HEURISTIC CLUSTERING WITH SECURED ROUTING

IN TWO-TIER SENSOR NETWORKS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Kanwalinderjit Kaur Gagneja

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

February 2013

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Heuristic Clustering with Secured Routing in Two-Tier Sensor Networks

By

Kanwalinderjit Kaur Gagneja

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Kendall E. Nygard
Chair

Dr. Brian Slator

Dr. Jun Kong

Dr. Joseph Szmerekovsky

Approved:

04/08/2013
Date

Dr. Kenneth Magel
Department Chair

ABSTRACT

This study addresses the management of Heterogeneous Sensor Networks (HSNs) in an area of interest. The use of sensors in our day-to-day life has increased dramatically, and in 10-15 years, sensor nodes may cover the entire world and could be accessed through the Internet.

The sensors have limited resources, and researchers have invented methods to deal with the related issues. Security and routing for sensor networks and sensor clustering have been handled separately by past researchers. Because route selection depends on the nodes' position and resource sets may change dynamically, cumulative and coordinated activities are essential to maintain the organizational structure of laid-out sensors. To conserve the sensor network's energy, it is better if we follow a holistic approach, taking care of both clustering and secure routing.

In the given heterogeneous sensor networks, the low-end nodes are clustered and report to a high-end node which, in turn, uses a network backbone to route data to a base station. Initially, we partition the given area into Voronoi clusters. Voronoi diagrams generate polygonal clusters using Euclidian distance. Because sensor network routing is multi-hopped, we apply a Tabu search to adjust some nodes in the Voronoi clusters. The Voronoi clusters then work with hop counts instead of distance. When some event occurs in the network, low-end nodes gather and forward data to cluster heads using the Secure Improved Tree Routing approach. The routing among the low-end nodes, high-end nodes, and the base station is made secure and efficient by applying a two-way handshaking secure Improved Tree Routing (ITR) technique. The secure ITR data-routing procedure improves the network's energy efficiency by reducing the number of hops utilized to reach the base station. We gain robustness and energy efficiency by reducing the vulnerability points in the network by employing alternatives for the shortest-path tree routing. In

this way, a complete solution is provided for the traveling data in a two-tier heterogeneous sensor network by reducing the hop count, making it secure and energy efficient. Empirical evaluations show how the described algorithm performs with respect to the delivery ratio, end-to-end delays, and energy usage.

ACKNOWLEDGEMENTS

First of all, I would like to thank my adviser, Dr. Kendall E. Nygard. His valuable and expert guidance enabled me to successfully complete this research and write the dissertation. In spite of his other engagements and preoccupations, he devoted time to read the successive drafts and make valuable suggestions for improvement. He took a keen interest throughout the dissertation's preparation and offered constant feedback, in most categorical terms, thereby, giving rewarding leads at every stage. It, indeed, has been a valuable experience for me to research under his supervision and guidance.

Next, I would like to thank my Ph.D. dissertation committee members, Dr. Brain Slator, Dr. Jun Kong, Dr. Joseph Szmerekovsky, and Dr. Wei Zhang, for taking the time to serve on my committee, reviewing my work, and providing valuable insight for the completion of my dissertation.

This research work is funded by the Designing Robust and Secure Heterogeneous Sensor Networks project. This project is sponsored by the Army Research Office (ARO). I would like to thank Dr. Nygard and Dr. Du, who provided funding for this project. My heartfelt thanks are due to my group mates and friends for their support and help.

I sincerely appreciate Carole Huber, Stephanie Sculthorp, and other staff members who have been greatly supportive during the process.

Finally, I salute the benevolence of my family members, who showed patience and understanding during this period and provided their cooperation, unfailing courtesy, and continuous inspiration. I shall be failing if I do not acknowledge that this effort would not have succeeded without their considerable support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1. Overview

The research work is focused on heterogeneous sensor networks. Such networks have two types of nodes; the low-end nodes are simple and low cost while the high-end nodes are costly but provide significantly more storage and processing power. In this type of sensor network, the low-end nodes are clustered and report to a high-end node which, in turn, uses a network backbone to route data to the base station. The low-end nodes are large in number compared to high-end nodes. The base station is assumed to be reliable and trustworthy, having essentially unlimited battery power, high computing speed, large memory, large bandwidth, and high-quality links.

In the literature on route specification for sensor networks, communication among nodes is limited to parent-child links which means that, when the network topology is complex, the shortest-path tree or logical-tree topology is used for data forwarding. However, when a node fails, the usual logical-tree topology is unable to forward data packets to their superior nodes. We assert that this network vulnerability could be reduced by employing alternatives for shortest-path tree routing. Thus, our first objective is to develop an improved routing methodology for two-tier sensor networks that forwards the data packets when nodes fail and also reduces the number of hop counts even under ideal conditions.

The second objective is to develop a key-management scheme which serves the purpose of securing the two-tier sensor networks. The key-management scheme should be embedded to the ITR (Improved Tree Routing) protocol so that there are fewer chances for data to be intercepted by adversaries while transmitting data to their cluster heads. However, Low Energy Adaptive Clustering Hierarchy (LEACH) [20], Sensor Protocols for Information via Negotiation

(SPIN) [35], and Directed Diffusion (DD) [21], [22] routing protocols already exist, but they do not support secure data transmission.

After the nodes are deployed in the given area of interest, if the low-end nodes form clusters around high-end nodes, then the network lifetime could be improved (due to limited battery life). Because clustering plays an essential role in a two-tier topology, forming high-performance clusters can considerably reduce the complexity of the network and can improve the network performance [29]. Hence, the third objective is to develop an algorithm to form such high-performance clusters for two-tier sensor networks.

The most prominent objective of the three is the third one: forming high-performance clusters. The first objective of Improved Tree Routing plays a key role in realizing the third objective. The NS-3 network simulator [12] is used to implement the methodology. The empirical evaluations are done to generate results for comparisons. Fundamentally, two agents are written: one for secure ITR routing, called the SecureITRAgent, and the second for partitioning the area of interest into Tabu-based Voronoi clusters, the TabuVoronoiClusteringAgent. The TabuVoronoiClusteringAgent generates clusters, and it is given as input to the SecureITRAgent to route the secured data into two-tier sensor networks.

1.2. Dissertation Outline

Chapter 2 describes sensor networks, their applications, technical problems that we encountered while working with them, their architecture, some functional issues, and the Tabu search background and its algorithm. The chapter also reviews the literature related to routing, clustering, and the security of sensor networks.

Chapter 3 discusses the dissertation papers, the published papers, the relationships among the papers, and the contributions for all the papers.

Chapter 4 has the complete paper titled “Enhanced Routing in Heterogeneous Sensor Networks” that was published in the IEEE Xplore and The Future Computing conference proceedings (pp. 569-574), held in Athens, Greece, on Nov. 15-20, 2009.

Chapter 5 has the paper “Key Management Scheme for Routing in Clustered Heterogeneous Sensor Networks” that was published in the proceedings of IEEE NTMS'2012 - Security Track (pp. 1-5), held in Istanbul, Turkey, on 7-10 May, 2012.

Chapter 6 has the paper “Tabu-Voronoi Clustering Heuristics with Key Management Scheme for Heterogeneous Sensor Networks” that was published in the proceedings of IEEE ICUFN 2012 (pp. 46-51), held in Phuket, Thailand, on July 4-6, 2012.

Chapter 7 offers the Conclusions and Future Work.

CHAPTER 2. BACKGROUND AND REVIEW OF LITERATURE

2.1. Overview

A wireless sensor network is a collection of small-sized sensing nodes deployed and structured in a cooperative network. These small nodes have processing capability, have a CPU chip, may have a memory chip, have a radio frequency transceiver, and have batteries and solar cells as a power source. They provide specific-type data obtained by their embedded sensors embedded. The nodes communicate among each other wirelessly. In some networks, nodes can even self-organize in an ad-hoc fashion immediately after deployment. A wireless sensor network can have thousands, or even tens of thousands, of nodes [40]. These networks are revolutionizing the way we do business, live in a house, workout, have medical treatment, and work.

Presently, the deployment of these sensor networks in our day-to-day life has increased, and in 10-15 years, the sensor nodes may cover the entire world and could be accessed through the Internet. Currently, wireless sensor networks are used for a number of different applications, such as vehicular movement tracking, nuclear power-plant monitoring, fire-incident reporting, traffic controlling, environment monitoring, etc. Wireless-sensor-network technology has a great potential for various applications, such as entertainment, drug trafficking, border surveillance, crisis management, underwater environment monitoring, and smart spaces.

Wireless sensor networks are not the same as wired or wireless networks, but are distributed real-time systems because wireless sensor networks share some similarities with these systems; therefore, some existing features could easily be used in these new systems. Unfortunately, the features are not enough, and little existing technology can be utilized. Therefore, innovative solutions are essential in a sensor-network area because existing

technology has features which use the set of suppositions or assumptions with significantly different features.

Research about distributed systems assumes that nodes communicate with each other through a wired/wireless medium, have infinite power resources, have user interfaces utilizing monitors and a mouse, are not real-time, and have a predetermined set of resources. Also, every node in the network is considered essential, and all nodes are location independent and autonomous.

On the contrary, wireless sensor networks work with radio frequency (i.e., they are wireless.), have limited power, are real-time, and utilize sensors and actuators for interfacing. Sets of resources may change dynamically, so cumulative and coordinated activities are essential, and organizational structure is vital. The sensor nodes' limited power supply puts additional burden on system performance and lets researchers think of better solutions.

Usually, a wireless sensor node has elements, such as sensing and computing capability, communicating sensed data, and having a power component. These elements are incorporated on one or numerous boards, and are enclosed in little, cubic inches. By means of modern technologies, a sensor node with 2 AA batteries and 1% low-duty cycle working nodes is capable of living 3 years.

Figure 2.1 shows the MICA2-DOT mote sensor nodes manufactured by Crossbow Technology, Inc. The top left is a quarter used to show the sensor node's size. These nodes have a CPU, a transceiver (transmitter and receiver) for wireless communication, some programmed data memory, and a battery. These sensors may be connected with various physical sensors. A wireless sensor network consists of many sensor nodes deployed in a wide geographical area,

and provides unprecedented opportunities to sense and instrument the large environments. These nodes could sense, for example, light, temperature, moisture, and video streaming.

Figure 2.1. MICA2 motes and STARGATE nodes.

With homogeneous sensor networks, all nodes have similar capabilities, such as processing power, memory, battery life, communication, and other components. It has been shown that homogeneous sensor networks have poor fundamental performance limits, scalability, etc. Therefore, to better the performance and to increase the security, a sensor network may contain different node types. Such networks are known as heterogeneous sensor networks. Usually heterogeneous sensor networks have a small number of powerful, high-end sensors. The STARGATE nodes shown in Figure 2.1 and a large number of low-end sensors, e.g., the MICA2 mote sensors, are both manufactured by Crossbow. High-end sensors have better potential than low-end sensors in terms of communication, computation, power supply, security, and other elements.

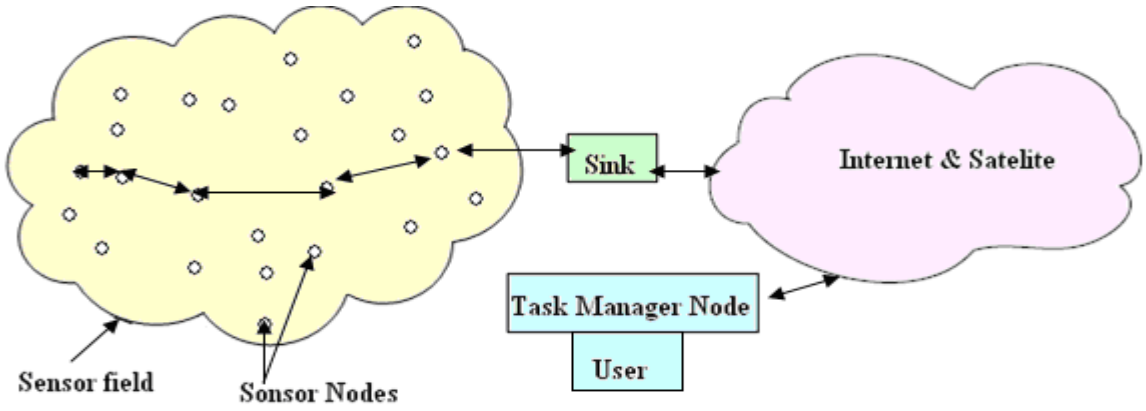


Figure 2.2. Sensor networks connected to the Internet.

