SECURED CLUSTERING IN WIRELESS SENSOR NETWORKS

BY

SAMIR GOKHALE

Bachelor of Engineering

DAVV

Indore, India

2002

Submitted to the faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2005

SECURED CLUSTERING IN WIRELESS SENSOR

NETWORKS

Thesis Approved:

Dr. Johnson P Thomas
Thesis Advisor

Dr. G.E. Hedrick

Dr. Debao Chen

Dr. A. Gordon Emslie
Dean of the Graduate College

ACKNOWLEDGMENT

I am greatly indebted to my parents for their moral and financial support to complete my Masters degree in Computer Science. Many people have ambitions in life but only few are fortunate to fulfill them. In this regard I am greatly thankful to the almighty for guiding me in the right direction and rewarding my efforts by fulfilling my ambition.

At this moment I must mention some important people who have always been a pillar to the successful completion of my master's program. Firstly, I would like to thank my thesis advisor Dr. Johnson P. Thomas for his valuable time and patience in guiding and supporting me with his ideas and suggestions. Secondly, my heartfelt thanks to my committee members, Dr. G.E.Hedrick and Dr. Debao Chen, for their comments and suggestions about my thesis.

Last but not the least; I would like to thank my friends for having encouraged me during the entire Masters Degree program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

1.1 Existing Technologies in Communication

With advancement in technology demands have increased. In today's world, the user always wants to remain connected to the world of information. The traditional wired systems are no longer enough to suffice current needs. As a result we see computing being applied to more demanding and diverse applications. Computation combined with mobility aids in the development of dynamic networks. One of the important discoveries in this direction is that of Wireless Sensor Networks (WSNs). Wireless Sensor Networks today play a critical role in helping to solve some of the most pressing problems facing human society. These include environmental monitoring, traffic control, detecting terrorism attacks, controlling nuclear power plants, monitoring human health, and detecting enemy movements in battlefield etc. to name a few. These sensors are fragile devices with batteries that provide minimal power. Hence computational and communicational capabilities of these devices are very simple and minimal. Typically, after the batteries are exhausted, the sensors die. Furthermore, there is no backbone infrastructure such as routers or switches in the network; therefore the information sensed by a sensor is routed through other sensors to a base station. In some environments these sensors are deployed in a controlled manner. However, in other environments there is little control on how these sensors are deployed. Hence, these sensors must be able to self-configure with no external support.

Due to constraints caused by limited resources of sensors, any sensor networking protocol must be energy efficient. A number of energy efficient communication protocols for sensor networks have been proposed to extend the lifetime of the network in the literature. It has been shown that clustering reduces the energy required for communications in sensor networks and a large number of communication protocol based on clustering techniques have been proposed in the literature. The security of these networks is important in some types of environments. For example in a battlefield scenario, it is essential to secure the network so that an enemy combatant cannot infiltrate the network either to gather information or to inject false information. A number of security protocols based on key management have been proposed in the literature. However, as far as we are aware of no protocol that takes into account both energy and security has been proposed to-date in the literature. In this thesis we propose a novel approach to secure, energy efficient sensor network communication protocol based on clustering and a key management scheme based on key rings first proposed by Gligor [9]. This scheme is proposed for flat level topology networks (networks that do not have clustering). In this scheme a trust model is established using this key sharing technique. The nodes that share keys are trusted nodes and only they can communicate with each other. Furthermore, we extend this protocol by applying the concept of 'force' to improve configuration of the network, thus further improving the energy efficiency of the protocol. The concept of force was first reported in [14]. This force is used to redistribute the nodes in the network and then clustering is applied to them. In this thesis we propose a number of energy efficient secure protocols. These include a secure clustered protocol,

a secured clustered protocol that takes into account the degree of neighboring nodes and finally a secure cluster protocol that considers both the degree of neighboring nodes and applies the concept of force. Simulations results show that our proposed protocols provide energy efficiency as well as security with the last protocol that applies force providing the best results.

The rest of the thesis is outlined as follows. In chapter two we review the previous work that has been done in energy efficient sensor protocols and secure sensor network protocols. In chapter three we define the problem and the motivation for this work. In chapter four we describe our secured energy efficient scheme and chapter five concludes the thesis with results and future work.

CHAPTER II

LITERATURE REVIEW

2.1 Routing Protocols for Wireless Sensor Networks

Routing in WSNs is very challenging due to several characteristics that distinguish them from contemporary communication and wireless ad-hoc networks. Some of the main reasons, which make routing challenging, are:

- It is not possible to build a global addressing scheme for deployment of sensor nodes due to sheer number of sensor nodes.
- All the nodes are required to sense data and send it to a common node typically a base station or the aggregation point.
- As there are large number of sensor nodes deployed in any network so the data sensed by adjacent nodes is more or less the same. Protocols must therefore be designed to exploit this redundancy and thus save power in the network.
- The sensor nodes are tightly constrained in terms of energy, transmission power, processing capability etc so any routing protocol must be able to do efficient resource management.

Since the evolution of WSNs, different schemes have been devised for routing. Some of the different protocols for data routing are described below.

2.1.1 Data Centric Protocols

In a network having large number of sensor nodes it is not possible to assign global identifiers to each node due to sheer number of nodes deployed. Due to lack of global identification along with random deployment of sensor nodes makes it very difficult to select specific set of sensor nodes to be queried. Therefore data is transmitted from every sensor node within the deployment region with a lot of redundancy. This result in wastage of lot of power of the network so routing protocol that will be able to select a set of sensor nodes and utilize data aggregation during the relying of data have been considered. This consideration has led to the discovery of data centric protocol. In data centric routing sink issues queries to certain regions and waits for the response from those regions. As data is requested through queries, attribute based naming is necessary to specify properties of data. Some of the protocols, which come under this category, are described next.

2.1.1.1 Sensor Protocol for Information via Negotiation (SPIN)

The idea behind SPIN is to name the data using high-level descriptors or metadata. Before transmission metadata are exchange between the sensors via a data advertisement mechanism, which is the key feature of SPIN. Upon receiving new data each node indicates this to its neighbors and the interested neighbors (which do not already have the data) retrieve the data by sending a request message. Thus in this way duplicate data can be prevented. There are three messages in SPIN to exchange data

between nodes: ADV to allow a sensor node to advertise a particular metadata, REQ to allow a sensor node to request the data and DATA that carries the actual data. One of the advantages of SPIN is that topological changes do not affect the communications between nodes, as a node only needs to know about its neighbors. The main disadvantage with spins is that it does not guarantee the delivery of data. For example if the source and destination are far away and the intermediate nodes are not interested in the data, then they may not request it from the source and in turn the destination would not get the data.

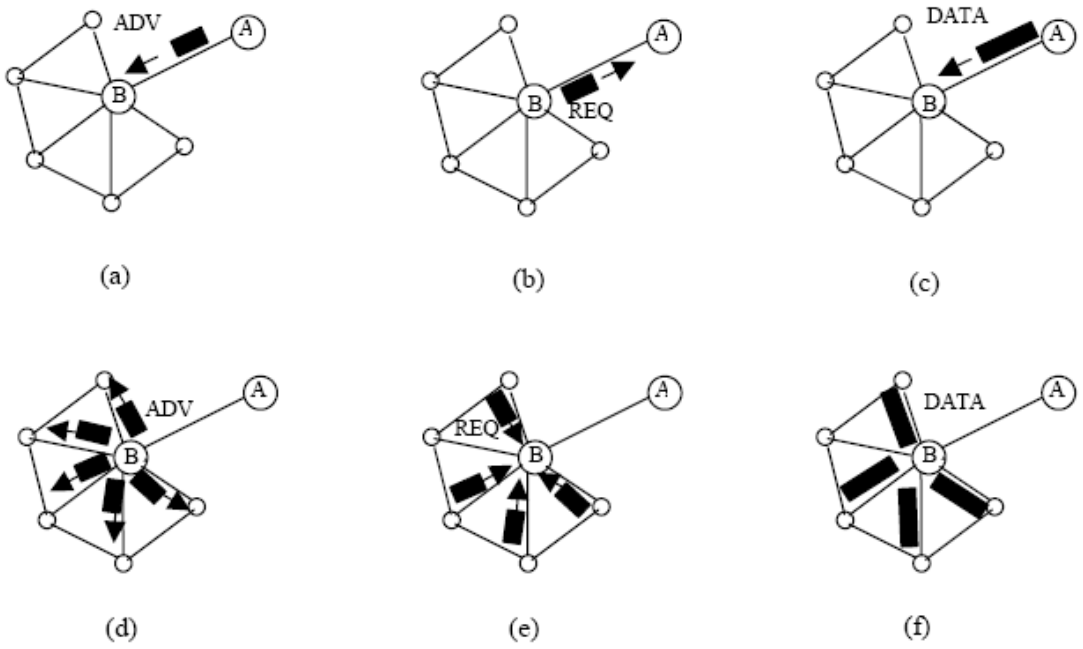

Figure 1. SPIN protocol.

2.1.2 Directed Diffusion

Directed diffusion is an important milestone in data centric routing research. The main idea is diffusing data through sensor nodes by using a naming scheme for data. Node A starts by advertising its data to node B (a). Node B responds by sending a request to node A (b). After receiving the requested data (c), the node B then sends out advertisements to its neighbors (d) who in turn send request back to B (e-f) [2]. This scheme aims to get rid of the unnecessary operation of network layer routing in order to save energy. Direct diffusion suggests the use of name value pair for the data and queries the sensor on an on demand basis by using those pairs. In order to create a query, an interest is defined using a list of attribute value pairs such as name of the object, interval, duration, geographic area etc. A sink through its neighbors broadcasts the interest. Each node receiving the interest can do the caching for later use. The interests in the caches are then used to compare the received data with the values in the interest. The interest entry also contains several gradient fields. A gradient is a reply link to the neighbor from which the interest was received. Hence by utilizing interest and gradients the paths are established between sources and sink. Several paths can be established so that one of them can be selected by reinforcement. The figure below shows the complete working of Directed Diffusion Protocol.

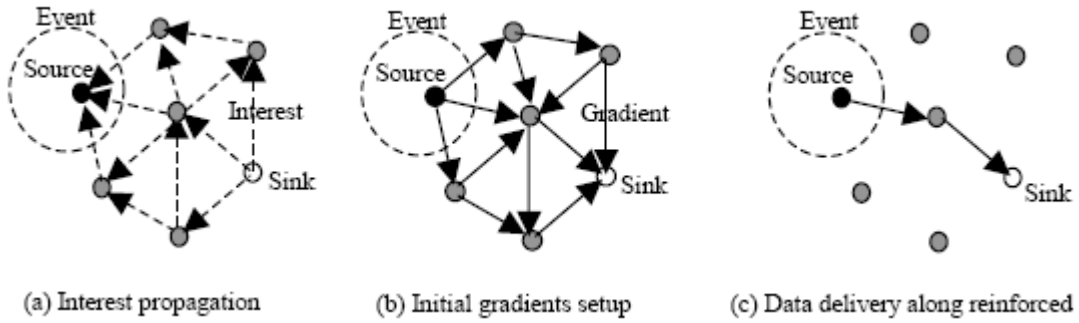

Figure. 2 Directed Diffusion Protocol Phases [2]

The main advantage of Directed Diffusion over SPIN is that in the former, all communications are neighbor to neighbor so there is no need for a node addressing mechanism. Each node can do aggregation and caching in addition to sensing. Caching is a big advantage in terms of efficiency. Furthermore, Directed Diffusion is on demand so there is no need for maintaining global network topology. However Directed Diffusion is not a good choice for applications, which require continuous data as it is based on on-demand query driven data model.

