicious node can attack and compromise only those nodes that it has a common key with. Thus the malicious node cannot attack any node outside its own grid. However, because there is no pair-wise key sharing between nodes, a malicious node can compromise other nodes, the rate of which is evaluated in the next section using the *compromiseratio* parameter.

Spoofing Attack: An outsider malicious node may pretend to be a cluster head. It may send cluster head advertisement messages to the nodes and force the nodes to send data to itself. This attack scenario is impossible for GS-LEACH unless the outsider has compromised some of the keys. Also, as before this attack will be contained within the nodes that the spoofer shares a common key with and within its grid.

Sybil Attack: In this type of attack a malicious node pretends to be a different node or many different nodes at the same time. In the clustering scenario, the malicious node will try to steal the identity of the cluster heads. The attacker needs to somehow guess the identity of a

cluster head. It can then pretend to be the cluster head and thus force all data to get to itself. This is different form the spoofing attack. In the spoofing attack, the malicious node does not steal the identity of the head; it merely pretends to be the cluster head. In case of GS-LEACH, since the CHs have different sets of keys, it will be difficult for the attacker to pretend to be many heads at the same time unless several nodes have already been compromised. A mere eavesdropper can easily get the id of a CH but unless the keys of the CH are known, the id is not of much use. Thus an attacker needs to compromise some nodes before it can launch a Sybil attack in the GS-LEACH scenario.

Packet Flood attack: In this type of attack scenario, a node is flooded with packets to eat up all its resources. In another flavor of this scenario, the Hello Flood attack, an entire network is flooded with advertisement message from the attacker. The attacker in such scenarios is usually much more resourceful and has unlimited energy and larger transmission range. So all the nodes choose the attacker as cluster head even though it is so far away. This diminishes the life-time of the network. Yet another flavor of this scenario is the Sinkhole Attack where the attacker sends out advertisement messages that make the victim look like a cluster head or base station. All the data is then directed towards the victim who cannot deal with so much data. These kinds of attacks on the extreme can result in DOS attacks. In GS-LEACH, since a node chooses another node as its CH only if it has a common key with it, the nodes will not necessarily choose the malicious node as the CH. However, because the advertisement message is not encrypted, a node can be bombarded with many advertisement messages from a powerful enemy.

Eavesdropper attack against TDMA: An eavesdropper can watch the traffic for a few rounds and can figure the sequence in the TDMA scheduling. This information can be used to jam the nodes in that order and make the network virtually non functional. In GS-LEACH, the TDMA schedule is encrypted using the common key shared between the node and its CH, hence the attacker needs to actually compromise this key in order to launch this attack.

Black Hole Attack: This is more relevant in cases of multi hop clustering protocols. In this scenario, the attacker can isolate the base station by compromising the neighbor nodes of the

base station. Once all the nodes around the base station have been compromised, the sensed data cannot reach the base station and the base station becomes a black hole. Since, the CHs from all the grids send data to the BS, this type of attack is not relevant for this one hop protocol.

Select forward: The attacker who spoofs or compromises a CH selectively forwards the sensed data. As before, in case of GS-LEACH an attacker can launch this type of an attack only if a key has been compromised. This type of an attack can be very difficult to detect.

4.3.3 Evaluation of the quality of security

As discussed earlier, if the sensor nodes in a network stores unique keys for communicating with every other node, the system would be perfectly secure. However, due to resource constraints of the nodes, this is not possible. There has to be a balance between the security level and the number of keys stored. The ratio of the number of keys stored in each node to the total number of keys in the key pool is a measure of the probability that a node is compromised given that another node in the system is already compromised. We call this parameter the security level of the system as in [9]. We formally define the security level of the system as: $sl = 1 - \frac{m}{|S|}$ where sl is the security level, m is the number of keys in each node and $|S|$ is the size of the entire key pool. We observe that for a given key pool size, the number of keys required to be stored in a node in GS-LEACH is less than in SecLEACH in order to maintain the same level of security. Another important parameter that affects security as well as the performance of the network is the probability of connectivity between two nodes which we denote by p_{conn}. We define p_{conn} as $1 - p_{non}$ and p_{non} is defined as

$$p_{\text{non}} = \frac{\binom{s}{m}\binom{s-m}{m}}{\binom{s}{m}\binom{s}{m}}. \tag{4.4}$$