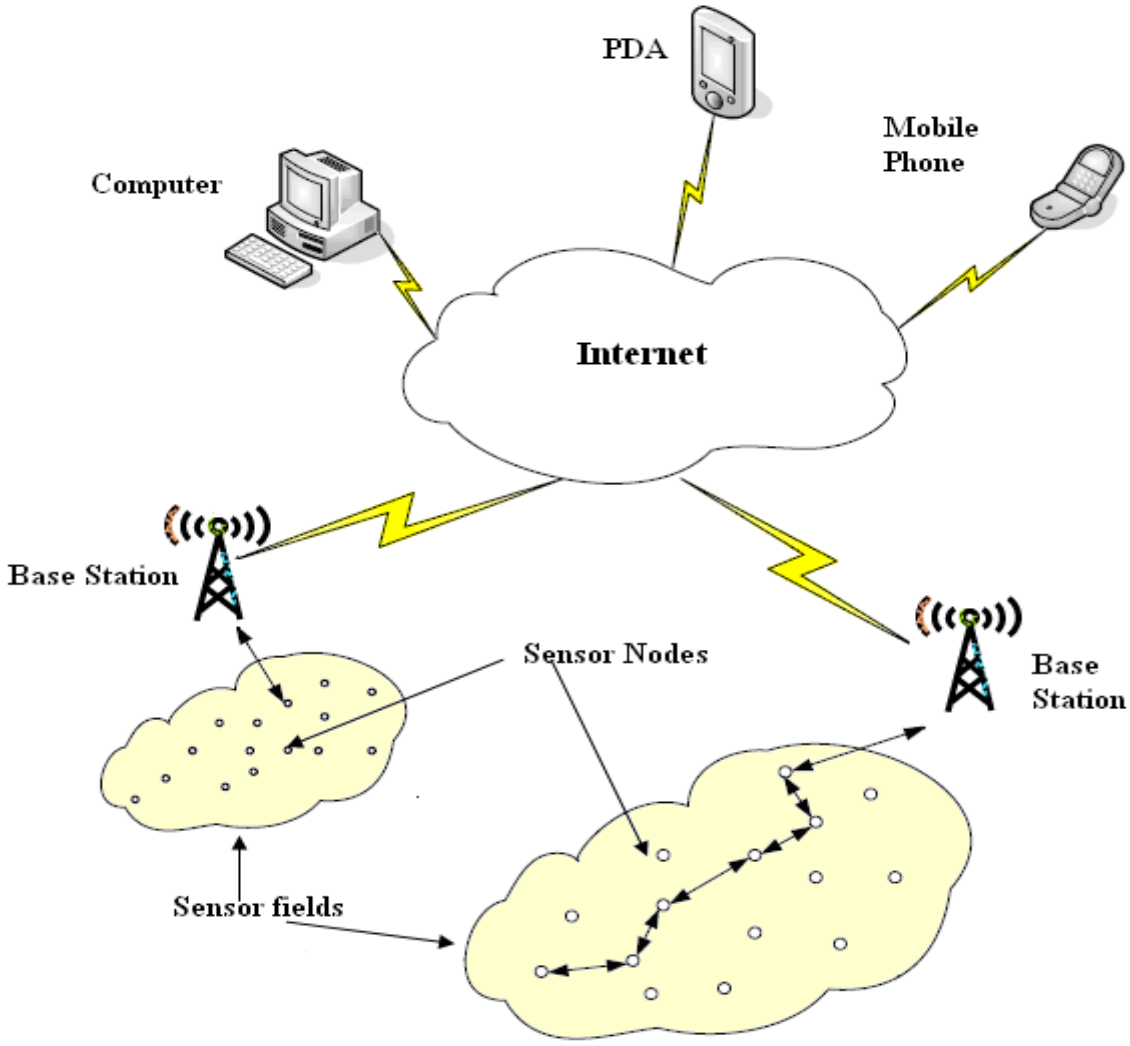


Figure 2.3. Global information sharing.

Once the sensor nodes are deployed, usually in an ad-hoc manner, the nodes are responsible for self-organizing in a suitable network organization, where they have multi-hop connections among other nodes. By means of a discrete or a continuous working mode, the sensor nodes then start collecting temperature, humidity, acoustic, seismic, infrared, or magnetic information about the environment. With a global positioning system (GPS) or local positioning algorithms, the location and positioning information can also be obtained.

Users can retrieve the required information from the sensor network by sending queries to the nodes, and the results could be obtained from the base station or sink, an interface among users and the nodes. That is why wireless sensor networks are considered to be distributed networks. Maybe future sensor networks will be connected to the Internet, and global information sharing will become possible as shown in Figures 2.2 and 2.3.

2.2. Applications

Primarily, the research on sensor networks was initiated by armed forces purposes, such as ground surveillance of unattended land sensors for target detection, acoustic surveillance for the ocean, etc. On the other hand, with the accessibility of small and cheap sensors, sensor networks can now be utilized for numerous applications, such as environment and habitat monitoring, traffic control, infrastructure security, and industrial sensing.

2.2.1. Environment and Habitat Monitoring

Temperature, humidity, etc. are measured for environment monitoring to study the vegetation response to environmental changes and diseases [49]. For habitat monitoring, acoustic and imaging sensors are used; they can recognize, track, and determine the population of birds and other species.

The Brazilian government has implemented a very large-scale system, the System for the Vigilance of the Amazon (SIVAM), for environmental monitoring, drug-trafficking, and air traffic surveillance for the Amazon Basin. It makes a huge network of different, interconnected small networks consisting of radar, imagery, and environmental sensors. The radars are situated on aircraft; imagery sensors are placed in space; and environmental sensors are distributed on the ground. The interconnected networks operate at different speeds because radar and imagery are connected through high-speed networks on satellites and aircraft while the ground-based sensors are connected through low-speed networks.

2.2.2. Traffic Control

Overhead sensors or buried sensors are used for vehicle-traffic monitoring and control at most traffic intersections, detecting vehicles and controlling traffic lights [40]. Moreover, video cameras are utilized to capture the view of road segments with heavy traffic. The pictured views are sent to a controlling site where human operators regularly scan the images. However, video surveillance is costly, so it limits the traffic monitoring to certain regions where the sensors could detect and count vehicle traffic and could estimate its speed. These sensors can communicate with each other and can process the collected data which, further, could be communicated through ad-hoc networks to report to the central site for monitoring by human operators or automatic controllers to produce control signals.

These sensors could also be used in a vehicle to assist with driving and making decisions to avoid traffic jams and plan alternative routes. When the vehicle passes by other vehicles or ground sensors, the sensors share information with each other about the location of traffic jams as well as the traffic's speed and density. Therefore, these sensors lead to distributed traffic.

2.2.3. Infrastructure Security

Sensor networks also help to protect vital infrastructure, buildings, airports, communication centers, dams, bridges, power plants, etc. against terrorist attacks [27]. A collection of different sensors, such as video, acoustic, and other sensors, could be positioned in and around these facilities to make information available prior to revealing potential threats.

This system can sometimes lead to false alarms, thus enhanced coverage and a reduced false-alarm rate could be attained by combining data from a number of different sensor types. More flexibility and enhanced coverage could also be achieved by connecting these networks to wireless ad-hoc networks [38]. Chemical, biological, and nuclear attacks could also be identified by using sensor networks.

2.2.4. Industrial Sensing

Sensor networks are helping industrial organizations lower production costs by improving machine performance and by lowering machine maintenance [40]. This could be achieved by placing sensors onto places which humans cannot reach; thus, the sensor could determine the vibration, or the wear and tear, and the lubrication levels.

2.3. Technical Problems

While working with sensor networks, we encounter technical problems, such as processing data, information transmission, and sensor management, because the environment is usually dynamic and could be harsh and uncertain. The networks generally face energy and bandwidth constraints. Furthermore, the wireless sensor ad-hoc networks cause supplementary technical challenges, such as the discovery of efficient routes, network management, querying, shared information processing, and tasking.

2.3.1. Discovery of an Ad-Hoc Network

For a sensor to function efficiently in the network, it is necessary that the sensor knows to which network it belongs, what the node identity is, where the nodes are located, and what neighbors it has for the purpose of processing the data and collaboration. When the nodes in the network are set with planning, then topology is known as a priori. When the nodes are distributed in an ad-hoc fashion, the topology of the network is constructed in real time and revised once sensors fail or new sensors are added to the network. When the nodes are mobile in the network, the topology is constantly changing, so there must be some method in which fixed and mobile nodes can find each other's location. Because nodes only interact with their neighbor nodes, keeping tabular information about each node's global positioning is not desired. It is very costly for nodes to find their own location using GPS, so relative positioning algorithms should be applied.

2.3.2. Routing Data and Controlling the Network

The sensor networks have limited resources, such as bandwidth, processing power, and energy. Once the sensor nodes in the network are deployed, the network operates autonomously, but the resources change dynamically. Because it is an ad-hoc network, it does not have planned connectivity among nodes. As the data-transfer need surfaces, nodes can get connected by the algorithms and software.

In such networks, the communication links are unreliable, and shadow fading may eliminate links; thus, the system design should be reliable enough that the nodes could transfer data successfully. Doing this call for more research into two areas of concern, for example, what the network size is or how many nodes and links are needed to make available sufficient redundancy? If the network on the ground is using radio frequency (RF) transmission, then

frequency attenuates with distance more rapidly than in free space, which implies that the distance for data transmission and energy utilization should be well managed.

2.3.3. Information Processing and Signaling

In sensor networks, the sensor nodes collect and process data to produce useful information. Information processing and signaling in a network is associated with distributed information fusion. Theissues, such as how much information the nodes should share with each other and how nodes process information gathered from other nodes should be addressed. For better results of gathered data in the network, the data should be processed from more sensors, resulting in more utilization of resources such as energy, bandwidth, etc. Once information is received from a node in the network, this information is fused with the node's own data. For the best results, while combining or fusing the data, the node could use model-based techniques that reflect how the information is generated.

2.3.4. Tasking and Querying

A sensor network collects numerous data sets with distinctive characteristics. The data are gathered from the environment by geographically dispersed nodes sensing them, and then they are distributed across nodes connected by unreliable links. Thus, it becomes essential that customers or users have a simple interface to interactively task and query the sensor network. Finding capable, distributed methods for query and task compilation is another challenge for data placement and organization.

2.3.5. Security

Generally, the sensor networks function in a hostile environment, thus security should be built into the given network. Different modus operandi could be used to make secure and

survivable networks available. Because these networks are used for detection, they should have built-in protection against spoofing and intrusion.

2.4. Sensor Network Architecture

The sensor nodes in the network could be deployed following two types of architecture, layered and clustered, as shown in Figure 2.4. A description of layered and clustered architecture is given below.

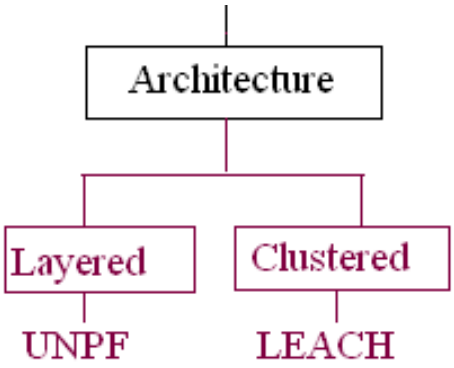


Figure 2.4. Architecture.

2.4.1. Layered Architecture

When the network is deployed, the sensor nodes that are not adequately close to the base station transfer their sensed data to their neighboring nodes, communicating over nodes to reach the base station. Figure 2.5 shows how the layers could help nodes to communicate their information to the base station. Unified Network Protocol Framework (UNPF) follows the layered architecture.

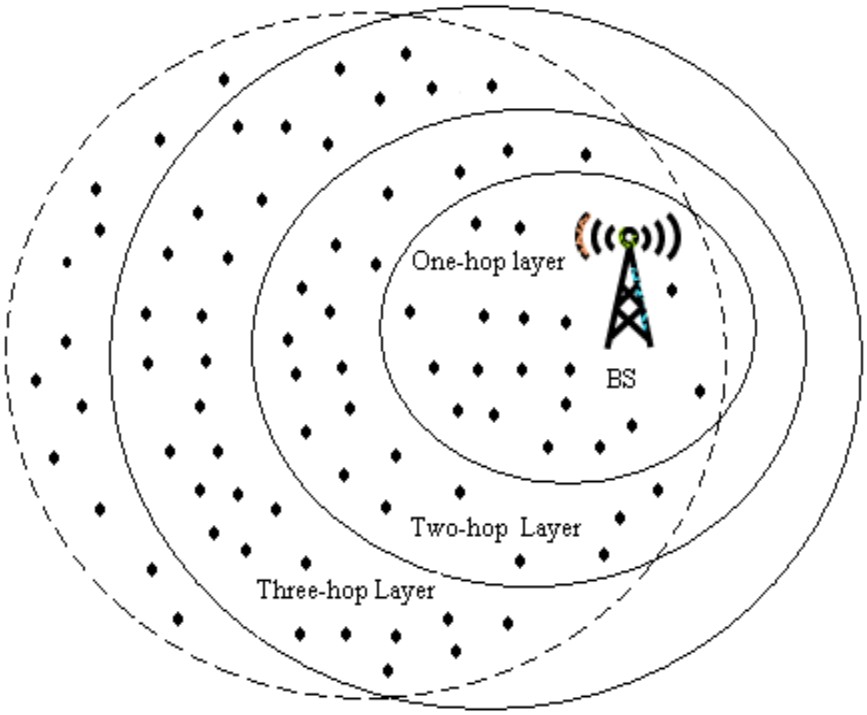


Figure 2.5. Layered architecture.