2.1.1.3 Cougar

Cougar is a data centric protocol that views the network as a huge distributed database system. The main idea is to use declarative queries in order to abstract query processing from the network layer functions such as selection of relevant sensor etc. and utilize in network data aggregation to save energy. Cougar proposes an architecture for the sensor database system where sensor nodes select a leader node to perform data aggregation and transmit the data to the gateway sink. The gateway is responsible for

generating a query plan, which specifies the necessary information about the data flow and in-network computation for the incoming query and send it to the relevant nodes. The query plan also describes how to select a leader for the query.

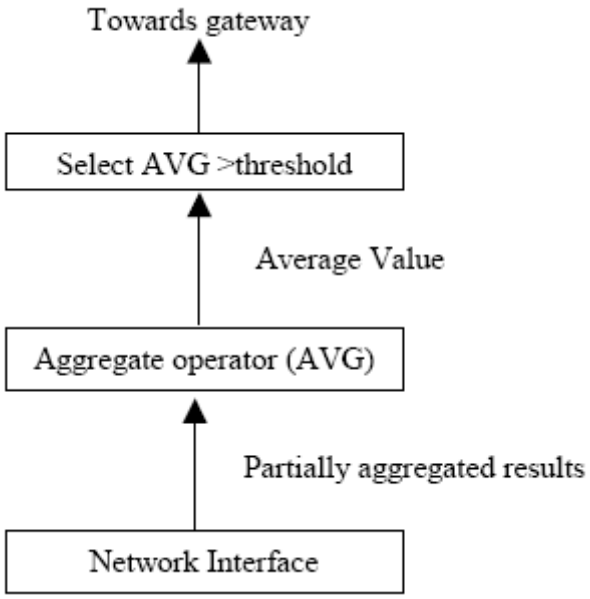

Figure 3. Query plan at a leader node.

The leader node gets all the readings, calculates the average and if it is greater than threshold then sends it to the gateway [2]. Although Cougar provides a network layer independent solution for querying the sensor, it has some drawbacks. First of all introducing an additional query layer on each sensor node will bring extra overhead to sensor nodes in terms of energy consumptions and storage. Second in-network data computation from several nodes will require synchronization i.e. a relying node should wait for every packet from each incoming source, before sending the data to the leader

node. Third the leader nodes should be dynamically maintained to prevent them from failures.

2.1.2 Hierarchical Protocols

The protocols presented till now were all of flat level topology. These protocols have the main disadvantage that they cause network congestion in a densely populated network by causing the gateway to overload. Such overload might cause latency in communication and inadequate tracking of events. Also the flat level topology is not suitable for large scale distributed WSN because sensor nodes are not capable of carrying out long haul communications. To allow the network to cope up with additional load and to be able to cover a large area clustering has been proposed. A cluster is a group of sensor nodes, which carry only localized communication. The whole network is divided into various clusters and for each cluster there is a cluster head, which can communicate with other cluster heads or directly with a base station. The main purpose of clustering is to efficiently maintain energy consumption of the sensor nodes by performing data aggregation and fusion in order to decrease the number of transmitted messages. Cluster formation is typically based on the energy reserve of the sensor and the sensor's proximity to the cluster head. Some of the major hierarchical protocols used are:

2.1.2.1 LEACH

Low Energy Adaptive Clustering Hierarchy (LEACH) was the first protocol, which used a clustering approach. The idea is to form clusters of the sensor nodes based on the received signal strength and use local cluster heads as routers to sink. This leads to saving of energy since transmission is done by such cluster heads rather than by all sensor nodes. All the data processing such as data fusion and aggregation are local to the cluster. Cluster heads change over time to balance the energy dissipation of nodes. The decision is made by the node choosing a random number between 0 and 1. The nodes becomes an cluster head for the current round if the number is less than the following threshold

$$T(n) = P/(1-P*(r \bmod 1/p)) \text{ if } n \text{ belongs to } G$$

$$= 0 \text{ Otherwise.}$$

Where P is the desired percentage of cluster head (e.g. 0.05), r = the current round, and G is the set of nodes that have not been cluster heads in last 1/P rounds. Though LEACH is distributed and requires no global knowledge, it proposes single hop communication inside the clusters, which is not applicable to networks deployed in large regions. Furthermore the idea of dynamic clustering brings extra overhead e.g. head changes, advertisement etc which may diminish the gain in energy consumption.

2.1.2.2 TEEN and APTEEN

Threshold sensitive Energy Efficient sensor Network protocol (TEEN) is the hierarchical protocol designed to be responsive to the sudden changes to the sensed attribute such as temperature. TEEN pursues a hierarchical approach along with a data centric mechanism. After the clusters are formed the cluster broadcasts two thresholds to the nodes. These are hard threshold and soft threshold from sensed attributes. The hard threshold value is the minimum possible value of an attribute to trigger a sensor node to switch on its transmitter and transmit to the cluster head. Thus the hard threshold allows a node to transmit only when the sensed attribute is in the range of interest, thus reducing the number of transmissions significantly. Once a node senses a value at or beyond the hard threshold, it transmits data only when the values of that attribute changes by an amount equal to or greater than the soft threshold. However TEEN is not good for applications where periodic reports are needed since the user may not get any data at all if the thresholds are not reached.

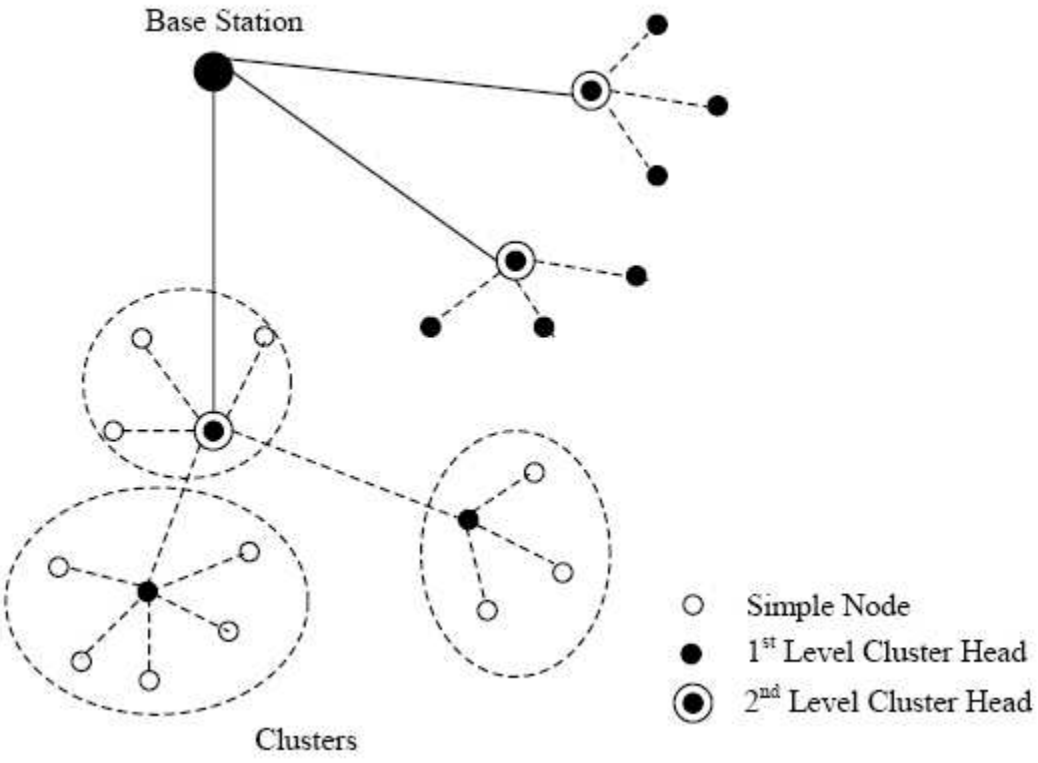

Figure 4. Hierarchical Clustering in TEEN and APTEEN [2]

2.1.2.3 Self-Organizing Protocols

The architecture for self-organizing protocols supports heterogeneous networks that can be mobile or stationary. Some sensors, that can be mobile or stationary, probe the environment and forward the data to a designated set of nodes that act as routers. Router nodes are stationary and form the backbone of the communication. Data through routers are forwarded to more powerful sink nodes. Each sensing node should be reachable to the router node to be a part of the network. Sensing nodes are identifiable through the address of the router it is connected to. The routing architecture is hierarchical where a group of

nodes are formed and merged as needed. The algorithm for self organizing the sensor nodes and creating the routing tables consists of four phases:

- Discovery Phase: The nodes in the neighborhood of each sensor are discovered.
- Organization Phase: Groups are formed and merged by forming a hierarchy. Each node is allocated its address based on its position in the hierarchy. Routing tables of size O (log N) are created for each node. Broadcast trees that span all the nodes are constructed.
- Maintenance Phase: Updating of the routing tables and energy level of nodes is made in this phase. Each node informs its neighbor about its routing table and energy level.
- Self-Reorganizing Phase: In case of partition due to node failures, group reorganizations are performed.

The proposed algorithm utilizes the router node to keep all the sensors connected by performing a dominating set. Since sensor nodes can be addressed individually in the routing architecture, the proposed algorithm is suitable for applications such as parking lot networks where communication to a particular node is required. The major advantage of using this algorithm is the small cost of maintaining routing tables and the routing hierarchy being strictly balanced. The disadvantage is in the organization phase of the algorithm that is not on demand and so introduces extra overhead. Another problem is in the case of hierarchy formation when there are many cuts in the networks. This will be expensive since network cuts increase the probability of applying a reorganization phase. [2]

2.1.3 Location Based Protocols

Most of the routing protocols for the sensor network require location information for sensor nodes. In most cases location information is needed to calculate the distance between two nodes so that energy consumption between the communicating nodes can be estimated. Since unlike traditional wired networks, sensor networks do not have any scheme like IP addressing in place so location information can be utilized in routing data in a energy efficient way. For instance if the region to be sensed is known, using the location of the sensor, the query can be diffused only to that particular region which will eliminate the number of transmissions significantly. There are also many location-based protocols for ad hoc networks such as Cartesian and trajectory based routing, but they are not applicable to sensor networks because they are not energy aware. Some of the Energy Aware location based routing protocols for sensor networks are described next.

2.1.3.1 MECN and SMECN

Minimum energy communication network (MECN) sets up and maintains a minimum energy network for wireless networks by utilizing low power GPS. Although the protocol assumes the mobile network, it is best applicable to sensor networks, which are not mobile. A minimum power topology for stationary nodes including the master node is found. MECN assumes a master site as the information sink, which is always the case for sensor networks.

The main idea of MECN is to find a sub-network, which will have a small number of nodes and require little power for transmission between any two nodes. In this way global

minimum paths are found without considering all of the nodes in the network. The protocol has two phases:

1) It takes the position of a two-dimensional plane and constructs a sparse graph (enclosure graph), which consists of all the enclosures of each, transmit node in the graph. This construction requires local computation in the nodes. The enclosed graph contains globally optimized links in terms of energy consumption.