Here $s = |S_i|$, the size of the ith key pool S_i (assumed to be the same for all $i = 1, \ldots, k$). In the above equation, the denominator gives the total number possible key assignments to two sensors without any restrictions, and the numerator gives the total number possible choices of m keys for the first sensor multiplied by the number of possible choices for the second such

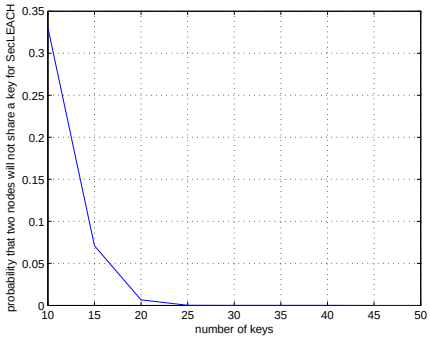

Figure 4.1 p_{non} for SecLEACH for S=100

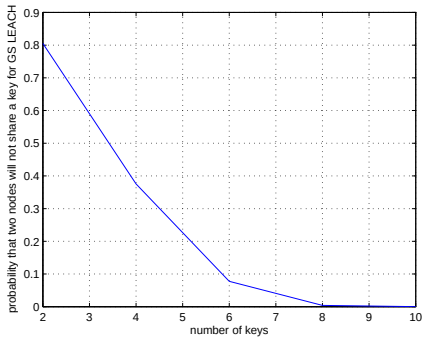

Figure 4.2 p_{non} for GS-LEACH for S=100 and 5 blocks

that it has no (common) key from the key set selected for the first. We provide estimates of the values of this parameter for SecLEACH and GS-LEACH.

In Figure 1 we plot the values of p_{non} for a network with a key pool size of 100 with different key ring sizes. We see that the value of p_{non} is 0.05 when the key ring size is 17, in case of SecLEACH. Thus when each node stores about 17 keys we can say that the connectivity will be very good. From Figure 2 we can see that for a key ring size of 7 p_{non} takes a value of about 0.05 for GS-LEACH. In the case of SecLEACH the value of the security level is 0.83 where as for GS-LEACH the value of security level achieved is 0.94. The next security parameter of significance is the number of keys that are likely to be captured given that some nodes have been compromised. Essentially, when some nodes are captured, this metric gives the estimated number of keys that are likely to be captured using this information. A similar parameter is referred to as resilience against node capture in some papers [9]. We call this the

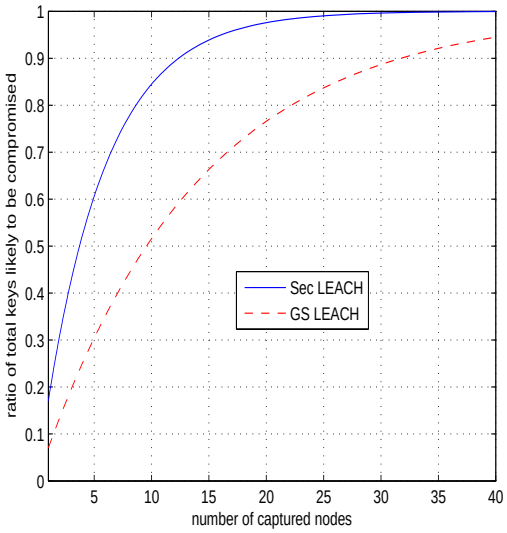

Figure 4.3 Compromise ratio for SecLEACH and GS-LEACH

the *compromise ratio* and we define this as the expected ratio of keys likely to be captured over the size of the whole key pool given that some nodes are captured. We assume that the number of nodes captured is n. Each of these n nodes has m keys, hence the probability that a key has not been captured is $(1 - \frac{m}{|S|})^n$. Hence we can formulate

compromise ratio $= 1 - (1 - \frac{m}{|S|})^n$.