2.4.2. Clustered Architecture

In this architecture, the sensor nodes are organized into clusters. Each cluster has a cluster head which has generally more resources, such as energy, processing power, bandwidth, etc., than other nodes in the cluster. All nodes in one cluster communicate to the base station through the cluster head. The cluster head is responsible for processing the data collected from all the nodes and then transferring them to the base station by finding the appropriate route, where cluster formation is an autonomous process [37]. Figure 2.6 shows how the clusters communicate information to the base station. Low-Energy Adaptive Clustering Hierarchy (LEACH) [20] follows the clustered architecture.

Data Dissemination is the method by which data, information, or queries are routed in the sensor network. It is a two-step process, first interest propagation and second data propagation.

An interest is considered data, an event, or an objective with which the node is working, such as to capture temperature, humidity, pressure, etc. When the base station or one node is interested in some event, it broadcasts its interest to all its neighbors and, from time to time, retransmits its interest [1]. The data propagation has been made proficient by different data-routing algorithms for different interests, such as flooding, gossiping, rumor routing, sequential-assignment routing, directed diffusion, sensor protocols for information via negotiation, the cost-field approach, geographic hash table, and the minimum-energy communication network [53].

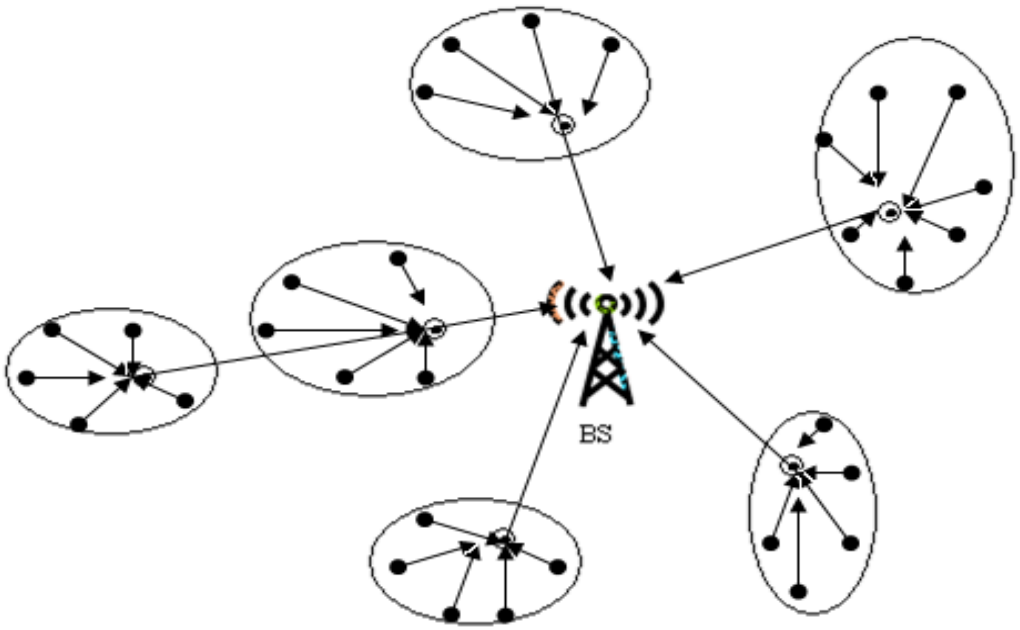


Figure 2.6. Clustered architecture.

2.5. Main Issues

The main issues that should be addressed in wireless sensor networks are as follows.

2.5.1. Routing

Routing is of significant importance in wireless sensor networks (WSNs). The routing approaches used for MANETs (Mobile Adhoc NETworks) and the Internet do not execute well in WSNs. The routing methods on the Internet use wired connections to reduce the error rate, but

WSNs cannot use wired connections. Generally, MANET connections are symmetric; i.e., if transmitting node Q can reach another node P, the reverse is also true, meaning that P can also reach Q with ease. However, this may not be the case with sensor nodes in a WSN. These differences lead to discovering new routing protocols for WSNs [53].

In WSNs, the sensors are generally deployed in an ad-hoc manner, and the first step towards routing is neighbor discovery. During neighbor discovery, the node sends a number of packets to its neighbor to generate its neighbor table. The neighbor table holds data about the neighbor's ID and its location. Thus the node should know its geographic location before neighbor discovery. Another attribute that this table holds is the remaining energy level of the node, a quality estimate of the link and packet delay when data pass through that node. Once this table is built, the routing scheme forwards packets from the source to the destination, depending upon geographic coordinates and not based on the node's ID. An example of this routing type is Geographic Forwarding (GF) Routing. Directed Diffusion (DD) is another routing technique that puts data aggregation, queries, and routing together. Usually, a remote node comes up with a query showing the interest in specific information, such as the temperature. The node with correct data replies with an attribute-value pair. This attribute value is forwarded to the remote node using the shortest physical distance, which is set up and updated throughout the query processing time. To reduce the transmission cost, the attribute value is aggregated while it is on its way to a remote sensor node. To increase the routing robustness, the data could follow multiple paths [43].

2.5.2. Reliability

The data hop from node to node in a link from the source to the destination, so it is important that the link is highly reliable; otherwise, the data could be easily compromised. Reliable links generally use some metrics, such as the packet delivery ratio, the received signal strength, and the link quality index which is based on the error rate, to ensure reliable data delivery. Generally, the links are asymmetric in the WSNs; i.e., if node A can communicate successfully with node B, then node B may not be able to communicate reliably with A. Because of this non reliability, the DSR and AODV routing algorithms of MANET would not do well in the WSNs.

2.5.3. Wake and Sleep Schedules

In WSNs, energy is a big constraint. To save energy, a number of WSNs put sensor nodes in the sleep mode. While routing data, this strategy creates problems because the awake node cannot select an asleep node to communicate its data to the destination.

2.5.4. Real-Time

Few applications want data to reach their destination within a specified time limit, but due to a number of uncertainties, the WSN routing algorithms face difficulty achieving this objective. SPEED (Stateless Protocol for Real-Time Communication in Sensor Networks) and RAP (Route Access Protocol) protocols use the velocity metric that unites the time limit and distance that the data should move to achieve the real-time transmission [24].

2.5.5. Mobility

In WSNs, some sensor nodes could be moving, so the routing becomes complex if the source sensor, destination sensor, or both are mobile. To resolve this complex situation, the sensor nodes should continuously update their neighbor tables [38].

2.5.6. Voids

It is possible that some sensor nodes do not find any forwarding node in the routing path in the direction they want to forward the data because of the limited transmission range. The GPSR (Greedy Perimeter Stateless Routing) [24] protocol deals with this problem by selecting another node that is not exactly in the desired direction but discovers a route around the void.

2.5.7. Security

The antagonist can commit a number of attacks on the routing algorithm, such as Sybil, selective forwarding, replays, black hole, wormhole [28], DoS (Denial of Service) attacks, etc. Only a few WSN routing algorithms address these attacks. Some protocols, such as two-tier HSN routing and SPINS (Security Protocols for Sensor Networks), have embedded security in the routing algorithms.

2.5.8. Congestion

Usually, the WSNs have cyclic or irregular traffic, so such networks are not prone to congestion. However, the WSNs that process audio and video and that have multiple base stations are prone to congestion. To avoid congestion, scheduling and a reduced transmission rate could be followed.

2.5.9. Clustering

Some WSNs are using two types of sensor nodes, low-energy and high-energy nodes, to secure the network and for efficient routing. Low-energy nodes are generally larger in number than high-energy nodes. When such a network is deployed, the high-energy nodes make clusters with the low-energy nodes and become the cluster head. All the low-energy nodes under one cluster head only respond to that node, and after gathering the required information from the low-energy nodes, the cluster head finds another cluster head to route the data to the base station.

2.6. Background of Tabu Search

The Tabu search was first introduced in the 1970s by Glover [17]. Supplementary attempts for its improvement are described in later research [8], [15], [16]. The Tabu search came into existance in 1986. Through numerous computational experiments, it has been proven that the Tabu search is a good heuristic technique. It could contend with any good technique. Thus far, there is no proper justification for its excellent performance. In recent times, the theory behind the Tabu search has been scrutinized [11], [14], [19], and an informative report on the Tabu search and its applications has been composed into a book [18].

Tabu's success could be attributed to the fact that its modeling is done sincerely from the start. To make this method work for different application types, numerous modifications have been suggested by different researchers. All these suggestions and modifications are based on a number of different ideas and are distinguished by a particular characteristic.

The iteration plays a significant role in heuristic techniques. The common steps followed in iteration are generating solution *q* from an initial solution, *p*; then, it is verified whether one should stop here or carry out one more iteration. Neighborhood search procedures are iterative, too. In such procedures, a neighborhood, *N(p)*, is given for every possible solution, *p*. The next good solution, *q*, is investigated from the given solutions in *N(p)*.

2.7. Tabu Search: Basic Idea

To advance the effectiveness of the investigation method, one should have information about the local optimum and the investigation method itself. An indispensable characteristic of the Tabu search is that it uses memory effectively. Almost all investigation methods keep the value, *f(p*)*, in the memory of the finest solution, *p**, calculated until the point. The Tabu search also maintains information about the path followed from the last-visited solutions. This

information is used to decide the next move from p to the next solution, q, that is selected from $N(p)$. The memory limits the selection to the subset of $N(p)$ by restricting some moves to neighborhood selection. Where practical, the arrangement of the neighborhood, $N(p)$, for solution p changes from one iteration to another. Therefore, the Tabu search is also known as the dynamic neighborhood search method.

In one optimization problem, say we are given a set, S, of all possible solutions and a function, f, where $S \rightarrow \psi$. We are to find a solution, p^*, in S such that $f(p^*)$ is suitable under some condition. Usually, a condition of suitability for solution p^* is, if $f(p^*) \leq f(p)$, for every p in S. In this type of cases, the Tabu search is a minimization algorithm if the investigation method assures that p^* will be found after a finite number of iterations. On the other hand, there is no assurance that such p^* could be found; consequently, the Tabu search is a common heuristic method. In view of the fact that the Tabu search consists of some heuristic method, it is better to describe it as a meta-heuristic. The Tabu search generally directs the search to different search methods. The pseudo code for the Tabu search is given as follows:

1. Select a preliminary solution, p, in S.
2. Create a subset, A*, of the solution in $N(p)$.
3. Search for the best q in A*, and set $p=q$ only if $f(q) \leq f(r)$ for any r in A*.
4. If the condition $f(q) \geq f(r)$ is true, then stop; otherwise, go to 2.

In the given case, if $A^*=N(p)$, then the search may get into much computation. Therefore, A^* should be selected with care. If $|A^*|=1$, the algorithm could skip the selection of a best solution, q.

This procedure is likely to trap in local minima where the solution may be far from the global minima. To solve this problem, several cases should also allow non-improving moves

from *p* to *q* in *A**, meaning that one should check if *f(q) > f(p)* so that no one could escape from a local minimum. Simulated Annealing (SA) could be used to do the same [6], but SA does not lead to the selection of *p*. The Tabu Search selects the best *p* in *A**. If there is a possibility of non-improving moves, the possibility of visiting the solution again and the possibility of cycling increase. To reduce such risks, one should use memory to keep track of revisits and cycles. To implement it, we need to introduce a for loop for *l* iterations, so the neighborhood structure is now *N(p,l)* in the form of a matrix, instead of *N(p)* as an array. An enhancement in the previous algorithm forces it close to the common Tabu search method. The improved algorithm is as follows, where *p** is the best solution brought into being and *l* is the number of iterations:

1. Select a preliminary solution, *p*, in *S*. Set *p*=p* and *l*=0.
2. Increment *l* by 1. Determine a subset, *A**, of the solution in *N(p,l)*.
3. In *f* or in modified *f'*, search for the best *q* in A*, and set *p=q*.
4. If we found that *f(p)<f(p*)*, then set *p*=p*.
5. If the condition *f(p) ≥ f(p*)* is true or the stopping criteria are met, then stop; otherwise, go to 2.

Some of the Tabu search-stopping criteria are as follows:

1. *N(p,l+1) = ∅*.
2. The maximum number of iterations allowed, *l*, exceeds its limit.
3. Since the previous enhancement of p*, the number of iterations is greater than a particular number.
4. It could be proven that the best possible solution has been found.