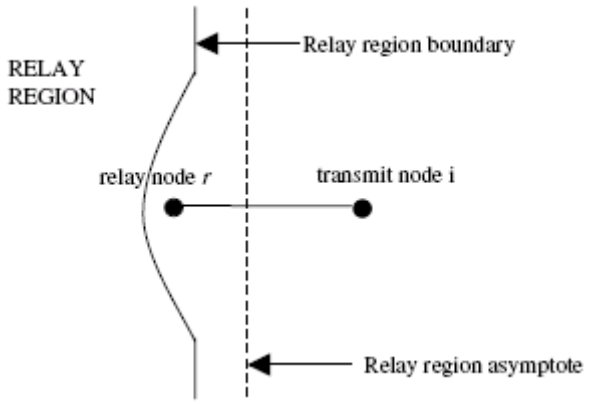

Figure 5. Relay regions of transmit-relay node pair (i, r) in MECN [2]

2) Find optimal links on the enclosure graphs. It uses distributed shortest path Bellman-Ford algorithm with power consumption as a path metric. In case of mobility the position coordinates are updated using GPS.

The Small Minimum Energy Communication Network (SMECN) is an extension to MECN. In MECN it can assumed that every node can transmit to every other node, which is not possible every time. In SMECN possible obstacles between any pairs of nodes are considered. However the network is still assumed to be connected as in the case of MECN. The sub network constructed by SMECN for minimum energy relying is probably smaller than the one constructed in MECN if broadcasts are able to reach to all the nodes in a circular region around the broadcaster. As a result the number of hops for

transmission will decrease. However finding a sub network with smaller number of edges introduces more overheads in the algorithm.

CHAPTER III

PROBLEM DEFINITION

In wireless sensor networks, protocols have been considered separately for security and for energy efficient routing. There have been very few protocols, if any, which consider both of these important parameters. Although sensors have very limited resources, security scheme for WSNs consume a lot of resources such as a sensor node's memory, processing power etc. Unlike their wired counterparts WSNs have very limited power as they are battery powered. Once deployed it is very difficult, if not impossible to change batteries. Furthermore, as the topology of the network is not known prior to deployment, it makes even more difficult to design any security scheme for these networks. To date the most efficient security schemes for WSNs are key pre-distribution schemes. But even these schemes have some drawbacks and prove to be inefficient in the long term [chapter V]. One of the most efficient security schemes for WSNs has been suggested by L. Eschenauer and V. D. Gligor [9]. It works on random key pre-distribution for networks with a flat level topology.

In this thesis we propose to improve the energy efficiency of the security scheme proposed by Gligor [9]. We propose a clustering scheme for energy efficient communication among nodes after this security scheme is deployed in the network. A Cluster is a group of nodes, which are within communication range of one another. Each cluster has a head node, which is responsible for communication inside the cluster. In WSN most of the energy is spent in the communication between different nodes. The power consumed is directly proportional to the second (or higher) power of the distance

between the nodes so that the greater the distance between the nodes, more power will be consumed when two nodes communicate. Clustering reduces the distance to which a node has to communicate and hence reduces energy consumption. One of the most important protocols for a sensor network, which implements clustering, is LEACH [5]. In this a mechanism of cluster formation is suggested, which divides the whole network into clusters. Each cluster has a cluster head and it collects data from all the nodes in its cluster and then sends it to the base station directly. Although this approach results in an increase in the lifetime of the network, this approach can be further improved for energy efficiency. In LEACH, a node chooses by itself to become a cluster head randomly. In contrast, in this work it is the existing cluster head, which decides on which node is going to be the next cluster head. LEACH works in rounds and tries to distribute energy evenly among all the nodes by picking a node randomly as the cluster head. LEACH also assumes that each node in the network has sufficient energy to reach the base station directly which is unrealistic for a large distributed WSN. In this thesis the decision of selection of the cluster head is based on the remaining energy level of a node and it is assumed that each node has enough energy to reach other nodes in its neighborhood (which will be much closer than communicating directly with the base station). Therefore instead of a single hop communication from the cluster head to the base station, as suggested in LEACH, we use multi-hop communication to further reduce the energy dissipation. For carrying out multi-hop communication there should be some efficient trust mechanism. Here we are using the security scheme given by L. Eschenauer and V. D. Gligor [9] to establish this trust model. Once the trust model is established we use balanced clustering to carry out communications. Balanced clustering is a process in

which nodes organize themselves to form clusters such that each cluster has more or less same number of nodes. This helps in even dissipation of energy among nodes. In another variation, first a force as described in [14] is applied on nodes so that they redistribute themselves evenly in the network and then balanced clustering is done which further improves energy efficiency. The whole clustering and security scheme is described in the proposed solution, which increases the lifetime and security of a WSN.

CHAPTER IV

PROPOSED SOLUTION

The main problems with the existing techniques are that either they deal with security or routing in WSNs but not both. When a routing protocol is designed then it makes some assumptions about security of the WSNs and when a security scheme is designed it does not take into consideration energy limitations of the WSNs. Some of the existing techniques, which involve key distribution in WSNs for achieving security, are:

1) Single Mission Key:

   In this scheme all the nodes in the networks share a global key and communications are based on presence of that key with the node i.e. the nodes that have the key are trusted nodes and they can communicate. Although the memory required for storage of keys is minimal, the capture of any single sensor node will compromise the entire DSN.

2) Pair-wise Private Sharing of Keys:

   This solution requires storage of n-1 keys on each sensor node. Hence, for a network of size n, each node will be required to store n-1 keys, one for each node in the network except itself. This will result in storage requirements of n (n-1)/2 keys for the whole DSN. For a network having more than 10000 nodes this storage requirement will be impractical and moreover pair-wise private sharing of keys between any two nodes is not realistic since node communications are limited by radio range of the sensor nodes.

3) Random Key pre-distribution:

Proposed by L. Eschenauer and V. D. Gligor [9] this involves random drawing of k keys out of a pool of P keys and loading each sensor node with a ring having those k keys. This is a probabilistic scheme and the success of the scheme lies in selecting the size of pool P out of which to draw random keys. Any two nodes that share a key have a link between them, which is encrypted by the shared key. Hence a trust model is established using shared keys. The main advantage of this scheme is that it is mid way between the above two schemes. It does not require each node to store numerous keys and also is not vulnerable to single key compromise.

The random key pre-distribution scheme described above is the best among all because it takes into consideration the limited resources of the sensor nodes and also provides good security. The drawback of this scheme is that it is proposed for a flat level topology network and thus does not take into account the energy consumption in the node in the long term. In a flat level topology, when nodes communicate to the base station for long time, their energies reduce drastically and they start to die out. The base station is normally situated at a large distance from the nodes. The other security schemes described in 1) and 2) suffer from a lack of security or excessive storage and energy consumption.

To overcome the deficiencies of existing techniques this thesis proposes a scheme that provides both security and energy efficiency based on clustering in WSNs. The main characteristics of this scheme are:

1) Minimizes energy consumption by balanced clustering. Clusters are formed using local information. Clusters are self-organizing which makes the clustering scheme work effectively even for random deployment. The clustering algorithm ensures that each and every node in the network belongs to some cluster and hence clusters are distributed throughout the network. In our scheme the node's remaining energy and its degree (number of nodes with which the nodes shares keys) collectively is the parameter for deciding its status as head. Before assigning any node to a cluster the energy level as well as degree of cluster head is taken into consideration. If the degree of cluster has already reached the threshold, then the incoming node, which is the new node, which shares a key with the cluster head and wants to become its member, is assigned to another potential cluster head. The Potential cluster head in this case is the node with second highest energy level in the cluster. Now that node is the cluster head of all other incoming nodes because the previous cluster head has reached its limit. Threshold in this case is the maximum number of nodes that a cluster head can accommodate in its cluster. Assigning nodes in this way makes sure that no cluster is overloaded. Although this increases the number of clusters, all the clusters are more or less balanced. Each node is assigned a unique id, to distinguish it from other nodes. The communications to the base station are minimized because instead of each node communicating with the base station, as in the flat level topology network, only cluster heads communicate with the base station in a multi hop fashion. Thus overall network traffic is reduced and energy is saved.

2) Using Gligor's security scheme, a trust model is established which provides for more security by reducing the key storage requirements at the nodes. This is achieved by distributing a key ring on each sensor node prior to deployment. The key ring is distributed such that the network is connected. When nodes share a key, link is formed between them, which is encrypted by that key. A shared key discovery phase is carried out to find which nodes share keys directly. After this a path key discovery phase is carried out. The security scheme described by L. Eschenauer and V. D. Gligor [9] is used to achieve this. The details of the scheme are described below.

3) We also derive some performance metrics, which are used to show that our scheme is better than a scheme which simply employs key pre-distribution for security in WSNs, such as the random key pre-distribution scheme described by L. Eschenauer and V. D. Gligor [9], which is for flat level topology networks and does not take into account the parameters such as uniformity, distance traveled by a moving node and energy consumption. Simulation results show that the scheme proposed in this thesis performs better on these metrics then the scheme described by Gligor.

It is assumed that we have homogenous, mobile and dense WSN. Homogenous means all the nodes have the same capabilities in terms of battery power, memory etc. It is assumed that the key distribution is already done prior to the deployment and shared key and path key discovery has been done. We propose a solution in which a set of nodes is initially chosen to be cluster heads by the base station based on energy level and degree of the node. Once heads are selected then the balanced clusters are formed such that each

cluster will have more or less the same number of nodes. A node belongs to a cluster if and only if it shares a key with the cluster head. If a cluster head node has already reached its maximum degree then other incoming nodes are assigned to the node, which has second highest energy level in the cluster, and it will form its own cluster with itself being the cluster head. After formation of clusters the nodes within the cluster communicate with the cluster head in a single-hop. For ensuring security we store a key ring in the memory of each sensor node prior to deployment. The keys for the key ring are drawn randomly from a pool of P keys. The nodes, which have common keys, can communicate with each other. If two nodes do not share a key and want to communicate they can do this by using a path key, which is established during the network initialization phase. This kind of key pre-distribution reduces the memory requirement to store keys at each node. Now nodes only need to store specific number of keys rather than keys for all the nodes in the network. Although nodes within a cluster will communicate only with the cluster head, keys may be shared between two nodes that are in different clusters, to take care of mobile nodes. As nodes move across clusters they come in contact with different cluster heads based on their position. So to facilitate communication of new nodes with the cluster head, keys are shared between any two nodes. Therefore nodes communicate among themselves only during the time of network initialization and whenever a mobile node enters a new cluster. For the rest of the time once clusters are formed all the communication takes place between nodes and the cluster head.

4.1 Network Initialization

Network initialization phase consists of deployment of nodes in the network. In this phase a trust model is established by doing a shared key and path key discovery and then clusters are formed. Key distribution is done as described in the paper by L. Eschenauer and V. D. Gligor [9]. This Key distribution technique relies on a probabilistic key sharing among nodes of the network. This essentially consists of 5 parts

- Generation of a large pool of P keys and of their key identifiers.
- Random drawing of k keys out of P to establish the key ring of the sensor.
- Loading of the key ring into the memory of each sensor.
- Saving of key identifiers of a key ring and associated sensor identifier on a trusted controller.
- For each node loading the i-th controller node with the key shared with that node.

The Key Pre-Distribution phase ensures that only a small number of keys need to be placed on each sensor node's key ring to ensure that network is connected with chosen probability. Now at the time of network initialization shared key discovery takes place, which is followed by path key discovery. This consists of two phases, Shared key discovery and path key discovery.

4.1.1 Key Pre-Distribution

- Shared Key Discovery:

The shared key discovery phase takes place during network initialization where every node discovers its neighbors in the wireless environment with which it shares keys. In this phase all the nodes broadcast a list of key identifiers of the keys on their key ring so that neighboring nodes can discover if they share a key with them. A link exists between two nodes only if they share a key. If a link exists between two nodes all the communication on that link is secured by link encryption using the shared key.