We next plot the *compromise ratio* for SecLEACH and GS-LEACH in figure 3 for a system with 100 key pool size and p_{non}=0.05. We can see that the keys can get compromised much faster in SecLEACH than in GS-LEACH when the same number of keys are already captured.

4.4 Simulation Results

In this section we present our simulation results. We simulate the LEACH, SecLEACH and the GS-LEACH protocol and then compare the performance of all three. For LEACH we use 99 nodes in a square deployment area of 100m X 100m.The base station is located at (150,75) in our model. Each node starts out with 2 Joules of energy. We plot the energy consumption of the network for 6 and 9 clusters in Figure 1. We see that the network performs better when the nodes arrange themselves in 6 clusters when we consider energy consumption as our performance parameter. We simulate the SecLEACH protocol for the same deployment area.We

give 6 keys to each node from a pool of 30 keys. We plot the total energy consumption of the network. We then simulate the GS-LEACH protocol for the same deployment area.However, we divide the 100mX100m square area into 9 squares and distribute the nodes in the squares. The nodes distributed in each square are given keys from one pool. In this model, we have 11 nodes in each grid. We give 6 keys to each node from a pool of 30 keys. We plot the total energy consumption of the network for different values of δ in Figure 4. We observe that the energy consumption is minimum for δ=0.1.

In Figure 5 we plot the energy consumption for the three different protocols over several rounds. We observe that GS-LEACH consumes the minimal energy. SecLEACH consumes the maximum amount of energy. The lifetime of GS-LEACH is the maximum. In the theoretical analysis and in the simulation for GS-LEACH we ignore the energy consumption for the cryptographic steps namely key storage, encryption, key exchange etc. In GS-LEACH, every grid has a cluster-head on an average. In some rounds there are no CHs in a grid or some of the nodes do not have common keys with the head in that grid, hence in some rounds some of the nodes are not able to communicate with the BS. Assuming that the grids are small compared to the total deployment area, we believe that even if the BS receives no data from a grid the overall data sensed by the BS will not be greatly affected. Since this is spread out evenly over the network, the life-time of the network is prolonged compared to LEACH and particularly SecLEACH. In SecLEACH the orphan nodes directly send the messages to the BS which diminishes the network lifetime.

In Figure 7, 8 and 9 we plot histograms of the variances of the sensed signal from the true signal observed over 200 rounds for the three different protocols. Our true signal is of the form

$$f(x,y) = \Big(\sin[2\pi x/100]\Big)\Big(\sin[2\pi y/100]\Big). \tag{4.5}$$

We observe that the variance is the smallest for GS-LEACH. In GS-LEACH the controlled node distribution ensures that the nodes in one grid or cluster which send data to a cluster head are close to each other. This results in smaller variance of the aggregated data.