The characterization of *N(p,l)* at every iteration, *l*, and the selection of A* are essential because they persuade the search process and its outcomes. The characterization of *N(p,l)*

involves removing various recently visited solutions from *N(p)*; those solutions are kept away from consideration in the next iteration and are marked as Tabu solutions; this approach, to some extent, avoids cycling. By keeping a Tabu list, T, for *l* iterations, and |T| are the last solutions visited, then up to |T| cycles could be prevented, which could be represented as $N(p,l) = N(p) - T$. It is mandatory to explain the discovery procedure in S with respect to moves to see how T is used. For every solution, *p*, in S, *X(p)* is the set of moves that could be applied to *p* to get the new solution, $q \Rightarrow q = p \oplus x$. Then, $N(p) = \{q / \exists x \in X(p) \text{ with } q = p \oplus x\}$. The moves used in the process are reversible, meaning that, for any move, *x*, there exists a move, *x-1*, such that $(p \oplus x) \oplus x\text{-}1 = p$.

By keeping only the last moves, |T|, or the last reverse moves, |T|, one can reduce the cycles, but with this limit, the information gets lost, which does not assure that it will reduce the cycles. At a time, one can use numerous lists, T_r, to improve the efficiency. Various components, t_r, of *p* or of *x* are tabooed to specify that these components are presently not permitted to be engaged in a move. The taboo condition of a move depends on the taboo condition of its components and could vary with every iteration, and taboo conditions could be devised as t_r (p,x) $\in$ T_r, where r=1,...,t.

The *x* move could be applied to solution *p* and could be set as taboo if this move satisfies all the conditions. One problem with this approach is that a taboo status could be assigned to a solution that has not been visited, so the aspiration-level condition is used to relax such tabu status when some Tabu solution appears optimal. The solution seems optimal if its aspiration level, *a(p,x)*, is better than its threshold value, *A(p,x)*, where *A(p,x)* is a set of favored values for a function. *a(p,x)*. The aspiration conditions are written as $a_r(p,x) \in A_r(p,x)$ (r=1,...,a). If any of these conditions are fulfilled by some Tabu move, x, applied on p, then x is an acknowledged

move. In this approach, *f* may sometimes be changed by a different function, *f'*; this way, we bring intensification and diversification to the search.

Sometimes, it is good to intensify the search in some specific area of *S* because it could have some suitable solution. Intensification is the process of providing more priority to the solution which shares the same characteristics with the existing solution. The above said is implemented by adding one element in the objective function. After exploring one area, it is valuable to explore other areas of *S* as well; this step is known as diversification and likely extends the exploration process to all other areas of *S*. Diversification is also implemented by adding one element in the objective function. Intensification and diversification elements have values that are changed in a manner that these both phases alternate with each other during the search where a changed objective function, *f',* is given as *f'=f* + Intensification + Diversification.

Therefore, the Tabu search could be described as follows:

1. Select a preliminary solution, *p*, in *S*. Set $p^*=p$ and $l=0$.
2. Increment *l* by 1. Determine a subset, A^*, of the solution in $N(p,l)$ such that either one of the Tabu conditions, $t_r(p,x) \in T_r$, where r=1,...,t, is violated or at least one of the aspiration conditions, $a_r(p,x) \in A_r(p,x)$, where $r=1,...,a$, is met.
3. In *f* or in modified *f'*, search for the best $q = p \oplus x$ in A^*, and set $p=q$.
4. If we find that $f(p)<f(p^*)$, then set $p^*=p$.
5. Revise the Tabu and aspiration conditions.
6. If the stopping criteria are met, then stop; otherwise, go to 2.

As we saw, the Tabu search was gradually improved. Still, some researchers apply additional heuristics to solve a particular problem.

2.7.1. Tabu List Size

The Tabu list mainly prevents cycling, but the question is about the size of the Tabu list. If the Tabu list is very small, then it may not prevent cycling, and if it is very long, then it may generate a number of limitations. Moreover, the mean value of the visited solutions may increase as the Tabu-list size increases, so a Tabu list with variable size could be used where every component goes through a number of iterations which have upper and lower bounds.

2.7.2. Intensification

An initial solution is provided; then, that problem is divided into smaller parts. A fast heuristic and a reasonably sized neighborhood are used to reduce the computation time for the search, and then, the memorized best solutions or best moves are evaluated in terms of the best components.

2.7.3. Diversification

Diversification is important while searching to circumvent the regions from being absolutely unexplored. One way is to do numerous random restarts. Another way is to penalize repeated moves or revisited solutions that guarantee that the unvisited areas will be visited.

2.8. Literature Review

In this section, we discuss some research work related to routing, clustering, and security for sensor networks. The researchers have started study about embedding security in the routing protocol's design. Some researchers discuss routing in terms of geographic routing, robust routing, and routing and aggregation. Many researchers are trying to improve routing by first applying clustering to the nodes, thus leading to efficient routes.

2.8.1. Security in Sensor-Network Routing Protocols

Security and routing protocols are main issues for sensor networks. The confidentiality and security concerns in the field of sensor networks correspond to a vast area of research problems. One improvement could be enhancing the hardware and software in the sensor nets, but still, new and supporting technology is required. Some researchers address this issue as follows.

Karlof and Wagner were the first to suggest that routing algorithms should be designed to support the security of routed data. They showed, in their paper titled "Secure Routing in Sensor Networks: Attacks and Countermeasures," that attacks which are prevailing for the ad-hoc and peer-to-peer networks could also be used against sensor networks [23]. They also mentioned that sensor networks could face sinkhole, hello, and crippling attacks, so secure countermeasures should be embedded in the routing algorithms' design.

Perrig et al. introduced a new routing algorithm, SPINS, in their paper, "SPINS: Security Protocols for Sensor Networks," which optimizes sensor networks [35]. It had two parts, SNEP and μTESLA. SNEP provided confidentiality and freshness to the data by allowing two-party data authentication. μTESLA supported the authenticated broadcast for the sensor networks, where resources were available in limited supply.

Du et al., in their paper "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks," discussed how security could be embedded in a routing algorithm because only few existing routing algorithms address the issue of security while designing the algorithm [9]. They used heterogeneous sensor networks to achieve this goal, and they proved this notion by showing the results using simulation.

2.8.2. Geographic Routing

In geographic routing, all the sensor nodes have information about their location, their neighbors, and the base stations. The routing node forwards packets to the closest neighbor with the shortest distance to the base station. The nodes do not keep routing information but the neighbors do, so any change in the network gets in effect immediately.

Karp and Kung presented a new routing protocol named Greedy Perimeter Stateless Routing (GPSR) for datagram forwarding in a wireless network. It makes the datagram forwarding decision based on the router's positions and the datagram's destination. It is called greedy because it, mainly, picks up a neighbor router for the datagram forwarding to the destination. In the experimental study, Karp and Kung found that this protocol performs better than the shortest path because the number of destinations increases in the network. They compared this protocol with dynamic source routing and found that GPSR demonstrates scalability on thickly populated wireless networks [24].

2.8.3. Robust Routing

The general idea of robustness is to find a path that minimizes the maximum regret in the case when the network topology changes. Some of the literature is discussed below.

Thanos et al., in their paper titled "End-to-End Routing for Dual-Radio Sensor Networks," described a control mechanism for the given topology: to establish an end-to-end route among dual-radio nodes [43]. They used secondary radios to control the wake-up mode for the selected nodes utilized to route the data to high-end nodes. They empirically showed that their methodology saves 60% more energy than the alternative methods and had higher latency based on the application.

In the paper titled “Disjoint Multipath Routing to Two Distinct Drains in a Multi-Drain Sensor Network,” Thulasiraman et al. increased the robustness of their routing algorithm by making use of two distinct, multiple data-collection nodes that are disjoint [45]. This activity decreased the average route length compared to routing to distinct, multiple data-collection nodes that are disjoint. The calculated time complexity of the demonstrated algorithm is O(|D||L|), where |D| is the number of data-collection nodes in the network and |L| is the number of links joining the nodes in the network.

2.8.4. Routing and Aggregation

The purpose of route aggregation is to reduce the routing table size by information hiding and by reducing the time to converge. Yang et al. in their paper, “SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks,” described how communication overhead and energy consumption could be reduced with hop-by-hop data aggregation during data collection in sensor networks. The divide-and-conquer and commit-and-attest principles were used for the SDAP protocol design. SDAP divides the nodes into logical groups of equal size, finally making a tree topology where the root is the base station [50]. Then, each logical group takes part in an attestation procedure to demonstrate the accuracy of its group aggregation.

The paper titled “Querying Sensor Networks Using Ad Hoc Mobile Devices: A Two-Layer Networking Approach,” written by Tian et al., presented a two-layer architecture for the network consisting of mobile nodes at the upper layer and static sensor nodes at the lower layer for the end-to-end query processing. This heterogeneous design helps the resource-constrained sensor networks work for a long duration by using powerful mobile nodes. The authors supported their methodology by showing numerical and simulation results [46].

Cheng et al., in their paper "Route Recovery in Vertex-Disjoint Multipath Routing for Many-to-One Sensor Networks," described how multipath routing could be applied in sensor networks for load-balancing and fault-tolerance as well as to increase security [7]. When a node with the data finds that a new path may not exist, then the algorithm guides the mobile nodes with the data to move themselves to a place that the new multipath could be constructed, and if it can be constructed, then it node can launch the route.

The Directed Diffusion (DD) algorithm for remote surveillance in sensor networks was implemented by Intanagonwiwat et al. in their paper "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks" [21]. With the progress in technology, the sensors are less expensive. Using Directed Diffusion routing in sensor networks, all the nodes are application aware, saving energy as a result of choosing good routes and caching the data to process them in the network.

2.8.5. Clustering

The routing algorithm for WSNs which are cluster-based minimizes the number of hops for the data to propagate through the network each time an event occurs. Misra and Xue in their paper, "SAS: A Simple Anonymity Scheme for Clustered Wireless Sensor Networks," proposed an efficient scheme for establishing anonymity in the clustered, wireless sensor networks [30]. For authentic and confidential communication, the nodes forming a cluster with the neighborhood nodes share pair-wise keys. To hide its identity, the node uses a range of pseudonyms as identifiers in the network. This way, the compromised node could not identify itself either to the sender or to the receiver for the communication taking place between other communicating nodes. The authors claimed that their approach needs low memory and works with low computation cost.

The paper titled “Secure Distributed Cluster Formation in Wireless Sensor Networks,” by Sun et al. describes that, if we want to achieve scalability and want the sensor nodes to self-organize, there should be less power usage, and the route should be robust for sensor networks; then, the sensor nodes should be clustered into small groups. The authors have proposed a secure, distributed cluster-formation protocol that organizes the sensor nodes into mutually disjoint cliques [41].

Misra and Xue, in their paper “CluRoL: Clustering Based Robust Localization in Wireless Sensor Networks,” described how anchors know their own positions and help the sensor nodes to localize themselves [31]. During localization, malicious anchors could give the wrong data about their position and distance from the given sensor node. In this paper, the researchers described how sensor nodes could identify malicious anchors and get rid of them during localization. The authors proposed a technique, CluRoL, which makes clusters of proximal points that help every sensor node to localize itself correctly so that it could discover the false anchors and keep them outside its localization. The authors’ simulation results showed that this protocol can identify 72% of the malicious anchors.

The paper, “Optimal Number of Clusters in Dense Wireless Sensor Networks: A Cross-Layer Approach” [47], by Wang et al., described how finding the optimal number of clusters in a sensor network is an important issue to reduce energy consumption and to reduce the cost for maintaining the link. In doing so, they proposed a physical (PHY)/medium access control (MAC)/network (NET), cross-layer analytical approach. Through simulations, they found that this technique reduces energy consumption by more than 80% in some cases. They also found that, in spite of the varying densities of sensors in various areas, the protocol functioned effectively.

Younis et al., in their paper "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," classified different clustering approaches as per their design rationale. They did this because clustering organizes the large number of sensor nodes in the connected hierarchy and balances the data-routing load, extending the network lifetime [52].

Bonivento et al. proposed a mathematical protocol named SERAN to optimize the parameters of the protocol, and also to initialize and maintain the given network in their paper, "System Level Design for Clustered Wireless Sensor Networks" [5]. The parameters for the protocol were automatically set to suit latency requirements and to optimize the power utilization. SERAN is a combination of random and deterministic methodologies. The protocol is robust because the randomization determinism evades packet collision and lets the protocol extend with the network size.

The paper titled "A Tabu Search Algorithm for Cluster Building in Wireless Sensor Networks," written by El Rhazi and Pierre proposed a new, centralized clustering approach derived from energy maps and QoS requisites of the network for data gathering in wireless sensor networks [1]. The authors compared it with the CPLEX-based (IBM ILOG CPLEX Optimizer) method and found that the Tabu search method gives better results with respect to cluster cost and execution time.