- Path Key Discovery:

After shared key discovery, there is a path key discovery phase that assigns a path key to the selected pair of nodes in the wireless communication range. In this way two distant nodes can communicate via a third node**.** For e.g. if node A shares a key with C and C shares a key with a node say X, then if A wants to communicate with X it can do it via C. As C is an authenticated node so a path key is established between A and X using C and then a link is formed between A and C which is encrypted using that path key. Any further communication between A and X is carried using this new link. The distribution of keys on the ring of sensor nodes is such that it provides for some extra keys, which can be used for path key. Thus multi-hop communication is only required at the time of network initialization to set up path keys. To establish the network shared key connectivity we need to answer two questions:

  - What value should the expected degree of a node, d have so that DSN of n nodes is connected and

- Given d and the neighborhood connectivity constraints imposed by wireless communication range, what value should the key ring size k and pool P have for the network of size n?

Let p be the probability that a shared key exists between two sensor nodes, n be the number of network nodes, and d = p *(n-1) be the expected degree of a node (i.e. the average number of edges connecting that node with its graph neighbors). For above graph G (n,p)

$p_c = \lim_{(n->infi)} pr[G(n,p) \text{ is connected}] = (e^e)^{-c}$ (4.1)

Where,

p = ln(n)/n + c/n and c is any real constant (4.2)

p_c is the given desired probability for the graph connectivity

Therefore given n we can find p and d = p *(n-1). (4.3)

Once we obtain the degree i.e. d we can determine k (number of keys on the key ring of a sensor node) by knowing the size of the memory of each sensor node and then we can find P that is size of the pool (from which to draw the keys). Also wireless connectivity constraints limit the number of neighbors to n' << n, hence the probability of sharing key between two nodes will be p'>>p. Now to determine P (size of the pool from which keys will be drawn) we can calculate p' = 1 – Pr [two nodes do not share a key]

And thus,

$p' = \frac{1 - ((P-k)!)^2}{(P-2k)!\,P!}$ (4.4) [9] $p' = 1 - \frac{(1-k/P)^{2(P-k+1/2)}}{(1-2k/P)^{(P-2k+1/2)}}$ (4.5)	pr(two nodes do not share a key) can be derived as: From a pool of P keys each key ring is drawn without replacement so the number of possible keys is: ${}^{P}P_k = P! / (k!\,(P-k)!)$. ${}^{P}P_k$ is the permutation of picking k keys at a time from P keys. Now after picking first key ring if we pick rest of the key rings from remaining (P-k) unused keys then these will not share any keys with the first key ring. Number of such key rings is: ${}^{(P-k)}P_k = (P-k)! / (k!(P-2k)!)$. Thus Pr(no key is shared between two key rings) = $\frac{k!((P-k)!)^2}{P!k!(P-2k)!}$ (4.6)

Table 1: Derivation of probability equation for two nodes sharing at least one key.

Using the above equation we can answer the second question above and can determine the size P of the pool from which to draw the keys so that shared key connectivity of the graph is established. The probability p' above is calculated for the network as a whole i.e. before any clusters are formed. Even if a node moves from one cluster to another, the distribution of the key ring on the node remains the same and the node can then do the shared key and path key discovery within its neighborhood to find a path to the head node. Because the probability that the other nodes share a key with the head node is p', as before, and the network is dense since there are many neighbors, the new entering node will share a key with some of the nodes.

4.1.2 Selection of Cluster Heads and Formation of cluster

All the nodes are assigned unique ids. Initially the cluster heads are selected based on the energy level and degree (number of neighbors that share a key) of the node. Each node in the network sends beacons to its neighboring nodes. The cluster head selection can be summarized as:

1) The beacon contains the node id, energy level of the node and status of the node i.e. head node or member node.
2) Upon receiving the beacon each node compares its energy level with the energy level in the beacon. If the energy level of the receiving node is less then or equal to the energy level in the beacon then it will change its status to member and send beacon to the sender node to inform the same.
3) If the energy level of the receiving node is greater than the energy level in the beacon then it considers itself as the head and send beacon to the sender node informing the same. Before adding the node to its cluster it makes sure that its degree (number of neighbors with which it shares keys) is less than the threshold limit. If the node has already reached its threshold then, the incoming node is assigned to the node, which has second highest energy level in the cluster, and that node will then form its own cluster with itself being the head. When the node is assigned to the new cluster head, first a path key is setup between that node and new cluster head via old cluster head. Now a link

is formed between this node and new cluster head, which is encrypted by the path key.

Once all the head nodes have been selected then the clusters are formed. Each head node will send a beacon to its physical neighbor nodes periodically. The nodes within the range of the head node that receive beacons respond by sending beacon back to the head node. The nodes can be in any of the four states:

(1) Transmitting (2) Receiving (3) Sleep (4) Idle.

Hence when a node is neither transmitting nor receiving it is in sleep state. In this state it listens and whenever it detects a message sent for it, it enters the receiving state and accepts the message. Hence, in this manner the radio of the nodes is kept in a minimum energy consumption state all the time. The nodes only respond to their neighbors when they hear something from them and hence no energy is wasted. On receiving the beacon from any node, the head node considers that node as its member if it shares a key with that node. This process is carried on a network wide scale until each node is either member of some cluster or is the cluster head itself. Each member node knows its cluster id. The cluster id is the id of the head node of the cluster. Some nodes are border nodes; i.e. they are in range of more than one cluster head. Such nodes belong to all of those clusters. Often there is more than one border node between two adjacent clusters so there may be multiple paths between two cluster heads.

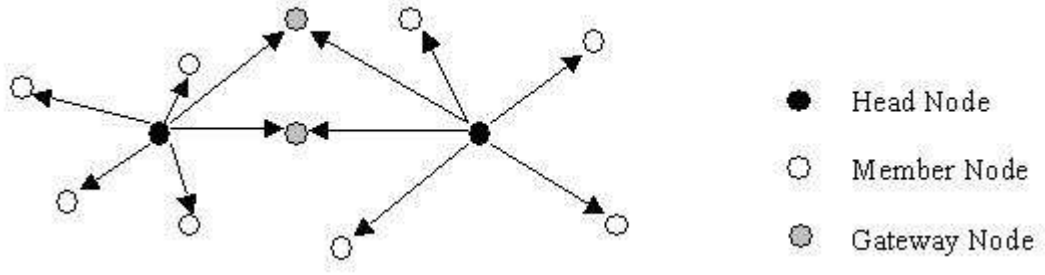

Figure 6. Cluster head sending beacons to the nodes within its range

After the cluster formation the cluster head is responsible for all the communication outside the cluster. Nodes inside the cluster communicate only with the cluster head so the energy of the head node is depleted faster than any other node in the cluster. Therefore we switch the cluster head depending on its energy level. Each node has different energy levels and depending on the number of messages sent or received the energy level of node keeps decreasing. Once the head node goes below a threshold level of energy it communicates with other nodes in its cluster to find out their energy levels. The node with the maximum energy level and having its degree less than the allowable maximum degree limit of the node is given the responsibility of the new head node. In case the energy level of all the nodes is the same then the existing head node continues as the head node. Whenever the cluster head is switched then all the information is transferred from the previous head to the new head. If the old head is incapable of doing so then the base station takes the charge and transfers all the information to the new head thus ensuring the consistency of all the communications.

Now this new head node sends beacons to other nodes in its vicinity to form its own cluster. If a new node enters the region of this cluster then it also becomes a part of this cluster, if it has a shared key with the head node. If it does not have a shared key with the head node then a path discovery will be made among the cluster nodes to establish a path key for communication between new node and head node. Hence, nodes do not become isolated as a result of the change in clusters. This is the other occasion when sensor nodes communicate among themselves, the first being network initialization. Thus the key distribution and cluster formation scheme makes sure that the mobility and cluster head sharing both are taken into account for increasing the life time and security of the network.

4.2 Algorithm for unbalanced clustering

A WSN of n nodes is pre-deployed with the key rings drawn randomly from a pool of P keys as described above and the shared key and path key discovery is done to setup links. The clustering done here does not take into account the degree of the cluster head and so clusters formed are unbalanced.

Unbalanced Clustering Algorithm

// $Node(id)_i$ means a node whose id is i
//$Node(En)_i$ means Energy of node whose id is i
//$Node(status)_i$ means status of a nodes whose id is i
//$(cluster\ id)_i$Head refers to the head node of cluster i
//$(cluster\ id)_i$ Head(En) means energy of head node of cluster i
//$((cluster\ id)_i(Node(id)_j\ ))$ means a node whose id is j and whose cluster id is i
//ThresholdEnergy is the minimum energy that a node needs to be cluster head
//BorderNodes are the nodes, which belong to more than one cluster

1) $Node(id)_i$ = rand() {for all i = 1….n}All the nodes are assigned unique node ids.

2) //Each node is assigned energy and status initially

$Node(En)_i$ = E and $Node\ (status)_i$ = undecided{where E is an integer and represents energy level of node and status = undecided or member or head}

3) $Node\ (id)_i$ sends beacon to neighbor(i) {for all i= 1…….k, k<n}
beacon = (id, status, En)

4) //Energy level of each and every node is compared to select cluster heads
//Each node contains a table, which has entry for every other node in its cluster.

if $Node(En)_i > Node(En)_j$ {for all j != i and j = neighbor(i)}
then $Node(id)_i$ = Head, cluster id = i;
Head sends beacon to neighbor (Head)
$Node\ (status)_k$ = member {for all k such that k = neighbor(Head)}

//Tables for node k contains ids of all the nodes, which are its neighbors i.e.,
//belong to the same cluster as k

$Node(id)_k$ has a table for $Node(id)_j$ {such that j = neighbor(k), cluster $id(Node(id)_k) = i$ for all k belongs to cluster id = i}
Node(Head) has a table for $Node(id)_j$ {such that cluster $id(Node(id)_j) = i$}

SelfOrganizeCluster() //Nodes move to make cluster head at center

else if $Node(En)_i = Node(En)_j$ {for any j such that j = neighbor(i)}then
{
if $Node(id)_i <= Node(id)_j$ then
{
$Node(id)_i$ = Head, cluster id = i
Head sends beacon to neighbor(Head)
$Node(status)_k$ = member {for all k such that k = neighbor(Head)}
$Node(id)_k$ has a table for $Node(id)_m$ {such that m = neighbor(k), cluster $id(Node(id)_k) = i$ for all k belongs to cluster id = i}
Node(Head) has a table for $Node(id)_m$ {such that cluster $id(Node(id)_m) = i$}

else Node(id)$_j$ = Head cluster id = j and above process is repeated for this Head as well.

}

SelfOrganizeCluster() //Nodes move to make cluster head at center

}

else Node(id)$_j$ = Head, cluster id = j; and above process of selecting members is repeated for this Head

5) //Energy losses due to communications

if ((cluster id)$_i$(Node(id)$_j$)) sends data to (cluster id)$_i$Head then
(cluster id)$_i$ Node(En)$_j$ - = V {where V is any positive integer}
(cluster id)$_i$ Head(En) - = W{where W is any positive integer < V}
else if (cluster id)i Head sends data to ((cluster id)$_i$(Node(id)$_j$)) then
(cluster id)$_i$ Head(En) - = V
(cluster id)$_i$ Node(En)$_j$ - = W {for all i such that i = cluster id and all j such that j = member of cluster with id i}

6) //Changing of existing cluster head if its energy is below ThresholdEnergy

if (cluster id)$_i$ Head(En) < ThresholdEnergy then
compare energy level of all nodes within the cluster
New Head = (cluster id)$_i$ Node(En)$_j$ {where j != i and energy level of j is > energy level of all the nodes inside the cluster}
cluster id = New Head(id)

7) Two cluster heads communicate via BorderNodes

//Cluster heads broadcasts requests to all of its members and all members know about //their neighbors which are in the same cluster. Therefore border node forwards //requests to other cluster head if the destination node is not in the same cluster.