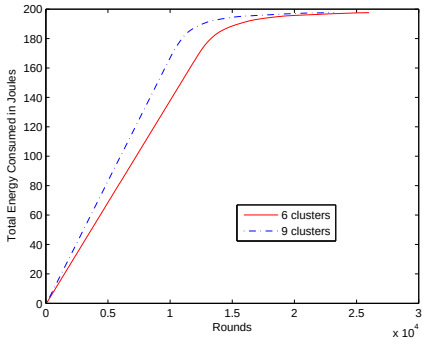

Figure 4.4 Total Energy Consumed in LEACH for different cluster sizes

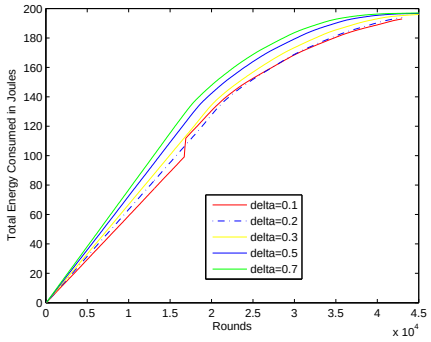

Figure 4.5 Total Energy consumed in GS-LEACH for different δs

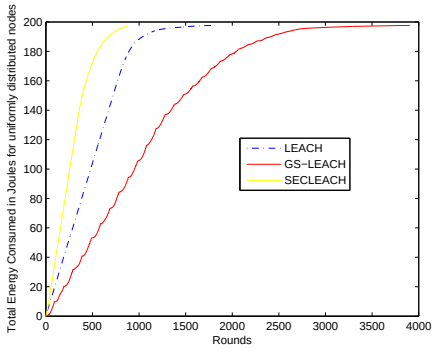

Figure 4.6 Total energy consumed in LEACH, SecLEACH and GS-LEACH for uniformly distributed nodes

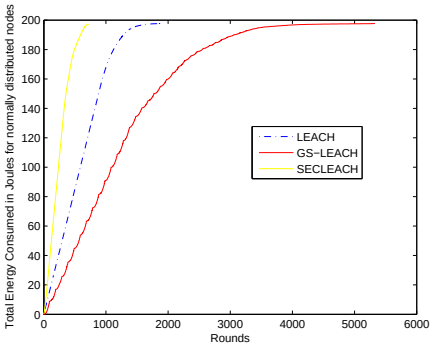

Figure 4.7 Total energy consumed in LEACH, SecLEACH and GS-LEACH for normally distributed nodes