Mhatre et al., in their paper "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," presented a method to minimize the data-routing cost in the network during its lifetime, where a lifetime is the number of accomplishable data-collection trips which are feasible in anticipation of connectivity and/or reporting a loss [29]. The assumption was that all nodes will run out of energy at the same time to minimize energy wastage. The authors

compared the output of grid deployment with random deployment and found that grid deployment is better than random deployment.

The paper titled “Large-Scale Distribution Planning—Part II: Macro-Optimization with Voronoi’s Diagram and Tabu Search,” written by Navarro and Rudnick, presents a planning methodology for the voltage-distribution network in the city of Santiago, Chile [33]. Initially, the regions are divided into Voronoi regions; then, network recombination is performed using the Tabu search. The authors have shown that Voronoi diagrams and Tabu searches, when applied in a sequential manner, provide the best results.

Lee and Kang used the Tabu search to decide the location and capacity of the new base station to fulfill the demand of increased data flow in their paper titled “Cell Planning with Capacity Expansion in Mobile Communications: A Tabu Search Approach.” The results showed that the Tabu search performs better for cost reduction and also outperforms the genetic algorithm for time complexity [26].

Thompson and Orlin, in their paper “The Theory of Cyclic Transfers,” suggested cyclic transfers for the neighborhood search among clusters [44]. They solved it in two steps. In the first step, they assigned the elements to the clusters optimally. In the second step within each cluster, they optimally configured different elements. They defined the data structures for the neighborhood search and established that there is a benefit associated with cyclic transfers.

2.8.6. Tabu Search Technique to Solve the Clustering Problem

The clustering problem is solved by grouping similar elements together from a large number of elements for retrieval. To find a global solution for this problem in realistic time, meta-heuristics are used frequently, such as Voronoi diagrams or the Tabu clustering technique.

It has been proven that the Tabu clustering technique is a successful approach to solve optimization problems, and it could be applied to the clustering problem.

Al-Sultan, in his paper "A Tabu Search Approach to the Clustering Problem" [2], presented a solution to the clustering problem of m objects into c clusters. The Euclidean space is n-dimensional. Those m objects correspond to points in that Euclidean space. He classified the m points into c clusters with the objective that the distance between the center and other points is reduced within the cluster. The clustering problem is a non-convex problem, meaning that it has a number of local minima. The k-means algorithm is a distinguished algorithm for solving this problem. Khaled used the Tabu search technique to develop a new algorithm to solve the clustering problem. The empirical results of the developed algorithm are promising and are contrasted with two algorithms, the simulated annealing and the k-means algorithms.

Sung and Jin, in their paper "A Tabu-Search-Based Heuristic for Clustering," used a heuristic technique to solve the clustering problem using the Tabu search [42]. They divided the given points into a subset of clusters based on similarities and differences between the points. Their heuristic algorithm combined two procedures, packing and releasing, with the Tabu search to get the best results. They compared their algorithm with two algorithms, the *K*-means and simulated annealing algorithms, and found that their algorithm performs better than the other two.

The paper "A Tabu Search Approach for the Minimum Sum-of-Squares Clustering Problem" was written by Yongguo et al. [51]. To handle the Minimum Sum-of-Square Clustering (MSSC) problem, they suggested a Tabu-founded clustering technique. They gave five improvement operations to improve the clustering problem, which they obtained through iterations, and they also gave three neighborhood modes that they used to construct the Tabu search's neighborhood. They used self-generated and real data sets to check the functionality of

the proposed technique, and they have compared it with existing and known clustering techniques.

"Parallelizing Tabu Search on a Cluster of Heterogeneous Workstations" described the parallelization of the Tabu search on parallel virtual machine workstations in a network. Al-Yamani et al. combined two parallelization techniques, the functional decomposition strategy and multi-search threads technique [3]. The authors implemented the domain-decomposition strategy probabilistically. They parallelized the process to speed up the search process. The results showed that both parallelization strategies are advantageous, where functional decomposition technique is giving somewhat good output than multi-search thread technique. The testing was done on VLSI (Very Large Scale Integration) cell placement, with the objective to attain excellent results with respect to area, circuit speed, and interconnection length.

"A Scatter Search Approach for the Minimum Sum-of-Squares Clustering Problem," by Joaqui'n , described a new approach [34]. This technique used the scatter search approach for non-hierarchical clusters to solve the Minimum Sum-of-Squares Clustering problem. This approach combined the local search, GRASP, Tabu search, or path relinking techniques to reduce the computation time. The empirical results showed that this technique gives better results and that the number of clusters has decreased, too.

2.8.7. Tabu Search Technique to Solve the Routing Problem

In the vehicle routing and packet switching networks, system performance mainly depends on routing. By applying the optimization algorithm, one can determine the shortest path in less time. Optimization could improve the throughput [36, 48]. One of the optimal routing techniques could be the Tabu Search (TS), where the objective is to reduce the computational

cost and time. Following are some of the papers that show the effectiveness of the Tabu search approach.

Fisher, and Jaikumar solved the generalized assignment problem for vehicle routing by using a heuristic in their paper, "A Generalized Assignment Heuristic for Vehicle Routing" [13]. They considered a problem where a vehicle fleet transports goods from a central depot to the clients around it. They divided the area into clusters in a manner that the centroid of one cluster is the seed point from which all clients in that cluster are serviced. Every vehicle has some load capability. The problem is to find what request will be serviced by which vehicle and what route should be followed to service the request, such that service cost is minimized. They solved this problem using a heuristic where client assignment to vehicles is calculated through a generalized assignment problem and where the objective function estimates the transportation cost.

Montane, and Galvao, in their paper, "A Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery Service," described the vehicle-routing problem where customers want simultaneous pick-up and delivery service [32]. The vehicle starts delivering supplies from one depot, whereas at the end of the trip, the pick-up consignments are taken to the same depot. The significant point of the given problem is that the vehicle load at any point in the route is a combination of the pick-up and delivery loads. The authors used the Tabu search algorithm to solve this problem. The proposed algorithm had three movement types that give inter-route adjacent solutions: the relocation, the interchange, and the crossover movements. There are four neighborhoods in the implementation; three neighborhoods are defined by the inter-route movements, and the fourth neighborhood is defined by the combination of these movements. The first admissible movement and the best admissible

movement were used to select the next movement. Experimental results were reported for 50-400 clients and for a set of 87 test problems.

The paper, "A Tabu Search Algorithm for the Vehicle Routing Problem," written by Barbarosoglu, and Ozgur, described a new Tabu search heuristic [4]. They developed this approach to resolve the vehicle-routing problem for a single depot of the distribution company. The vehicle carries merchandise from a depot to different, committed wholesalers. The new approach uses the scattering pattern of dealers' locations to set up the neighborhoods. First, a function that ignores the vertex clustering is executed; then, the function that accounts for the relative locations of the dealers is executed; and finally, the realistic non-Tabu move, along with the superlative objective function value, is implemented. A blend of conventional improvement techniques is used during neighborhood construction to realize the replacement of vertices. The mathematical results showed that the algorithm's output is good with respect to other well-known algorithms.

The paper titled "Transportation Planning in Freight Forwarding Companies: Tabu Search Algorithm for the Integrated Operational Transportation Planning Problem" presented the integrated planning problem and proposed an approach for solving it with a Tabu search heuristic [25]. In this paper, Krajewska, and Kopfer described a situation when a fleet and different delegate types are required for demand realization. Then, the authors analyzed the cost to see if it is profitable to own and maintain one fleet.

Scheuerer in his paper, "A Tabu Search Heuristic for the Truck and Trailer Routing Problem," presented a Tabu search heuristic for the truck-and-trailer routing problem [39]. The empirical results for 21 problems showed that the proposed heuristic performs better than existing methods. He treated the truck and trailer as different elements, and there were some

conditions where the clients (places) may be present that are not accessible through trailer. Therefore, the trailer should be detached from the truck, should be left at some parking spot, and should be picked up en route. Side by side, the problem has to solve for the best parking spot and how many times the trailer may be detached. The author introduced a Tabu search heuristic to solve the problem.

In this chapter, we learned that all the researchers applied heuristics to solve their particular problem. Fisher and Jaikumar [13] solved their vehicle routing problem by first partitioning the complete area into clusters in a manner that the centroid of one cluster is the seed point from where all clients in that cluster are serviced in an effective and efficient manner.

Similarly, in our problem, we first see how the clusters will look using Voronoi diagrams, and then, a Tabu search procedure is applied to adjust the nodes in the clusters in a sophisticated way so that the data can be securely routed to the base station with fewer hops. The quality of solution for a candidate set of clusters is measured by the neighborhood of the low-end nodes and high-end nodes, respectively, as well as how many hops/distance apart the nodes are from each other. We have evaluated the cluster set to see how the clusters perform with respect to secure routing while using the least energy. However, routing among the low-end nodes, high-end nodes, and base station has been made efficient by applying the Secured Improved Tree Routing (SITR) technique. This way, we are providing a complete solution for the data in a heterogeneous sensor network by reducing the hop count, making the protocol secure and energy efficient.

CHAPTER 3. DISSERTATION PAPERS

3.1. Introduction

While going through the literature, we found some weaknesses in the current routing and clustering techniques. In the existing literature on route specification in sensor networks, communication among nodes is constrained by parent-child links only, which means that, when the network topology is complex, the shortest-path tree or logical-tree topology is used for data forwarding. However, when nodes fail, this logical-tree topology fails to forward the data packet to the superior nodes. Moreover, no security is added to the design of the routing algorithm.

To generate clusters in sensor networks, either the area under investigation is divided into equal-sized cells, or only Voronoi diagrams are typically used, as shown in Figure 3.1. Although the Voronoi diagram follows the clustering logic of according to the minimum Euclidean metric distance, a more appropriate metric would also consider routing patterns that follow the clustering step. To seek an optimal route, which will be secure and would lead to less energy consumption, we used the Tabu clustering technique along with Voronoi clusters in a sequential manner and then evaluated the method using simulation results. Sequential implementation of Voronoi clusters with the Tabu technique only performed better than Voronoi clusters.

3.2. Papers

Paper 1 is titled “Enhanced Routing in Heterogeneous Sensor Networks.” In this paper, we have configured a new data-routing procedure that potentially improves the network’s energy efficiency by reducing the number of hops utilized to reach the base station. The Improved Tree Routing (ITR) scheme decreases the hop counts for data. ITR is a special tree where the number of offspring is limited to three.

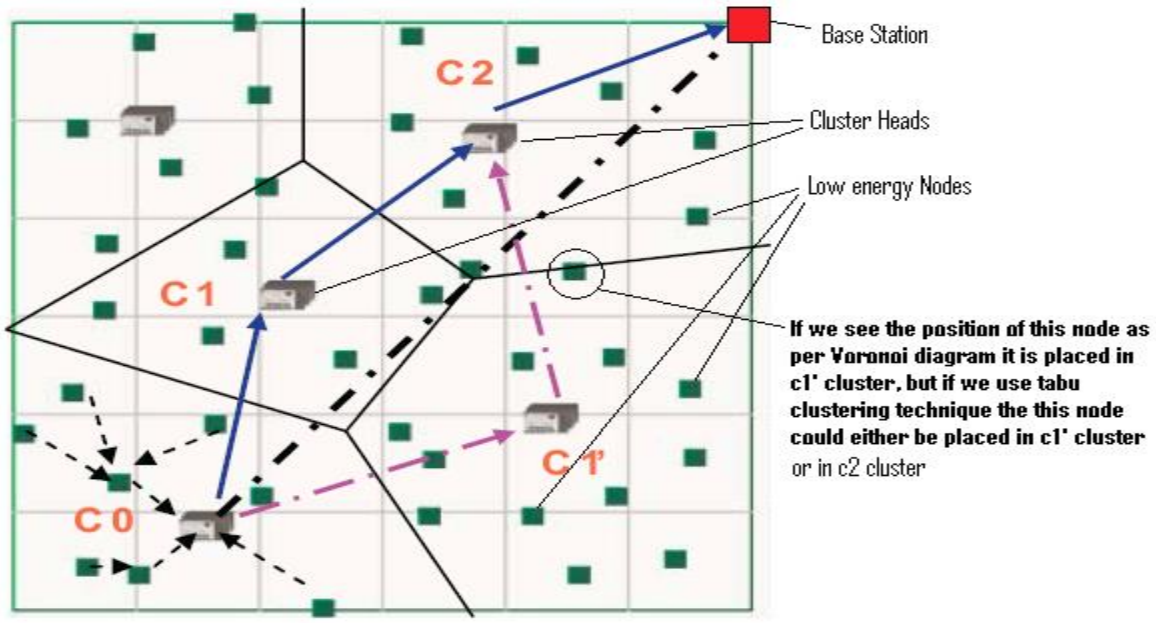


Figure 3.1. The network is divided into clusters using the Voronoi diagram.