SelfOrganizeCluster ()
//This is the definition of the function call, invoked in step 4 (in if and else blocks).
// This function organizes cluster such that cluster head lies in the approx center and
// all other member nodes are around it. We move cluster head by distance D, which
// is the mean of distance of all of its members, in the direction Theta, which is mean
// of angle which of all of its members form with head.
//Theta is calculated by finding relative position of cluster head and its members
// (whether member is in 1st 2nd 3rd or 4th quadrant //relative to the head node).

{

For each node i such that i is cluster head
do
for each node j such that j belongs to cluster of i
do
Theta = Theta + $\frac{Y_j - Y_i}{Xj - Xi}$ Sign of Theta depending on relative position of i and j
D = D + P_i - P_j //Find distance between i and j
od
end for
MeanDistance = FindMean of D
Mean Theta = Find Mean of Theta
Move the node i by angle Theta from original position by MeanDistance
End For
}

Repeat above step 3-6 6 till every node in the network is a member or a cluster head

4.3 Algorithm for balanced clustering solution

A WSN of n nodes is pre-deployed with the key rings drawn randomly form a pool of P keys as described above and shared key discovery and path key discovery is done to setup the links. This algorithm takes into account the energy level as well as degree of cluster head.

Balanced Clustering Algorithm

// $Node(id)_i$ means a node whose id is i
//$Node(En)_i$ means Energy of node whose id is i
//$Node(status)_i$ means status of a nodes whose id is i
//$(cluster\ id)_i$Head refers to the head node of cluster i
//$(cluster\ id)_i$ Head(En) means energy of head node of cluster i
//$(Node(degree)_x$ means degree of node whose id is x
//$((cluster\ id)_i(Node(id)_j$)) means a node whose id is j and whose cluster id is i
//ThresholdDegree is the max number of nodes that a cluster head can have in its cluster

//ThresholdEnergy is the minimum energy that a node needs to be cluster head
// BorderNodes are the nodes, which belong to more than one cluster

1) $Node(id)_i$ = rand() {for all i = 1….n}All the nodes are assigned unique node ids.

2) $Node(En)_i$ = E and $Node\ (status)_i$ = undecided{where E is an integer and represents energy level of node and status = undecided or member or head}

3) $Node\ (id)_i$ sends beacon to neighbor(i) {for all i= 1…….k, k<n}
beacon = (id, status, En)
//Energy level of each node is compared to other nodes to select the cluster heads

4) if $Node(En)_i$ > $Node(En)_j$ AND Node(j) shares key with Node(i) And $Node(status)_j$!= Head {for all j != i and j = neighbor(i)}then
{
if($Node(degree)_i$ < ThresholdDegree)
{
$Node(status)_i$ = Head, cluster id = i;
Node(status)j = member
Cluster id(Node(id)j) = i

}
else
{
While(($Node(degree)_x$ >= ThresholdDegree))
x = FindMaxEnergy(neighbor(i)) such that x shares key with i
End While

Node(status)x = Head
Node(status)j = member
Cluster id(Node(id)j) = x
}
SelfOrganizeCluster() //Nodes move to make cluster head at center

else if{
($Node(En)_i$ <= $Node(En)_j$) AND (Node(j) shares key with Node(i)) And ($Node(status)_i$!= Head)
then if{
$Node(degree)_j$ <= ThresholdDegree Then
$Node(status)_j$ = Head, cluster id = j
$Node(status)_i$ = member

Cluster id(Node(id)$_i$) = j
}
else{
While(Node(degree)$_x$ >= ThresholdDegree)

x = FindMaxEnergy(neighbor(i)) such that x shares key with i
End While

Node(status)x = Head
Node(status)i = member
Cluster id(Node(id)i) = x
}

SelfOrganizeCluster() //Nodes move to make cluster head at center
}
//Energy losses due to communications
5) if ((cluster id)$_i$(Node(id)$_j$)) sends data to (cluster id)$_i$Head then
(cluster id)$_i$ Node(En)$_j$ - = V {where V is any positive integer}
(cluster id)$_i$ Head(En) - = W{where W is any positive integer < V}
else if (cluster id)i Head sends data to ((cluster id)$_i$(Node(id)$_j$)) then
(cluster id)$_i$ Head(En) - = V
(cluster id)$_i$ Node(En)$_j$ - = W {for all i such that i = cluster id and all
j such that j = member of cluster with id i}

6) //Changing of existing cluster head if its energy is below ThresholdEnergy
if (cluster id)$_i$ Head(En) < ThresholdEnergy then
compare energy level of all nodes within the cluster
New Head = (cluster id)$_i$ Node(En)$_j$ {where j != i and energy level of j is > energy
level of all the nodes inside the cluster}
cluster id = New Head(id)

SelfOrganizeCluster() //Nodes move to make cluster head at center

7) Two cluster heads communicate via BorderNodes

8) Apply Force on the network (Each node one at a time) to move the nodes to improve the coverage. Force on each node i by node j is given by

$$F^{i,j}_n = \frac{D^i_n}{(Mu)^2} (cR - |p^i_n - p^j_n|) \frac{p^j_n - p^i_n}{|p^j_n - p^i_n|} \quad [14]$$

$F_i = \sum F^{i,j}_n$ where j = physical neighbor of i And F_i is total force on node i

EnergyConsumed of node i = k * F_i * D_i where

D_i = distance moved by node i on application of F_i
K = proportionality constant

F_i and EnergyConsumed is calculated in this way for each node in the network

```
SelfOrganizeCluster ()
  {

    For each node i such that i is cluster head
    do
       for each node j such that j belongs to cluster of i
        do
           Theta = Theta + (Yj – Yi)/(Xj - Xi) Sign of Theta depending on relative position of i and j
            D = D + Pi - Pj    //Find distance between i and j
         od
       end for
           MeanDistance =  FindMean of D
           Mean Theta = Find Mean of Theta
           Move the node i in the direction given by Theta from original position by
           MeanDistance
     End For
  }
```

Repeat above step 3-6 till every node in the network is a member or a cluster head

4.3.1. Analysis of the Algorithm

In this section we analyze the properties of the above proposed algorithm (Balanced Clustering)

Lemma1: The Algorithm will take constant number of iterations to terminate.

Proof: Step number 3 of the algorithm shows that every node sends beacons to its physical neighbors. Next step 4 checks that whether a node shares a key with its physical neighbor and if it does then decides about the head node based on their energy level. The Number of key shared neighbors of a node is far less than the number of physical neighbors. Hence the algorithm will terminate in finite iterations. If the total number of

nodes are N and each node has on an average K physical neighbors then the algorithm will terminate in N*K iterations

Lemma 2: The worst case processing time complexity at each node is constant.

Proof: Each node has to process messages from its physical neighbors. Let total number of nodes in the network be N and let K be the number of physical neighbors on average for each node. Here N>>K. So each node will take O(K) time to process messages from its neighbors because processing time per message is constant. Next a head has to decide about membership of a node depending on whether it shares a key with it; this computation takes an additional time say t. K being very small compared to N will result in small time complexity i.e. O(K) + t.

Lemma 3: The probability that two nodes within each other's cluster range are both cluster heads is zero i.e. cluster heads are well distributed.

Proof: Step 4 of the algorithm ensures that a node will only be selected a cluster head if it not one already. The algorithm thus iterates through all the nodes in the network and if a node is already a cluster head then it will not belong to any other cluster. a cluster in this case contains only those nodes that share keys as well as are within communication range of each other. If condition in step 4 is false for any node because it is head, then it cannot belong to any other cluster. In other words, suppose two cluster heads come into direct contact with each other. As they exchange their energy levels and/or their degrees, one of them will win out and become a cluster head and the other one will become a member. Hence no two cluster heads, can be in direct communication range within each other

Lemma 4: Cluster heads of two adjacent clusters communicate with each other via a border node.

Proof: The way in which two cluster heads communicate is via border nodes. Border nodes are the nodes that belong to more than one cluster. As shown in Lemma 3 a head node can only communicate with member nodes. Now while checking condition for each node in step 4 of above algorithm, if a node is already a member of some cluster, it still can be a member of other cluster if it satisfies the other conditions. Now as these nodes belong to more than one cluster that means that they share keys with more than one cluster head and so can communicate with them. So now two adjacent cluster heads will communicate via these nodes.

Lemma 5: Time complexity of the algorithm is constant

Proof: The algorithm starts by assuming that the security scheme is already deployed in the nodes. While doing a shared key discovery, nodes store a list of their physical neighbors as well as shared key neighbors. Now each and every node sends and receives beacons from its neighbors. This is done in parallel. The time required to send beacon by a node is constant and there are say N nodes in the network so total time required is O(1). Next nodes exchange messages with their neighbors to transfer data. This takes a constant time, which is very small. Adding both makes time complexity as O(1) + C, where C is a small constant.

Lemma 6: Worst case message exchange complexity per node is O(1)

Proof: Each cluster head generates a constant number of messages. A node does not respond to a message until it receives one. At any given instance there will be more than

one cluster heads and so remaining all other nodes will respond to beacons send by them. Now a cluster head sends only constant number of beacons to its surroundings and hence message exchange complexity per node is O(1).

4.4 Performance analysis of the proposed scheme

The proposed scheme for providing security as well as clustering is very energy efficient and results in improvement in more than one aspect of WSNs. The importance of this scheme is that it provides security to clustered WSNs. Any security mechanism for sensor networks use a lot of resources for computations and communications and this leads to decrease in battery power of sensor nodes and thus the life time of the network as a whole reduces. To avoid this first of all the security mechanism must involve as fewer computations as possible. The security scheme described above involves minimal computations but it involves communications at the time of network initialization to establish shared and path keys. Later clusters are formed in the network to make the network more energy efficient. Some of the parameters, to be considered while analyzing the performance of clustered networks, are:

1) Uniformity
2) Time for deployment
3) Effect of force between nodes, on the network:

1) Uniformity:

Uniformly distributed nodes spend energy more evenly through WSN than sensor nodes with an irregular topology. When the distance between sensor nodes become similar then less transmission power is required to transmit data to any node. In our proposed scheme when the clusters are formed then the network as a whole becomes structured. The cluster head lies in the center of the cluster with the member nodes placed all around it. The algorithm makes sure that clusters are self-organizing. This means that after formation of cluster, nodes rearrange themselves so that the head lies in the center of the cluster. This makes the topology even. Moreover nodes only need to communicate with the cluster head and cluster head is nearly at the same distance from all the member nodes. As we know the transmission power required is directly proportional to the square or higher power of the distance between the nodes [8] so uniformity can be defined as the average local standard deviation of the distances between the nodes. In [14] uniformity is defined as

$$U = \frac{1}{N} \sum U_i \text{ and} \quad (4.7)$$

$$Ui = \left(\frac{1}{K_i} \sum (Di,j - Mi)^2\right)^{1/2} \quad j = 1 \ldots K_i \quad (4.8)$$

Where, N is the total number of nodes

Ki is the number of neighbors of ith node

Di,j is the distance between the ith and the jth nodes,

U is the uniformity; inverse of U gives measure of uniformity of network.

The smaller the value of U, the more uniform the network.

Mi is the mean of inter nodal distances between the ith node and its neighbors.