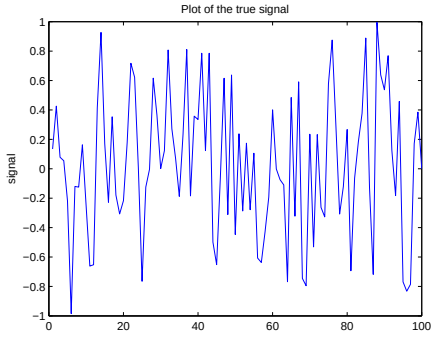

Figure 4.8 True Signal

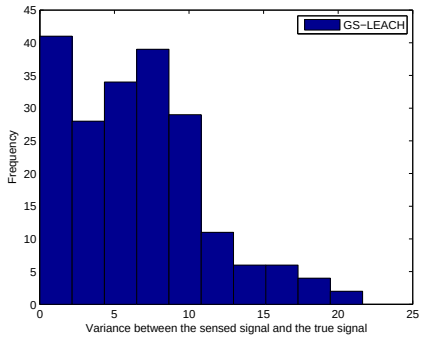

Figure 4.9 Variance of sensed and true signal in GS-LEACH

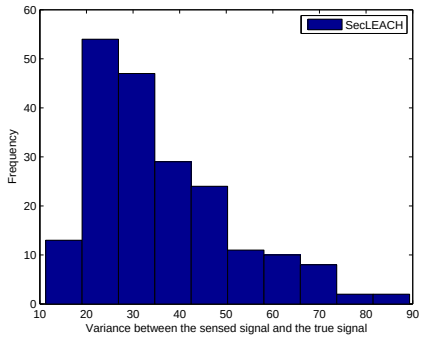

Figure 4.10 Variance of sensed and true signal in SecLEACH

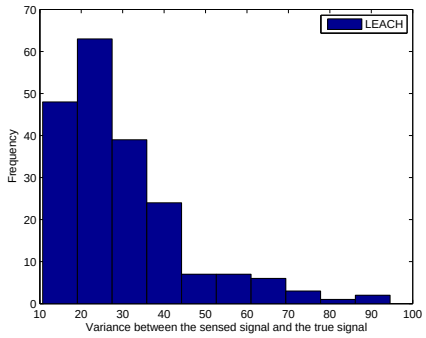

Figure 4.11 Variance of sensed and true signal in LEACH

4.5 Comparison with LEACH and SecLEACH

In this paper we have proposed a new protocol, the GS-LEACH protocol that provides a secure solution to LEACH and it also consumes less energy per round while sensing a signal with least variance. In addition to ensuring security of the information, two other major advantages of GS-LEACH over the standard version of LEACH are first, scalability/ robustness against different shapes and sizes of the coverage domain. Secondly, coverage of information/signal over the domain of interest is significantly more accurate under the grid structure compared to the LEACH, which may have a poor coverage of the signal across the region of interest over several frames or rounds, particularly when all CHs lie in a small sub-area of the domain R, due to unconstrained/independent self-election of the CHs over all of R. We observe that the variance of the sensed signal is the least for GS-LEACH. The mean variance in case of GS-LEACH in our experimental setup is 6.539 while that for LEACH is 29.6455.

Although the security construct of the GS-LEACH is based on the SecLEACH, it improves upon SecLEACH on the following aspects: firstly, prolongs the life of the network under the GS-LEACH, by taking turn in becoming CH through an energy-balancing probability assignment, while in SecLEACH, the orphan nodes either communicate with the BS during each frame, or sit out a given round. In the first case, as the BS is presumably far away, each orphan node drains its energy significantly in a single round, and reduces the lifetime of the network. In the second case, the SecLEACH protocol fails to balance the energy differentials between the orphan and the communicating nodes.The second improvement is the following: Communicating non-CH sensors in the GS-LEACH communicate only *locally* with a member of the group, which presumably has a shorter distance, as all members of the group are deployed in a neighborhood of a common grid point. In comparison, a non-CH sensor in SecLEACH can communicate with the CHs that share a common key, but may be located anywhere in the entire domain R. Thus, it appears that on the average, the distance to the closest communicating CH will be larger.

Also, as seen in section 3.3, in GS-LEACH to maintain the same level of connectivity, fewer keys are required. For the same level of connectivity, the value of security level achieved for GS-

LEACH is higher. Also, the value of the compromise ratio discussed is lower for GS-LEACH which indicated that it has more resistance to attacks.

SecLEACH, which also employs an un-constrained self-election rule for CH formation like the LEACH, suffers from the same disadvantages as the LEACH vs. GS-LEACH, outlined above. We observe that the energy consumption is the most for this algorithm. We also observe that the variance of the sensed signal is the least for GS-LEACH. The mean variance in our experimental setup for the true signal given by equation (10) is 6.539 for GS-LEACH and 36.16 for SecLEACH.

4.6 Conclusion and Future Work

We have described a secure grid bases clustering protocol in this chapter. We have shown that GS-LEACH outperforms LEACH and SecLEACH when we use the energy consumed as the metric. We have compared LEACH, SecLEACH and GS-LEACH in terms of the quality of the sensed signal. We have observed that GS-LEACH outperforms the other two protocols. We have not used the total amount of data collected at the BS as a metric. Because some of the nodes sit out some rounds, the total data bits sent to the base station is smaller in case of GS-LEACH. The sit outs can be minimized by designing the protocol by choosing the number of block such that the probability of finding a common key between two nodes in the same block is maximized given the key ring and the key pool size. We plan to carry out that analysis in the future. We have obtained some very encouraging results in our initial simulations; we need to simulate the GS-LEACH protocol for different block sizes and different shapes of the deployment area. We believe that for non-rectangular deployment area like and annulus, GS-LEACH will outperform LEACH to a greater extent.We have also done a thorough security analysis of our protocol and we have shown that our protocol is more secure than SecLEACH.