The main idea behind such a tree is to use alternative paths with decreased hops for efficient routing. The spanning tree cannot forward data if some node on the route fails. In such situations, the ITR scheme uses a neighbor table to find an alternative path to forward the packets. Upon deployment, the sensors record data about their parent, child, and all neighbor nodes in their neighbor table. A binary address is assigned to each node using the neighbor table. With the neighbor table, nodes generate parent-child links. A node can use alternative path to forward that data, if it finds that this path is shorter. We gain robustness and energy efficiency by reducing the vulnerability points in the network by employing alternatives for the shortest-path tree routing. The new data-routing procedure is compared with the Directed Diffusion algorithm. The empirical results show that ITR is outperforming Directed Diffusion. We learn that the hierarchical network topology, if used with our indexing scheme, can reduce the hop count, leading to fewer chances for the data to be intercepted by adversaries. This topology, in turn,

makes the heterogeneous sensor networks more secure and robust to vulnerable points while also reducing the energy use for nodes transmitting data to their respective cluster heads.

The second paper is "Key Management Scheme for Routing in Clustered Heterogeneous Sensor Networks." This paper puts forward an effective and efficient security scheme for launching anonymity in clustered, heterogeneous sensor networks. The clustered, heterogeneous sensor network has two types of nodes: high-end nodes that work as cluster heads and low-end nodes. The clusters are initially formed by using Voronoi diagrams, and then, some node adjustment is done in the clusters using the Tabu search. The secure scheme uses a unique technique to conceal its true identity (ID). Once the sensor nodes are deployed, the nodes set up their neighbor table and share their secret key with neighbors while ensuring that their true identity is not disclosed and that the communication is secured. The empirical performance evaluations support the given approach's efficiency and effectiveness. The presented scheme is compared favorably with some existing schemes, such as SBK (Self-Configuring Framework for Bootstrapping Keys), E-G (Eschenauer-Gligor), and LEAP (Lightweight Extensible Authentication Protocol), and the presented scheme has less computation overhead, uses little memory to store its keys, and shows a high degree of connectivity with neighboring nodes. Therefore, this paper presents an effective and efficient key-management scheme that is specifically designed for two-tier, heterogeneous sensor networks.

The third paper is "Using Tabu-Voronoi Clustering Heuristics with Key Management Scheme for Heterogeneous Sensor Networks." This paper describes the embedded security in the ITR routing algorithm along with Tabu-Voronoi clustering. In the given sensor networks, the area of interest is initially partitioned into clusters. Voronoi diagram clusters are initially employed and are adjusted using a Tabu search meta-heuristic. Tabu uses a hop parameter to

route the data, whereas Voronoi uses Euclidean distance, leading to a number of anomalies. Moreover, the Tabu search overcomes routing anomalies attributable to holes, wrap-around regions, nodes that lie near the boundaries, and varying node density in the different clusters. Researchers mainly focus on routing the data efficiently and effectively in the sensor network. Only a few researchersconsidered embedding security in the routing protocol's design. The Improved Tree routing protocol is secured by embedding security in it. In this paper, through simulation, we show that the Voronoi-Tabu clustering technique, in conjunction with secure Improved Tree routing, performs better than the Directed Diffusion; LEACH (Low Energy Adaptive Clustering Hierarchy) routing protocol; and some of its own refined protocols, such as V-ITR (Voronoi-ImprovedTreeRouting, TV-ITR (TabuVoronoi-ImprovedTreeRouting), etc. In this paper, an effort is made to provide a complete solution for sensor networks by first applying clustering and then securing the proposed routing algorithm. Hence, this scheme is defending against numerous attacks. Through empirical evaluations, we have found that secure TabuVoronoi-ITR is outperforming the Voronoi-ITR, ITR, DD, and LEACH protocols. Simulation results also show that the described method increases coverage and throughput, and reduces energy utilization and network delays.

3.3. Relationship among Papers

The incremental and bottom-up approaches have been used in this study. Hence, you will find some parts of Paper 1 in Papers 2 and 3. Initially, the Improved Tree Routing algorithm was written and compared with the Directed Diffusion algorithm (Paper 1). Then, the area of interest was divided into Tabu-Voronoi clusters, and two-way, hand-shaking, secure ITR routing was used to route the data and was compared with the results of Paper 1, where ITR without clustering was used to route the data. The two-way hand-shaking, key-management scheme is

embedded in the ITR routing algorithm to make it secure. Added security is defending against a number of attacks, such as, Sybil, wormhole, sink hole, hello flooding, and DoS (Paper 2). Because Voronoi clustering has limitations, a new Tabu search was developed to overcome them by adjusting the nodes in clusters; then, secure ITR routing is applied to route the data (Paper 3).

3.4. List of Advances

This research work was funded by the project titled Designing Robust and Secure Heterogeneous Sensor Networks. This project was sponsored by the Army Research Office (ARO). Dr. Du was the principal investigator (PI); Dr. Nygard was the co-investigator (Co-PI); and there were eight graduate student members. The NS-3 (Network Simulator-3) was used to check our approach's performance. The algorithms were written in C++, and TCL scripts were written to set up the environment.

3.4.1. Contribution of the First Paper

Paper 1 describes the new routing protocol, the Improved Tree Routing (ITR) technique, that we developed. This routing technique improves the network's energy efficiency by reducing the number of hops utilized by data to reach to the base station. The ITR scheme reduces the hop counts for the travelling data. The ITR is a special tree where the number of offspring is limited to three. this tree mainly uses alternative paths with reduced hops for proficient routing. If some node on a route fails, the spanning tree cannot forward data. In such circumstances, the ITR scheme uses the neighbor table to find an alternative path to forward the packets. Because the sensors record data about their parent, child, and neighbor nodes in their neighbor table upon deployment, a binary address is assigned to each node in the network. Using the neighbor table, nodes generate parent-child links. A node can use an alternative path to forward that data if it finds that this path is shorter.

My contributions for this paper are designing, developing, coding, and implementing the ITR algorithm in C++ as an ITRAgent in the NS-3 simulator. This algorithm alone cannot provide the complete solution for the described problem. However, a good heuristic is required to solve the problem using less energy and other resources. Different heuristics have already been used and are an active research area.

3.4.2. Contribution of the Second Paper

The second paper uses a key-management scheme to secure the routing data. The secure-ITR technique was applied to the two-tiered, clustered topology to route the data. It uses an asymmetric, post-distribution cryptography scheme. Once the keys are set up under a given addressing scheme, the nodes use a two-way hand-shaking approach to route the data. Hence, this scheme is defending against numerous attacks, such as Sybil, wormhole, sink hole, hello flooding, DoS, etc. Although embedding security comes with a price of overheads, almost 19% of the overheads are there because of keys. It is found that the overall performance of the Secure-ITR routing protocol is better than the ITR algorithm, although security involved some overheads to route the data. With the simulation results, we find that two-tiered, cluster-based, Secure-ITR routing performs better than the existing ITR routing variant as well as some existing security techniques, such as SBK, LEAP, and E-G. Secure-ITR also performs less computation, uses little memory to store its keys, and shows a high degree of connectivity with neighboring nodes.

My contributions for Paper 2 are designing, developing, coding, and implementing the Tabu-Voronoi-clustering algorithm in C++ as the TabuVoronoiClusteringAgent in the NS-3 simulator. The TabuVoronoiClusteringAgent's output becomes input for the Secure-ITRAgent. The ITRAgent is implemented in Paper 1.

3.4.3. Contribution of the Third Paper

Voronoi is a good heuristic and requires the Euclidean distance to position nodes in a cluster. We are using a hop count to evaluate the data routing. However, Voronoi clustering is based on Euclidean distance, leading to routing anomalies, such as holes, wrap-around regions, nodes that lie near the boundaries, and varying node density for different clusters. With Voronoi clustering, some additional heuristic is required to solve the anomalies. Tabu search is a meta-strategy that has been successful in adjusting local solutions in combinatorial problems. We apply the new Tabu search to adjust the node assignment to clusters, which overcomes the Voronoi-cluster limitations and results in improved routing. We have made Improved Tree routing secure by embedding security in it as mentioned in Paper 2.

My contributions for Paper 3 are designing, developing, coding, and implementing the TabuVoronoi-clustering and Secure-ITR algorithms in C++ as the TabuVoronoiClusteringAgent and Secure-ITRAgent in the NS-3 simulator. The TabuVoronoiClusteringAgent's output becomes input for the Secure-ITRAgent. The asymmetric, post-distribution, cryptography scheme is implemented in the Secure-ITRAgent. The ITRAgent is implemented in Paper 1.

After long and careful discussions with the adviser, we defined the problem statement and put forward a step-by-step solution strategy. First, I designed, developed, and implemented the routing algorithm on the NS-3 simulator [12] as an ITRAgent and compared its performance with the Directed Diffusion algorithm. Next, I divided the given area of interest into Voronoi clusters by writing the VoronoiClusteringAgent in NS-3 and then used the ITRAgent to route the data. The results were compared with the ITRAgent without clustering, and it was found that the VoronoiClusteringAgent with the ITRAgent performed better than the ITRAgent without clustering. The ITRAgent routing was secured by embedding the key-management technique in

it, leading to the Secure-ITRAgent. Finally, the TabuVoronoiClusteringAgent was designed and implemented with the SecureITRAgent, and results were compared with the already obtained results of the VoronoiClusteringAgent with the ITRAgent routing, the ITRAgent routing without clustering and the Directed Diffusion, and LEACH algorithms.

CHAPTER 4. PAPER 1: ENHANCED ROUTING IN HETEROGENEOUS SENSOR NETWORKS †

Kanwalinderjit Kaur
Dept. of Computer Science
NDSU
Fargo, ND, USA
E-mail:
kanwalinder.gagneja@ndsu.edu

Dr. Xiaojiang Du
Dept. of Computer Science
NDSU
Fargo, ND, USA
E-mail:
xiaojiang.du@ndsu.edu

Dr. Kendall Nygard
Dept. of Computer Science
NDSU
Fargo, ND, USA
E-mail:
Kendall.Nygard@ndsu.edu

4.1. Abstract

The research is centered on sensor networks with two types of nodes. The low-end nodes are simple and low cost while the high-end nodes are costly but provide significantly more processing power. In this type of sensor network, the low-end nodes are clustered and report to a high-end node which, in turn, uses a network backbone to send data to a base station. In this research, we have configured a new data-routing procedure that potentially improves the network's energy efficiency by reducing the number of hops utilized to reach to the base station. We gain robustness and energy efficiency by reducing the vulnerability points in the network by employing alternatives to the shortest-path tree routing. The new data-routing procedure is compared with the Directed Diffusion algorithm.

4.2. Keywords

Heterogeneous sensor networks, Base station, Hierarchical routing, and Security.

4.3. Introduction

Wireless sensor networks are used for many different applications, including vehicular movement tracking, nuclear power-plant monitoring, fire-incident reporting, traffic controlling, and environment monitoring. Some characteristics which make the wireless sensor networks that we consider different from standard wireless networks are short battery life, heterogeneous nodes,

a large numbers of nodes with high density in a small area, the ability to tolerate bad environmental conditions, and unattended operations.

In a two-tier Heterogeneous Sensor Network (HSN) [2], two different node types are used to help support the effective transmission of data and to promote energy efficiency. A hierarchical network topology is a natural choice for shortening the path traversed by data to reach the base station. In a hierarchical topology, the sensed data from low-end nodes are forwarded to a high-end node and then onto the base station using high-end nodes. The low-end nodes are expected to be large in number compared to high-end nodes. The base station is assumed to be reliable and trustworthy. It has essentially unlimited battery power, high computing speed, large memory, large bandwidth, and high quality links. After the network is deployed, the low-end nodes must form clusters around high-end nodes to help maximize the network lifetime (due to limited battery life). Because clustering plays an essential role in hierarchical topology, forming high-performance clusters can considerably reduce the network complexity and improve the network performance [8].

Our objective is to make the heterogeneous sensor networks more secure by minimizing the hop counts so that there are fewer chances for the data to be intercepted by adversaries, also minimizing nodes' energy use [1] while transmitting data to their cluster heads. Although the LEACH, SPIN, and Directed Diffusion routing protocols already exist, the new ITR methodology reduces the hop count, and makes the network more secure and energy efficient.

4.4. Tree-Routing Methodology

In our methodology, the Improved Tree Routing (ITR) [7] scheme, criteria for reducing the number of hops required to reach a base station are identified. In our work, we form a special tree in which each node has up to three offspring, which allows indexing the tree in a very

efficient way. The key concept is not only to support the routing tree, but also to rapidly identify, evaluate, and utilize alternative hops with a reduced hop count.

In the related work in the literature on route specification for sensor networks, communication among nodes is limited to parent-child links, which means that, when the network topology is complex, the shortest-path tree or logical-tree topology is used for data forwarding [7]. However, when a node fails, the usual logical-tree topology is unable to forward data packets to their superior nodes. In our methodology, we exploit a neighbor-table property to forward the data packets when nodes fail and also use the same scheme to reduce the number of hop counts even under ideal conditions. Overall, our methodology is an alternative to the shortest-path tree routing and potentially reduces the number of hops.