In determining the value of Ui, that is, uniformity at the ith node, only nodes, which reside within the communication range of the ith node, will be considered. In the

proposed scheme a cluster contains only the nodes lying in neighborhood of a node so this equation can be applied to each cluster to obtain uniformity and then all the uniformity values can be added and divided by N to obtain uniformity for the whole network. A smaller value of Ui means that nodes are more uniformly distributed in the network. In a cluster nodes are more or less organized in a structured way with the cluster head in the center and thus the value for Mi (inter nodal distances) will be the same for all nodes resulting in a low value of Ui. In contrast, in a flat topology such as Directed diffusion [2] the nodes are randomly spread all over. A node can therefore sometimes communicate with a distant node and can sometimes communicate with near node thus resulting in unequal consumption of energy. This causes some nodes to die earlier than others resulting in the average lifetime of the network to be reduced.

2) Time for deployment:

This is one of the main criteria in measuring performance of any network. This includes time elapsed till nodes reach their stable state, that is, it is the total time required for the network to stabilize. The network is in a stable state when nodes have discovered their cluster membership and are ready to sense and communicate data. In our scheme, the nodes do a shared key discovery followed by a path key discovery to find all the neighbors. All the nodes broadcast at the same time. If we assume that the total number of nodes in the network is N then the time spent for broadcast from N nodes will be O(1). Nodes come to know about their neighbors only after they see the packets sent by them. Therefore shared key discovery and path key discovery takes the time O(1). Once the key discovery phase is over then the clusters heads are selected. This will take time O(1) because each node will again send beacons to its neighboring nodes to inform about its

status, id and energy level. Once the cluster heads are selected then the clusters will be formed which will take time O(1). Clusters are self-organizing and they arrange themselves in such a way that cluster head lies in the center and all the members surround it. This rearrangement takes a small time as nodes move some distance to accomplish this. So overall this process will take time O(1) + O(1) + O(1) which is constant time. This constant time is a little bit more as compared to the time required for flat level topology networks to stabilize, as flat level networks do not need to send beacons to form clusters and decide cluster heads. However, once the network is stabilized then the clusters result in saving of communication energy in the network.

3) Effect of force between nodes on the network:

Nodes move in a sensor network in order to increase the coverage. They normally move in a direction where node density is less so that network can overall sense more area. The distance traveled by each node in the mobile WSN will depend on the distribution of the nodes in the network. Distance traveled by a node is also related to the energy required by that node to travel that distance. The less the distance traveled by a node, the less the energy consumed. Each node is assumed to have some GPS system attached to it, which helps a node to know its location. We consider the concept of force to define the movement of nodes. The concept of a force is first reported in [14]. The force on a node by another node is dependent on the distance of that node from this node. Greater is the distance lesser is the force applied. We define force $F^{i,j}_n$ as force on ith node by jth node at time step n to be

$$F^{i,j}_n = \frac{D^i_n}{(Mu)^2} (cR - |p^i_n - p^j_n|) \frac{p^j_n - p^i_n}{|p^j_n - p^i_n|} \quad [14] \tag{4.9}$$

Where, cR stands for communication range

p^i_n stands for location of ith node at time step n

D^i_n stands for local density of ith node at time step n (equal to the number neighboring nodes of ith node at time n)

Mu stands for the expected density

Mu can be calculated by the formula $Mu = \frac{N*pi*(cR)^2}{A}$, N is total number of nodes, cR is the communication radius of each node and A is the area of the network.

The direction of the force can be calculated using vector addition. Due to force on a node by other node, it moves in a particular direction that can be calculated as:

Let F be the magnitude of the force, as calculated by above equation, on a node i by node j. First we find direction of node j relative to node i, i.e. whether node j is in first, second, third or fourth quadrant relative to i. That can be done as

Let the position of node i be xi, yi and of node j be xj, yi.

Theta be the angle between i and j,

Theta =

1) $\frac{Pi}{2}$ If xi – xj = 0, where pi = 3.142 (4.10)

2) $\frac{(yj - yi)}{(xj - xi)}$ if xj >= xi And yj >= yi (4.11)

3) $\frac{(yj - yi)}{-(xj - xi)}$ if xj <= xi And yj >= yi (4.12)

4) $-\frac{(yj - yi)}{-(xj - xi)}$ if xj <= xi And yj <= yi (4.13)

5) $-\frac{(yj - yi)}{(xj - xi)}$ if xj >= xi And yj <= yi (4.14)

Depending on the above condition we will obtain Theta and then calculate the force in the x direction and y direction using Fx = F cos Theta and Fy = F sin Theta

Where Fx and Fy are the components of force F in the x and y directions respectively.

We thus resolve each and every force acting on a node into x and y components. After this we will add all the x components and then all the y components. The resultant x and y component forces can be given as:

$$Fx(res) = Fx1 + Fx2 + \ldots \quad (4.15)$$

$$Fy(res) = Fy1 + Fy2 + \ldots \quad (4.16)$$

Total force on a node will be

$$F(res) = Sqrt[(Fx(res))^2 + (Fy(res))^2] \quad (4.17)$$

Due to this force each and every node moves and the network rearranges itself. The clustering algorithm is applied after this. The results for the performance of the network after this force is applied are shown in the results chapter. Results also show that the loss of energy in moving the nodes is comparable to the energy lost in the communication while nodes form clusters and do data transfer. Hence there is no large overhead in moving the nodes. The algorithm to achieve this is shown below.

Force Calculation Algorithm

Apply Force on the network (Each node one at a time) to move the nodes to improve the coverage. Force on each node i by node j is given by

$$F^{i,j}_n = \frac{D^i_n}{(Mu)^2} (cR - |p^i_n - p^j_n|) \frac{p^j_n - p^i_n}{|p^j_n - p^i_n|} \quad [14]$$

$F_i = \sum F^{i,j}_n$ where j = physical neighbor of i And F_i is total force on node i
Node i moves by application of force Fi

EnergyConsumed of node i = $k * F_i * D_i$ where
D_i = Final(P_i) – Initial (P_i)
D_i = distance moved by node i on application of F_i

K = proportionality constant

F_i and EnergyConsumed is calculated in this way for each node in the network

4.5 Communication and Computation Overheads

There is some overhead involved in the formation of clusters. We have defined four different states of node operation and a node always tries to be in the minimum power consumption state thus reducing the overall energy consumption. The energy spent in cluster formation is later compensated as the nodes inside clusters have to communicate shorter distances and only a head is responsible for all the communication inside a cluster. Moreover, the head is switched at regular interval so that no node runs out of energy faster then other nodes in the cluster. Oscillations that may arise due to constant switching of heads is reduced by keeping an oscillation count at the base station for each cluster head. As soon as this count reaches a maximum limit then that particular node is no longer assigned the responsibility of cluster head.

The proposed algorithm includes the following overheads. In the calculation of uniformity of the cluster there are extra calculations at each and every node. The Mi (mean of the inter nodal distances) for every node, the uniformity for every node, for cluster and network has to be calculated. But all the nodes already have the information about their neighbors (because during the determination of the shared key, the nodes identify the nodes within their communication range i.e. their physical neighbors). Hence there is no need to calculate Di,j (distances between a node and its neighbors) and therefore no communication is required. Whether a node lies in the communication range of other node is determined using signal strength. Similarly for determining the force that must be applied on the network there is an overhead on every node. This includes

determining the expected density (Mu) for each node. We already know the other parameters such as the position of nodes (we have assumed that each node has a GPS system) and local density (D^i_n), which is equal to the number of physical neighbors of a node. Time for deployment is the other parameter, which involves some communications overhead as, described above, but this can be ignored because the base station can carry the load for this overhead at the time of deployment Hence the overheads in the proposed approach are within the capabilities of sensor networks and the batteries lifetime would not be affected greatly.

CHAPTER V

RESULTS AND CONCLUSION

Aim: To simulate and compare the following schemes

1. Key ring scheme for security [9] which does not involve clustering
2. Key ring scheme for security [Sec 4.1] with clustering based on energy levels of a node.
3. Key ring scheme for security [Sec 4.2] with clustering based on energy levels and degree of a node in terms of the keys shared with neighboring nodes
4. Key ring scheme for security with force calculations to redistribute nodes followed by clustering based on energy levels and degree of a node in terms of the keys shared with neighboring nodes

Environment: VC++ .NET

Power Consideration:

In WSNs each sensor node has a limited battery power and hence a limited lifetime. The power of the nodes is mainly consumed in communication and computations. Since we are using a key pre-distribution scheme as the security scheme, minimum computations are involved. There are some communications at the time of network initialization for cluster formation but the energy lost in that is compensated later when nodes only have to communicate to the cluster head and hence loose less energy. The power required for transmission depends on the distance between the nodes; the greater the distance, the more the power required. Clustering reduces the distance

between the nodes because then nodes have to communicate only with their respective heads resulting in overall less power wasted.

Data Structures:

All the nodes have a neighbor table, which contains the list of their physical neighbors. The border nodes will contain more tables because they will have neighbors in more than one cluster. The values in the tables are updated at regular intervals based on the packets received from the neighbors. A linklist is used to maintain table(s) at each node. This allows entries to be added and deleted to the list as nodes move in or out of range of other nodes. In addition to the neighbor table there is another table called shared key neighbor table that contains entries for all the nodes with which a node shares at least one key.

Simulation Parameters:

Total Number of nodes in the network: 50

Size of the key pool from which to draw the keys: 1000

Size of the keyring on each node: 50

Area used for nodes boundaries: 1000 units

Broadcast range of the sensor nodes: 10 units

A network of 50 sensor nodes is formed and pre deployed with keys drawn randomly from a pool of 1000 keys. The size of the key pool is decided based on the

probability equations provided by Erdos and Renyi [9]. After deployment a shared key and a path key discovery is made to find out nodes with which other nodes share keys.

5.1 Graphs

1) Number of Communications Vs Ave remaining power of a node:

Simulations are carried out to find out average remaining power of a node versus the Number of communications in the network for all the four schemes. Number of communications is the random number of communications carried out between nodes for data transfer during one program run. The graph obtained is shown below:

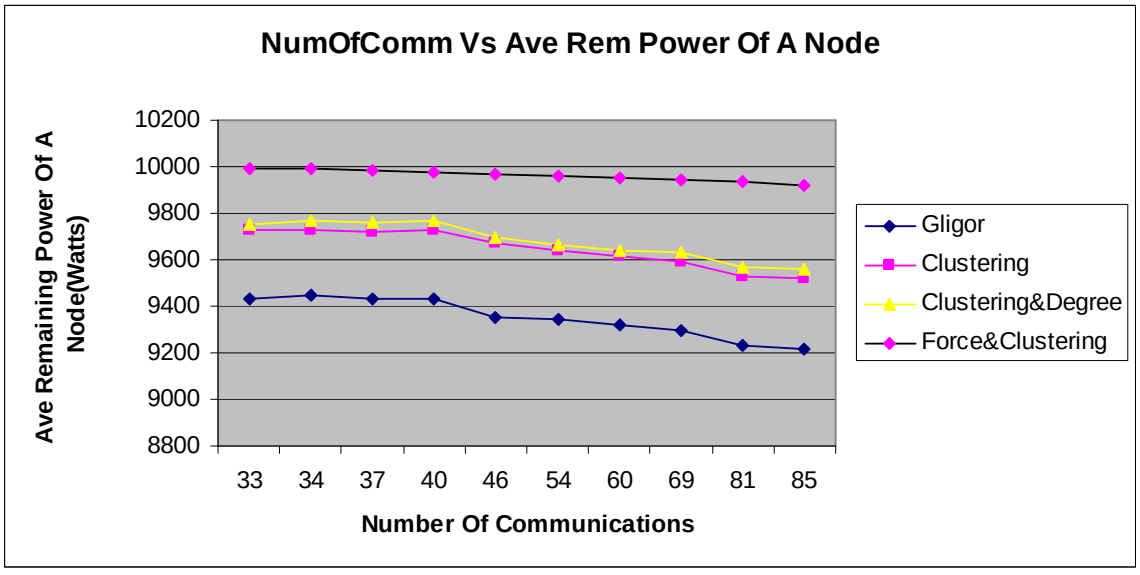

Figure 7. Graph for number of communications vs ave remaining power of a node

The graph shows the variation of average remaining power of a node in all the 4 schemes. It shows that for equal number of communications, ave remaining power of a

node is greatest in our proposed Force & Clustering scheme. First nodes are spread out evenly by applying force. Evenly here means that nodes distribute themselves in such a manner as to improve the coverage of the network. To accomplish this nodes move from region of high density to region of low density resulting in a more even spread of nodes and clusters. As expected, in the Clustering & Degree scheme, where there is no force applied but only balanced clusters are formed depending on the energy level and degree of a node, consumption of energy is more as compared to the first case. The reason for this is that as clusters are self-organizing so they form a structure such that the head lies in the center and all the members surround it. To form such a structure nodes move. These movements cause energy losses. In unbalanced clustering, energy consumption is even more due to the unequal distribution of nodes in clusters. The head nodes of clusters, which are heavily loaded, losses energy faster than the head nodes of lightly loaded clusters. This leads to unequal consumption of energy. The difference between energy consumption in this case and the previous one is small because balanced clusters are formed only when nodes start reaching their ThresholdDegree (Max nodes which a head can support). In the last case of Gligor's scheme there is no clustering involved so all nodes communicate over longer distances, especially to the base station directly. This results in drastic reduction in their energy levels as indicated in the graph.