BIBLIOGRAPHY

[1] O. Younis and S. Fahmy (2004). Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *IEEE Transactions on Mobile Computing,, vol. 3, no. 4, pp. 366–379, October-December 2004.*

[2] W.Heinzelman and H.B. Chandrakasan (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications to IEEE 802.11 Standard-1999 Edition vol 1, no. 4, October 2002.*

[3] P.Banerjee and D. Jacobson (2006). Optimal configuration of clustering protocols for sensor networks. *Proceedings of PDCS 2006, Dallas, TX, Nov 2006.*

[4] Stephanie Lindsey and Cauligi S. Raghavendra (2002). PEGASIS: Power-Efficient Gathering in Sensor Information Systems. *Aerospace Conference Proceedings, 2002. IEEE, 2002.*

[5] A.Manjeshwar and D.P. Agrawal (2001). TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *Ist International Workshop on Parrallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 2001.*

[6] A.Manjeshwar and D.P. Agrawal. (2002). APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. *Proceedings of the Parrallel and Distributed Processing Symposium, 2002.*

[7] Mao Ye; Chengfa Li; Guihai Chen; Wu, J. (2005). EECS: an energy efficient clustering scheme in wireless sensor networks. *Performance, Computing, and Communications Conference, IPCCC. 24th IEEE International, 2005.*

[8] A.Perring, R.Szewczyk, V. Well, D. Culler and J.D.Tygar. (2002). *S*PINS: Security Protocol for Sensor Networks. *Wireless Networks.vol 8, no.5, Sept. 2002. Also appeared in MobiCom'01, 2002.*

[9] A.C. Ferreira, M.A. Vilaca, L. B. Oliveira, E. Habib, H. C. Wong and A.A.F. Loureiro. (2005). *O*n the security of cluster-based communication protocols for wireless sensor networks. *4th IEEE International Conference on Networking (ICN'05). vol 3420 of lecture notes in Computer Science, pages 449-458, Reunion Island, April 2005.*

[10] L.Eschenauer and V.D. Gligor. (2002). A key management scheme for distributed sensor networks. *In the 9th ACM conference on Computer and Communications security. pages 41-47.*

[11] L.Oliveria, H. Wong, M.Bern, R. Dahab, A.A.F. Lourerio (2006). SecLEACH-A Random Key Distribution Solution for Securing Clustered Sensor Networks. *In the Fifth IEEE International Symposoum on Network Computing and Applications (NCA'06), 2006.*

[12] T.Pham; E.Kim and M. Moh. (2004). On Data Aggregation Quality and Energy Efficiency of Wireless Sensor Network Protocols-Extended Summary. *Proceedings of the First International Conference on Braodband Networks (BROADNETS'04) 2004.*

[13] Wenliang Du, Ronghua Wang and Peng Ning (2005). An Efficient Scheme for Authenticating Public Keys in Sensor Networks. *Mobihoc 2005, Urbana-Champaign ACM 1-59593-004-3/05/0005.*

[14] Mohamed Eltoweissy, Mohammed Moharrum and Ravi Mukkamala. (2006). Dynamic Key Management in Sensor Networks. *Topics in Ad Hoc and Sensot Networks, IEEE Communications Magazine, April 2006.*

[15] Arjan Durresi and Vamsi Paruchuri. (2005). Adaptive Clustering Protocol for Sensor Networks. *In Aerospace Conference, IEEE 2005 ISBN: 0-7803-8870-4 pages 1:8.*

[16] N. Vlajic and D.Xia. (2006). Wireless Sensot Networks: To Cluster or not to cluster. *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMom'06) 0-7695-2593-8/06.*

[17] C.R. Lin and M. Gerla. (1997). Adaptive Clustering for Mobile Wireless Networks. *IEEE J. Select. Areas Commun.*

[18] S. Banerjee and S. Khuller. (2001). A clustering scheme for hierarchical control in multihop wireless networks. *in Proceedings of IEEE Infocom, April 2001.*

[19] M.Demirbas and H. Ferhatosmanoglu (2003). Peer-to-Peer spatial queries in sensor networks. *3rd IEEE International Conference on Peer-to-Peer Computing (P2P '03) pages 32-39.*

[20] Chengfa Li, Mao Ye, Guihai Chen and Jie Wu. (2005). An energy-efficient unequal clustering mechanism for wireless sensor networks. *MASS 2005 0-7803-9466-6/05.*

[21] Marshall Brain. (2007). How motes work? *http://computer.howstuffworks.com/mote4.htm last accessed: June 25th, 2007*