At the time of deployment in most sensor networks, each node records data for the other nodes within its radio range (such as their addresses) and keeps this information in a neighbor table. Such a neighbor table includes information about child nodes, the parent node, as well as other neighbor nodes within a range. The construction of the neighbor table takes place during the node's link-making process while the node examines its vicinity with the purpose of determining neighbors and locating a promising parent.

In our methodology, the neighbor table is implemented using node addresses which are assigned to each node prior to deploying the network. Using the neighbor table, the nodes create parent-child links. Nodes can also utilize links to other one-hop neighbors if it is discovered that the alternative path, in terms of the number of hops, is shorter than the shortest-tree path. From the literature, we know that exploiting the address structure allows for the shortest route to be found with minimal storage and minimal computing cost.

Our algorithm is implemented in the NS-3 network simulator as a new routing protocol. Essentially, a new routing agent, called ITRAgent, was written to implement the protocol. An ITRAgent stores the address for the node to which it is attached, and every node also has an ID which starts at 0. For example, n0 has ID 0; n1 has ID 1, etc. The node's address is specified in a specific way so that the depth of the nodes and the depth of the common ancestor for two nodes can be calculated with the node addresses. In our addressing scheme, the root node has address 0; the first child has address 01 (binary); the second child has address 10 (binary); and the third child has address 11 (binary). The binary addresses, such as 11, correspond to 3 in integer. Therefore, 2 bits represent the address of a child relative to its parent. For example, if a node has an address of 10 (binary), then its first child address is calculated by pre-pending 01 to the parent's address, meaning that the first child's address would be 0110 (6 in integer value). The address of the second child is 1010 (10 in integer,) and for the third child, it is 1110 (14 in integer). Because only 2 bits are used to represent the relative address of a child, a node can have a maximum of 3 children (denoted by 01, 10, and 11), which directly reduces the computational cost and saves the network's energy. The method easily generalizes by simply adding additional bits. This addressing scheme reduces the parent-child links by increasing the tree depth, which leads to fewer sibling, grandparent, or grandchild relationships for a node while increasing the number of neighbors which, in turn, reduces the number of hops while routing. Before routing the data, the algorithm finds the hop count from the source to destination nodes, and if the hop count is reduced by using the ITRAgent, then neighbor is used to transmit data. Otherwise, the data follow the shortest-tree routing approach.

We use an example to clarify the procedure. Suppose that the source node, S, does not share any relationship (grandparent, sibling, or grandchild) with the destination node, D, and that

there is one node, n, which is either a parent or child of destination node D; then, node n is assumed to be a neighbor of source node S by the ITRAgent. In Figure 4.1, node S, the source node, wants to transmit data to destination node D, where the link from S to n is shown as a dotted line which is temporarily created for data transmission by the ITRAgent. Using the shortest tree routing, the route would be S->c->b->a->D, which is 4 hops. Denote the hop count with the shortest-path tree as $H_{SPT} = 4$. With our methodology, the route is S->n->D, which is 2 hops, denoted as the hop count with ITR=H_{ITR}= 2. If $H_{SPT} - H_{ITR} > 0 \Rightarrow H_{ITR} < H_{SPT}$ and the symmetric difference between these two hop counts is given as $H_{SPT} - H_{ITR} = \Delta$, then there exists one neighbor node, which is neither a sibling nor a grandparent/grandchild of the selected nodes for routing, that is chosen for data routing.

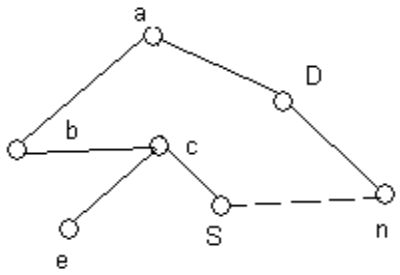
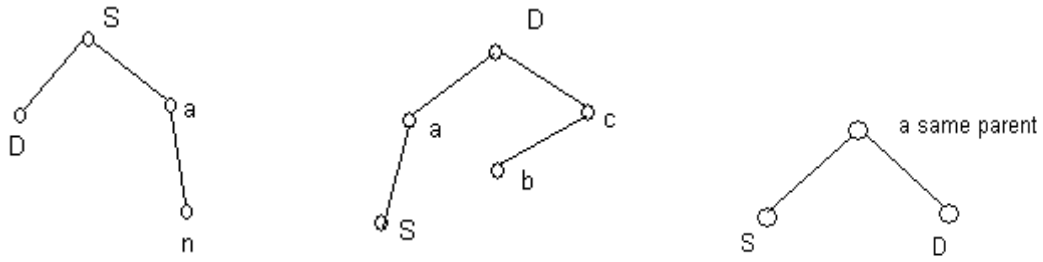


Figure 4.1. Node n is a neighbor node of S because S is neither a sibling nor grandparent/grandchild of D.

Now suppose the source nodes, S, and destination node, D, share any of the parent-child, grandparent-grandchild, or sibling links as shown in Figure 4.2. Then, selecting a neighbor node to transmit data would not reduce the number of hops. The routing scheme is effective only if the source node, S, and destination node, D, have a non-parent and non-child neighbor node, n, with hop count h. If the hop count with the shortest-path tree is greater than the hop being evaluated with the new technique, then n is a one-hop neighbor to the source. In our experiments, we run the algorithm first for intra-cluster routing among the low-end nodes; then, we run it for inter-

cluster routing among the high-end nodes. The algorithm is resilient in the sense that, if a node is disabled, the method can find another path.



(a) (b) (c)

Figure 4.2. Source and destination nodes sharing a) parent-child link, b) grandparent-grandchild link, and c) sibling link.

The steps followed in the ITRAgent are as follows:

Step 1: Check if the neighbor node is a destination node, and if so, then route the data to the neighbor and exit.

Step 2: Check if the destination node is a sibling or grandparent, and if so, then route the data through the existing nodes and exit.

Step 3: Check if the destination node is a grandchild, and if so, then set the shortest-path tree route through that node and exit.

Step 4: Check if the destination node is neither a sibling nor grandparent/grandchild of the nodes chosen for routing the data, but has a neighbor node which is within its radio range and, if chosen for routing and holds the property, $H_{SPT} - H_{ITR} > 0$. If so, then select that node to route the data and exit.

4.5. Enhanced Security

The use of hierarchical routing to forward sensed data is expected to reduce hop counts for data to reach the base station, directly enhancing the security of the deployed sensor network system.

- Using this routing technique, energy utilization will be reduced because energy expanded in the hop count is an important factor. More efficient routes also mean an adversary will have less time to intercept the data, which increases the network security.
- Sybil Attack Defense: In a Sybil attack, a malicious node presents different identities to nodes in the network. In our scheme, node addresses are assigned prior to network deployment, and neighbor tables for each node use these addresses for authentication; therefore, one node cannot act as if it is another node, completely preventing a Sybil attack.
- Wormhole and Sink hole Attacks Defense: The hierarchical routing has intra-cluster routing and inter-cluster routing. Only the low-end sensors send data to their neighbor using their neighbor table, and a parent-child association is established by the cluster head (a high-end node) [6]. With inter-cluster routing, the cluster heads send data to the base station, and no other node can route; therefore, wormhole attack and sinkhole attack cannot occur.
- Hello Flood Attack is avoided as it is a collaborative protocol, using hierarchy to travel data from low-end nodes to high-end node and finally to base station.
- Adjusting transmission power is also helping low-energy nodes reach neighbors up to certain distance. By minimizing the radio range to some threshold level, fewer neighbors are reached, resulting in a reduced breach of information and a lower consumption of energy.

4.6. Evaluation

We use the NS-3 networks simulator [3], [4] to evaluate the developed routing algorithm. The algorithm is implemented using 13, 40, and 121 nodes in a randomly deployed network. In

this network, the node with address 0 is the sink node, and all other nodes are source nodes. This algorithm has been compared with Directed Diffusion [5] to see how it performed. The Directed Diffusion algorithm is built in NS-3. It is the most renowned algorithm and initially uses broadcasting to find the best route to reach to the sink, energy-efficient, and failure-tolerant node. Being available in NS-3 and holding all the attributes with which we are experimenting, the best choice to compare the developed algorithm with it.

4.6.1. Simulation Environment

To facilitate comparison with Directed Diffusion, the same energy model is adopted for the nodes of these two routing protocols, i.e., ITR and Directed Diffusion. The nodes are placed on a square field of 670 m x 670 m. The protocol used for data routing is ITR and Directed Diffusion. Any source could get ready for data transmission. The packet size is 28 bytes. Each connection starts at a time from 0.1 simulated second, and the simulation runs for 50 simulated seconds. The antennas are omni-directional. The data-sharing interface has been initialized similar to the Direct Sequenced Spread Spectrum (DSSS) Wave LAN by Lucent Technologies with a radio frequency of 914 MHz. Each result is averaged over three random network topologies.

4.6.2. Performance Analysis

The performance of the algorithms is evaluated using the NS-3 simulator. The following key issues are addressed.

4.6.2.1. Network Throughput

It is measured as the number of packets received per second at the destination. It is observed that the network throughput for the ITR protocol is better than Directed Diffusion as shown in Figure 4.3. The reason is that the Directed Diffusion uses a Publish/Subscribe API, and

depending upon the scope of the data, DD may use flooding, which leads to reduced throughput for the network.

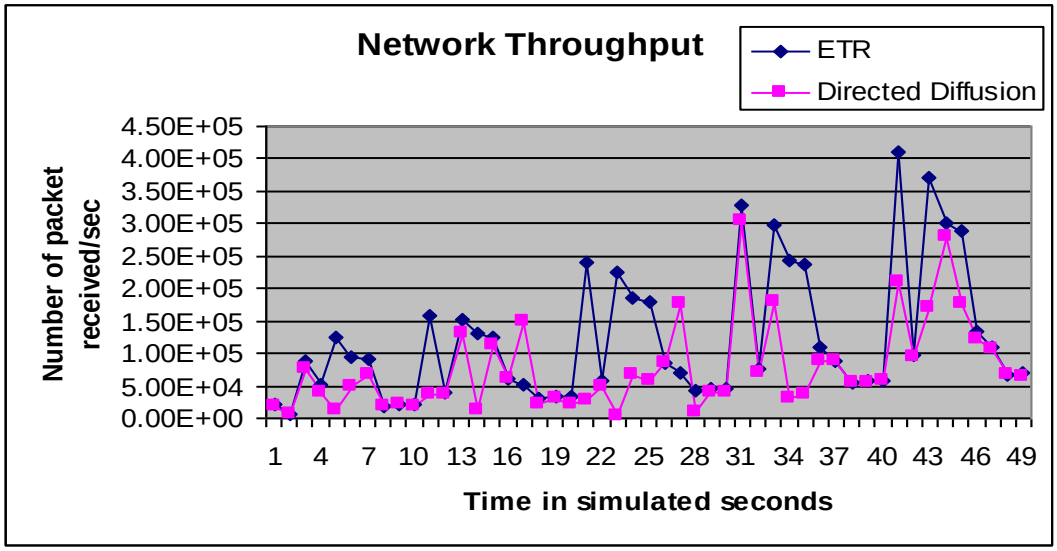


Figure 4.3. Network throughput.

4.6.2.2. Success Rate

It is the total number of packets sent/received in the network. From Figure 4.4, it is clear that, initially, both protocols are sending the same number of packets, but gradually, ITR is outperforming Directed Diffusion. The cumulative sum of sent packets is taken as the average of sent packets with 13-node, 40-node, and 121-node networks, respectively. The reason is that, with the increased tree depth, the ITR is performing better because it could find more neighbor nodes and use fewer hops.

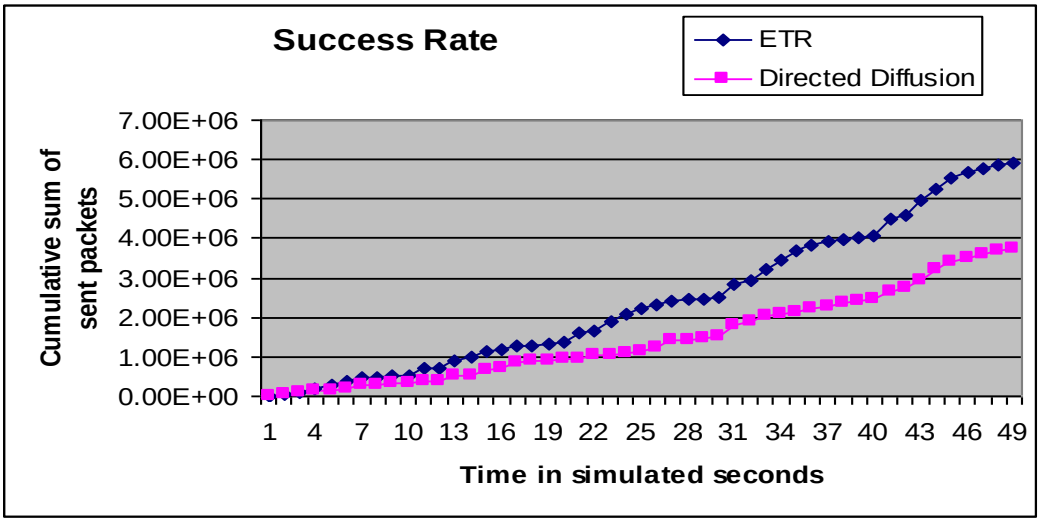


Figure 4.4. Success rate.