2) Number of Communications Vs Number of Nodes Alive:

Simulations are carried out to obtain number of nodes alive in all the four schemes and the results are shown below:

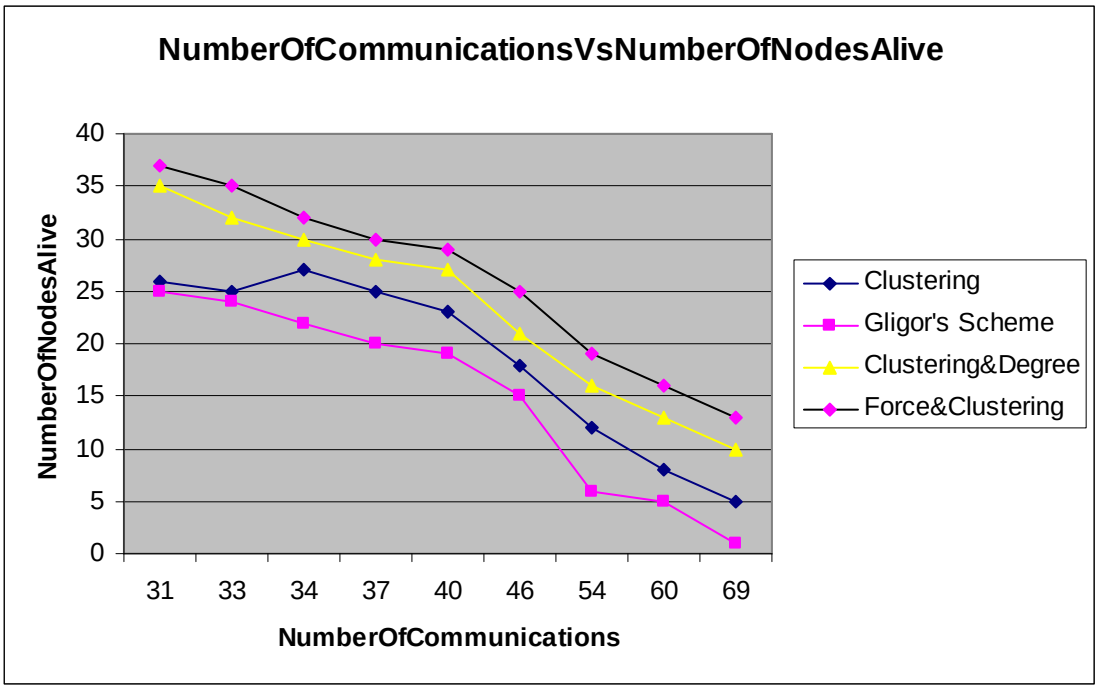

Figure 8. Graph for number of communications vs number of nodes alive

The graph above shows the number of nodes alive after a fixed number of communications has been carried out in the different schemes. What is interesting here is that Gligor's scheme and clustering scheme start off with same performance but gradually as the numbers of communications increase the performance of Gligor's scheme degrades more. This happens because when nodes communicate in a flat level topology for a long time, their energy decreases considerably and they start dying off. The communications between nodes are random. Initially there is not much difference between two schemes because nodes, which are communicating, are not far enough. So energy dissipated in clustered and flat level topology network is nearly same. . In the clustering schemes, nodes communicate with each other via other nodes and not directly so less energy is spend in communication. In the third scheme, which is clustering&degree the number of nodes alive is even more. This is because clusters are

balanced so no cluster head is over loaded and hence all heads dissipate their energies equally. The performance of the fourth scheme is the best of all the schemes because on applying force it provides a good distribution of nodes on which clustering is done later. As nodes are distributed evenly so they have to move less distance to form a cluster with head in center and nodes around it. So energy loss in forming clusters in this case is less and hence more numbers of nodes are alive. This shows that as we move from flat level to clustered scheme and then to a balanced clustered scheme that number of nodes dying due to energy depletion gets less and less.

3) Number of Nodes Vs Uniformity:

Uniformity gives a measure of distribution of the nodes in the network. If nodes are distributed evenly then it results in even distribution of energy in the network. Simulations are carried out to show that when clustering is done then nodes are more uniformly distributed in the network as compared to a network in which there is no clustering. The graph showing the comparisons of uniformity for both the schemes is:

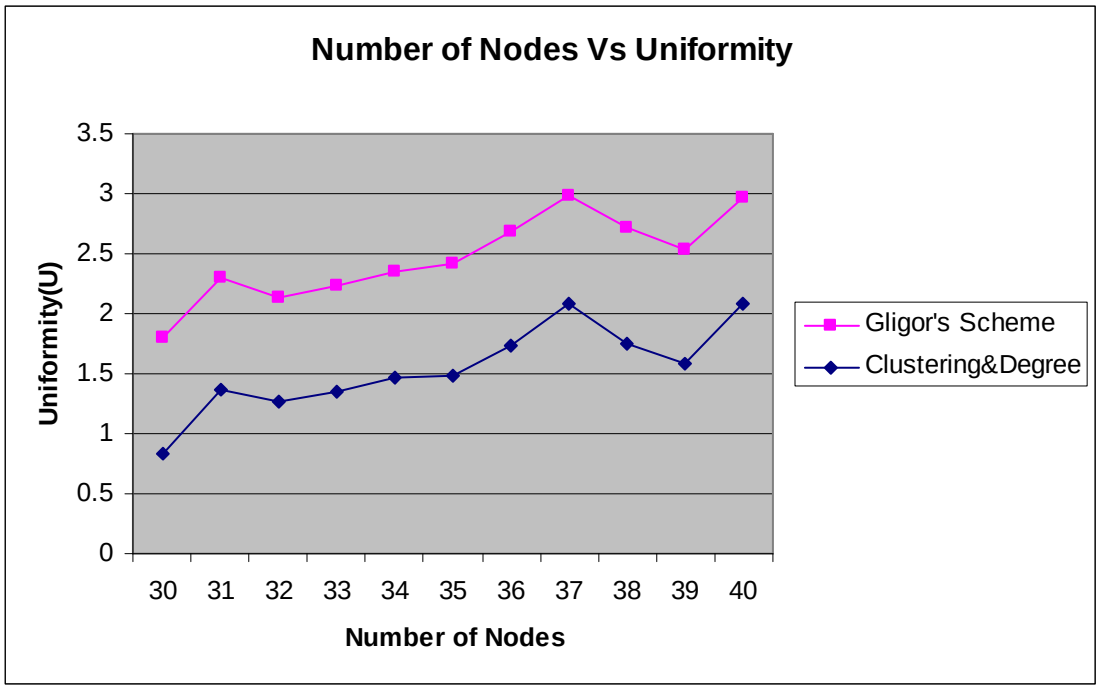

Figure 9. Graph for number of nodes vs uniformity

In the graph above Uniformity (U) is defined by the equations (4.7) and (4.8). Here lower value of U indicates more uniformity. The main factor in determining U is the means of inter nodal distances. A Cluster is formed such that the head is at the center and all other nodes surround it. Due to this distance between a node and its cluster head i.e. two communicating nodes in this case is more or less equal to the mean of inter nodal distance. So the difference of distance between a node and its cluster head, and mean of inter nodal distance, is much less i.e. (Di,j – Mi) [sec 4.4] and which in turn results in low value of U. In Gligor's Scheme, the network has a flat level topology so no such conclusions can be drawn about distance between communicating nodes. As shown in the graph the values of U are less for clustering&degree scheme than Gligor's scheme at all times.

4) Key ring size Vs Pr (At least one key is shared):

Simulations are carried out to show how the probability of sharing key between two nodes varies as size of key ring varies. The equations formulated by Erdos and Renyi [14] are used to find out the probability of key sharing between two nodes.

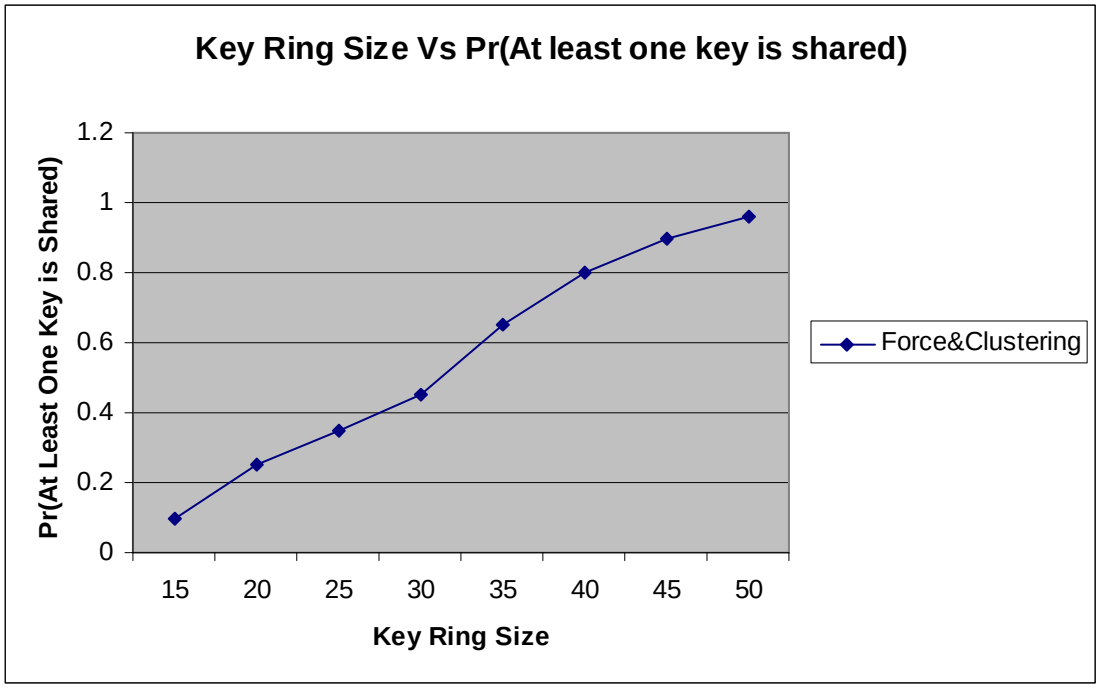

Figure 10. Graph for key ring size vs pr (at least one key is shared)

In this graph as the size of the key ring increases the probability of key sharing between two nodes also increases. This is predictable because the size of key pool from which we draw keys for storing them on the key ring of nodes is the same. The simulations show that if the size of key ring is made 50 then there is an almost 100% probability that two nodes will share at least one key if the keys are drawn from a key pool of size 1000 keys. In carrying out these simulations we have varied the size of the key pool and the size of the key ring to find out their effect on various other parameters.

5) Energy loss in communication Vs Energy loss in moving nodes due to force:

The comparison of energy loss during communication to energy loss during moving of nodes due to force is to determine whether it is worth moving the nodes using the force equations described in section 4.4 (3). If the difference between both energy losses is small then we can consider moving the nodes by applying force so that it can eventually result in less energy consumption. The reason for moving the nodes is to increase the coverage of the network. Along with increase in the coverage, energy consumption in future communications is also reduced after nodes have been moved and then clustered. The graph below refers to random number of communications for a given number of nodes. The energy consumed in communications is scaled according to values in [6]. The standard values for power consumption in the transceiver of sensor node are 24 mw (milli watts) for transmission and 14 mw (milli watts) for receiving. These values give the minimum power required to do one communication. Power is energy consumed per second. We are assuming that there is one communication per second. Therefore energy consumed per second will be in mJoules (milli Joules). Now the energy consumed in communication also depends on the distance between the sensor nodes. We multiply this distance by the minimum energy required as given above depending on whether a node is transmitting or receiving and thus get the final value of the energy lost in that communication. Hence, we calculate energy consumed for all the communication between various pairs of nodes and then sum that up to get total energy lost for certain number of communication for whole network. The equations used for calculating energy lost in communications are:

$$Ec = k * (p_i - p_j) * [(EL)_t + (EL)_r] \quad (5.1)$$

Where, i = 1 to T (Total number of nodes involved in transmission)

j = 1 to R (Total number of nodes involved in receiving of messages)

Ec = Total energy lost in communication

k = proportionality constant

p_i = position of node i

p_j = position of node j

$(EL)_t$ = Energy loss of transmitting node (mJoules)

$(EL)_r$ = Energy loss of receiving node (mJoules)

The energy lost in motion is calculated depending on the force applied. When nodes move by application of force as calculated by equation (4.9) they cover some distance. We multiply this distance by the amount of force to get total work done on the node and total work done is equal to the total energy spend. The equation used for calculation of energy lost in motion is:

$$Em = Fi * (p^i_f - p^i_s) \quad (5.2)$$

Where i = 1 to N (Total number of nodes in the network)

Fi = force on node i as given by equation (4.9) in Newton

p^i_f = final position of node i

p^i_s = initial position of node i

Unit of $(p^i_f - p^i_s)$ is meters

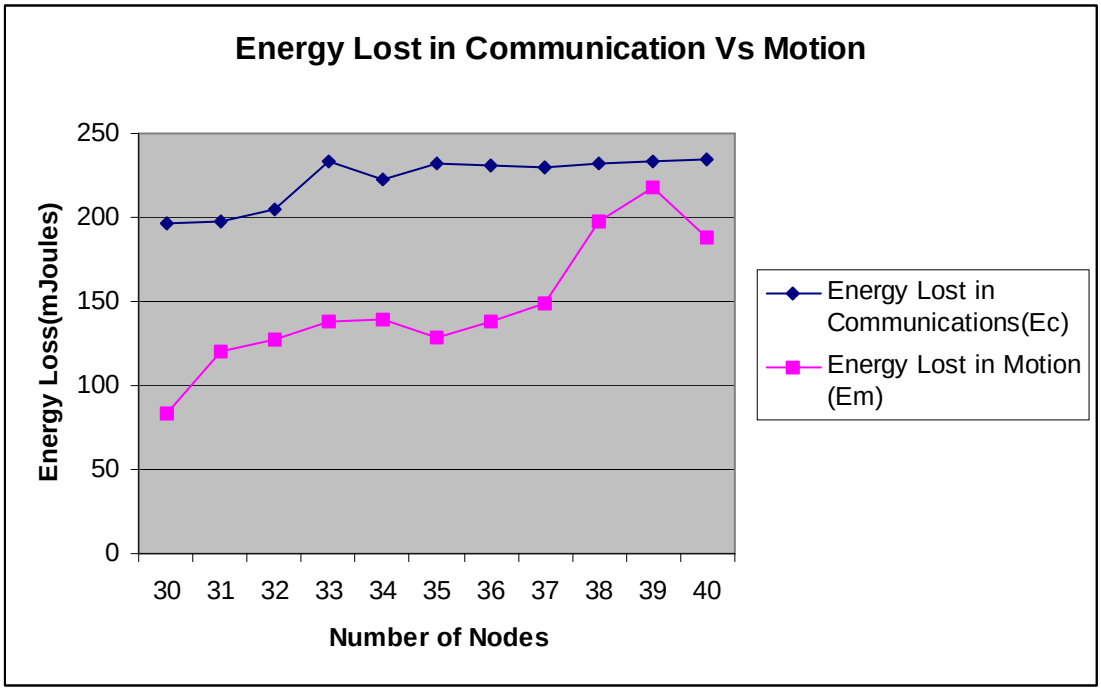

Figure 11. Graph for energy lost in communication vs energy lost in motion

The graph shows that difference between energy losses in both cases is relatively small and hence the idea of moving the nodes by applying force is viable. This is particularly true as the number of nodes in the network increases. Initially as there are few nodes in the network so resultant force on each node by other nodes is small and nodes thus move a small distance and hence energy consumed for motion is little. There are also fewer communications when the number of nodes is small but there are few clusters as well. So cluster heads have to communicate over long distances to send messages to each other. Due to this more energy is lost in the beginning. As the number of nodes increase, the network becomes dense and the nodes exert more force on each other and so more energy is lost in their motion. However, the energy spent in communications does not increase much because there are more clusters (as there are

more nodes in the network) resulting in more cluster heads. These cluster heads are closer to each other in a dense network that in a sparse network. As the number of nodes increase, the number of communications also increases because there are more nodes, which share keys with other nodes in the network, and hence they carry out trusted communications with them. Thus at the beginning there is a large gap. Hence in a sparse network it is worth applying force to move the nodes. However, in a dense network the advantage of applying force is less. Other factors such as the rate of communications and the energy spent in communications must be factored in when determining whether it is worth applying force.

5.2 Conclusion

In this thesis we have proposed a novel approach to energy efficient secure sensor networks. The random key pre-distribution scheme proposed by L Eschenauer and V.D. Gligor ensures that it works for any distributed wireless sensor network regardless of its topology. Furthermore, due to the probabilistic method used for key distribution the need for storing large number of keys on each node is greatly reduced. A relatively small number of keys ensure that there is a high probability of network connectivity. The proposed approach uses clustering to save energy. Moreover, by clustering there is no need for each and every node to transfer its data directly to the base station. This research shows that performance metrics such as uniformity are improved. Uniformity gives the measure of the energy dissipation in the network. A concept of force between the nodes is used to move nodes in such a way as to improve coverage. When nodes are moved due to the force and then clustering applied, it results in reduced energy consumption

providing more coverage. We also considered the overhead caused by applying force. Simulation results show that the overhead in moving the nodes due to force is comparable to the energy lost in communications. Although there is an initial expenditure of energy, in the long run this saves energy as far as we are aware no previous work has tried to combine energy efficiency and security for sensor networks.

This work can be further extended to reduce the computation overheads on the nodes. While we apply force and calculate the uniformity of the network there are some computations involved, which result in loss of energy of nodes. Though application of force to redistribute nodes provides advantages in long term but it would be beneficial if energy losses while calculating force can be lowered. Another area for further research is investigating the effect of large number of clusters on the network. As clustering results in formation of many clusters each having few members, it would be interesting to know what effect it would have on the performance of the network if we reduce the number of clusters and increase the number of nodes per cluster.

REFERENCES

1. Agrawal D. P. and Deng Hongmei, Wei Li, "Routing Security in Wireless Ad hoc Networks", *IEEE Communications Magazine*, Vol. 10, No. 2, pp. 70-75, October 2002.
2. Akkaya Kemal and Younis Mohammed, "A Survey on Routing Protocol for Wireless Sensor Networks", *Elsevier Ad Hoc Network Journal*, Vol 3/3 pp. 325-349, 2005
3. Bandyopadhyay Seema and Coyle Edward J. "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", *Proceedings IEEE INFOCOM,* 2003.
4. Chan H,. Perrig A, and. Song D, "Random key pre-distribution schemes for sensor networks", *Proceedings IEEE Symposium on Security and Privacy*, May 11-14 2003, pp. 197–213.
5. Chandrakasan Anantha, Heinzelman Wendi and Balakrishnan Hari, "Energy Efficient Communication protocols for Wireless Microsensor Networks", *Proc. Hawaii Int'l Conf. on System Science,* January 2000.
6. Chatterjea S, Van Hoesel L.F.W. and Havinga P.J.M., "An energy efficient medium access protocol for wireless sensor networks", *SenSys, Baltimore (USA), 11-2004*
7. Deng J, Du W.,. Han Y. S and. Varshney P K, "A pairwise key pre-distribution scheme for wireless sensor networks", *Proceedings 10th ACM Conference on Computer and Communications Security (CCS),* October 27-31, 2003, pp. 42–51.
8. Ghiasi Soheil, Srivastava Ankur, Yang Xiaojjan and Sarrafzadeh Majid, "Optimal Energy Aware Clustering in Sensor Networks", http://www.mdpi.net/sensors. Invited paper, *Sensor 2002*, 2, 258-269, 2002.

9. Gligor V.D., Eschenauer L, “A key-management scheme for distributed sensor networks”, *Proceedings of the 9th ACM conference on Computer and Communications Security*, Washington, DC, USA, November 18-22 2002, pp. 41–47.
10. Kahn J.M, Katz R.H and Pister K.S.J, ”Mobile Networking for Smart Dust”. *Proceedings ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM 99),* 1999, pp.271-278.
11. Karlof Chris and Wagner David, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”. *Elsevier Ad Hoc Networks Journal*, 1, 2003, pp. 293–315
12. Katz. R.H., Subramanian L, “An Architecture for Building Self-Configurable Systems”. *Proceedings of IEEE/ACM workshop on Mobile Ad hoc Networking and Computing,* August 2000.
13. Perrig A, Szewczyk R, Wen D. Culler V and Tyrag J, “SPINS Security Protocols for Sensor Networks”. *Proceedings of Mobile Networking and Computing* , 2001.
14. Varshney Pramod K and Heo Nojeong, “A Distributed Self-Spreading Algorithm for Mobile Wireless Sensor Networks”. *Proceedings of IEEE wireless communication and Networking Conference, WCNC*, 2003

VITA

Samir Gokhale

Candidate for the Degree of

Master of Science

Thesis: Secured Clustering in Wireless Sensor Networks

Major Field: Computer Science

Biographical:

Personal Data:

Born in Indore, India on Dec 11th, 1979, the son of Ravindra Gokhale and Rashmi Gokhale

Education:

Graduated from higher secondary school in India, received a bachelors degree from Institute of Engineering and Technology (DAVV) in India in May 2002 in the field of Computer Science and Engineering. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in May, 2005.

Experience:

Worked as a Research Assistant with the School of Geology at Oklahoma State University from Aug 2003 to Sep 2004.