4.6.2.3. Packet Generation Rate

It is defined as the number of packets a network transmits in one simulated second. While observing Figure 4.5, when the network size is 13 nodes, the packet-generation rate of Directed Diffusion is higher than ITR, but as the network size grows, the ITR packet-generation rate is better than Directed Diffusion. First, the reason that ITR handles expanded networks in a much better manner than Directed Diffusion is that, as the tree depth increases, the nodes could find more neighboring nodes to route the data to the base station than only using a parent-child link. Second, with Directed Diffusion, the number of dropped packets increased as the network size grows, and the number of dropped packets decreases with the increased number of nodes with the ITR. The number of dropped packets with both protocols is shown in Table 4.1.

Table 4.1. The number of dropped packets.

Network Size	**ITR**	**Directed Diffusion**
13 node	1.71E+02	1.46E+02
40 nodes	6.10E+02	1.05E+04
121 nodes	0.00E+00	2.96E+05

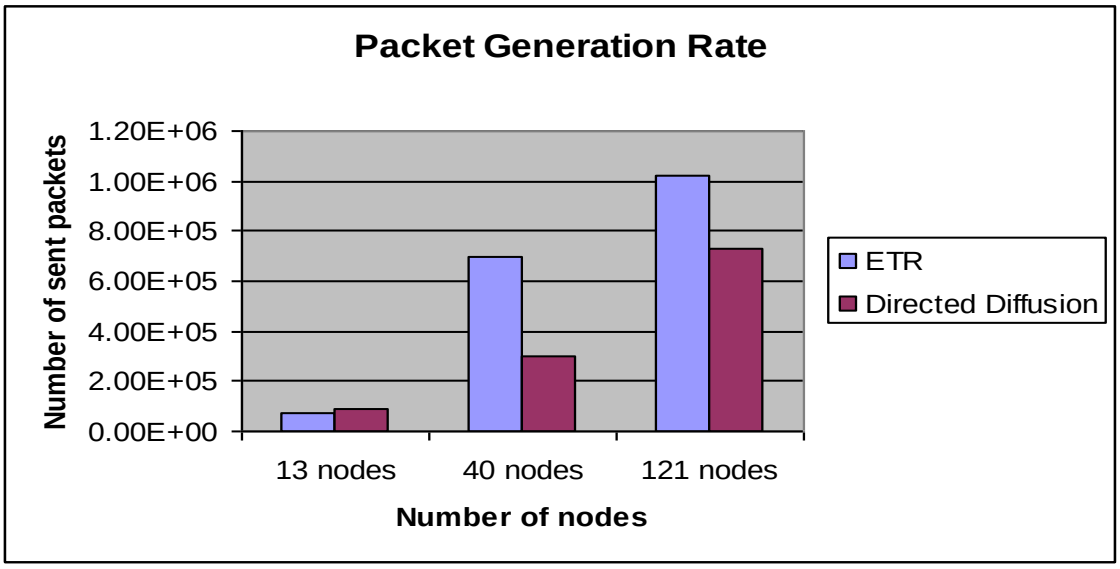


Figure 4.5. Packet generation rate of network.

4.6.2.4. Network Delays

It is a measure of the average end-to-end packet delay from the source to the destination, or we can say that it is the average time taken by a packet starting from the source and the time to reach to the destination. Furthermore, the end-to-end delays also increase the propagation delay and the queuing delay to the packets. From Figure 4.6, it is clear that routing packets with ITR takes less time than Directed Diffusion. As the network is expanding, the average end-to-end delay is also increasing in both protocols because the number of intermediate nodes is increasing, but we observe that the average end-to-end delay with ITR is much less than with Directed Diffusion.

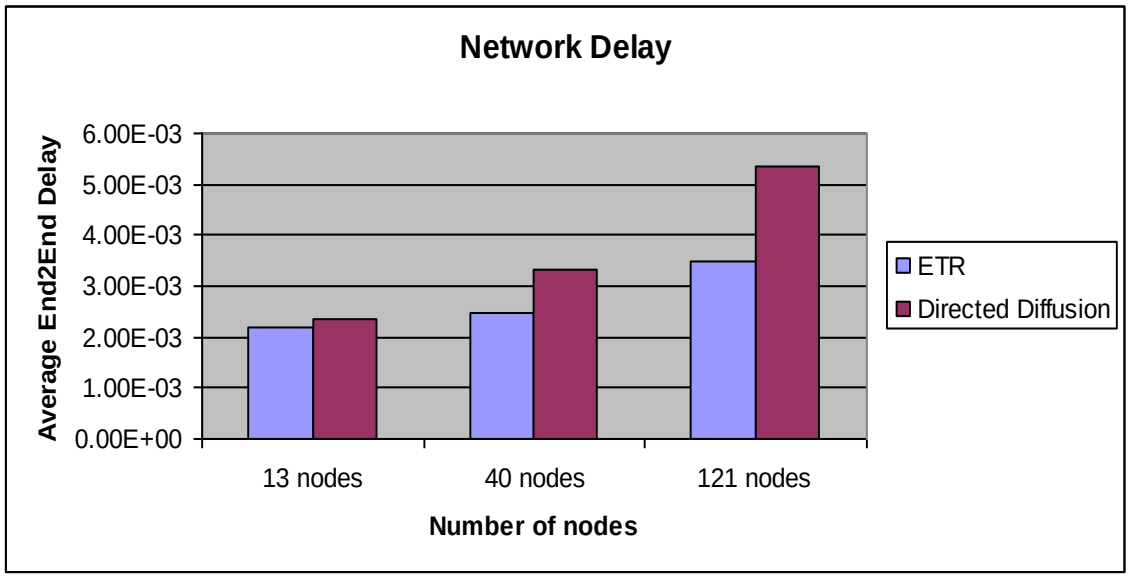


Figure 4.6. Network delays.

From the above results, we learned that ITR is making the data reach the base station securely, taking much less time and using less energy than Directed Diffusion.

4.7. Conclusion

The empirical results show that ITR is outperforming Directed Diffusion. As the network size grows, the ITR packet-generation rate improves, and the number of dropped packets becomes negligible, whereas with the network expansion, the number of dropped packets increases in Directed Diffusion, and the packet-generation rate is lower than ITR. The network throughput and success rate is better with ITR than Directed Diffusion, and the network's average end-to-end delay is less in ITR than with Directed Diffusion. We learned that the hierarchical network topology, if used with our indexing scheme, can reduce the hop count, leading to fewer chances for the data to be intercepted by adversaries. This topology, in turn, makes the heterogeneous sensor networks more secure and robust to vulnerable points while also minimizing the use of energy by nodes for transmitting data to their respective cluster heads.

4.8. Future Work

While routing the data, if a network is divided into high-performance clusters, then routing performance could be increased. In the future, we want to work on different clustering techniques which will have reduced hop counts, further leading to a secure and low-energy consuming ITR protocol.

4.9. References

[1] Ai, Chunyu, Hou, Hailong, Li, Yingshu, and Beyah, Raheem, "Authentic Delay Bounded Event Detection in Heterogeneous Wireless Sensor Networks," Journal Ad Hoc Networks, Vol. 7, Issue 3, pages 599-613, May 2009.

[2] Du, Xiaojiang, Guizani, Mohsen, Xiao, Yang, and Chen, Hsiao-Hwa, "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks," IEEE Transactions on Wireless Communications, Vol. 6, Issue 9, pages 3395-3401, Sept 2007.

[3] Fall, Kevin and Varadhan, Kannan, http://www.isi.edu/nsnam/ns/doc/ns_doc, last accessed on Feb., 2008.

[4] Henderson, Tom, Lacage, Mathieu, http://www.nsnam.org/, last accessed on July, 2008.

[5] Intanagonwiwat, Chalermek, Govindan, Ramesh, and Estrin, Deborah, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proceedings of ACM Mobicom'00, pages 56-67, Aug 2000.

[6] Madria, Sanjay, and Yin, Jian, "SeRWA: A Secure Routing Protocol Against Wormhole Attacks in Sensor Networks," Ad Hoc Networks, Vol. 7, Issue 6, pages 1051-1063, Aug 2009.

[7] Qiu ,Wanzhi, Skafidas, Efstratios, and Hao, Peng, "Enhanced Tree Routing for Wireless Sensor Networks," Ad Hoc Networks, Vol. 3, pages 638-650, July 2008.

[8] Younis, Ossama, Krunz, Marwan, and Ramasubramanian, Srinivasan, “Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges,” Network, IEEE, Vol. 20, Issue 3, pages 20-25, May-June 2006.

CHAPTER 5. PAPER 2: KEY-MANAGEMENT SCHEME FOR ROUTING IN CLUSTERED HETEROGENEOUS SENSOR NETWORKS

Kanwalinderjit Kaur Gagneja, Kendall E. Nygard
Dept. of Computer Science
North Dakota State University
Fargo, ND, USA

5.1. Abstract

Adding security using pair-wise key establishment in heterogeneous sensor networks is a very challenging task because sensor networks are usually resource crunched and generally deployed in hostile environments. This paper puts forward an effective and efficient security scheme for launching anonymity in clustered, heterogeneous sensor networks. The clustered, heterogeneous sensor network has two types of nodes: high-end nodes that work as cluster heads and low-end nodes. The clusters are initially formed by using Voronoi diagrams, and then, some node adjustment is done in the clusters using the Tabu search. The secure scheme uses a unique technique to conceal its true identity (ID). Once the sensor nodes are deployed, the nodes set up their neighbor table and share their secret key with neighbors while ensuring that their true identity is not disclosed and that the communication is secured. The empirical performance evaluations support the given approach's efficiency and effectiveness.

5.2. Keywords

Heterogeneous sensor networks, Routing, Security, Key management, and Clustering

5.3. Introduction

For applications such as military operations, border surveillance, and security monitoring, it is essential to secure communications in the sensor networks. Thus, to make communication reliable and trustworthy, data sent through a wireless medium should be protected.

It is hard to embed security in sensor nets mainly because sensor nodes are generally resource crunched with small memory, low computation capability, and little power. Consequently, using public-key cryptography puts a lot of burden on these tiny nodes. Another difficulty is that these nodes usually work in an unattended environment and could be exposed to capturing, which may lead to the potential disclosure of information. Therefore, the secured route should be resilient to such captured nodes. Finally, an eavesdropper could attack any sensor node and could use gathered information to compromise other nodes in the network and could promote to disparity between the risk and the security in sensor nets.

A number of key-management schemes are proposed for sensor nets designed for networks where all sensor nodes have similar capabilities. The paper presents an efficient key-management scheme designed for networks with two types of nodes: high-end nodes and low-end nodes organized into clusters. The paper is organized as follows. Section 5.4 describes the Related Work. In Section 5.5, the Clustering Approach is discussed. Section 5.6 discusses the Addressing Scheme. In Section 5.7V, the key-management scheme is discussed. Performance analysis is discussed in Section 5.8. Section 5.9 presents the Conclusions.

5.4. Related Work

Eschenauer and Gligor first proposed a pre-distribution key management necessitating a significant amount of memory to store keys [3]. The authors of papers [1] and [9] improved the pre-distribution key-management technique by permitting pair-wise keys to be generated once the network is set up and the nodes are sharing keys from the given key space. Another key-management technique, LEAP designed by Zhu et al., assumed that, once the network is deployed, it is secure for a short period of time [17]. For the boot-strap key management network uses a preloaded global key.

The SBK scheme was proposed by Liu et al. [10] and utilizes the key-space representations defined in [1] and [9]. With this technique, the nodes store a set of parameters that are used to set up pair-wise keys among neighbor nodes, where each node creates a key space. Furthermore, neighbor nodes use key space to generate pairwise keys. In SBK, *a priori* deployment awareness is not required. However, node still attains very good connectivity among neighbor nodes. On the other hand, the node that gets elected as a service node has to do repeated computations, thus die early once keys are setup.

The clustered heterogeneous sensor network (HSN) topology is used in this research. With the clustered HSN topology, there are a few high-end sensor nodes and many low-end sensor nodes [16]. High-end nodes have higher frequency, calculation capacity, memory, and energy levels than low-end nodes. One cluster has one high-end node that acts as a cluster head (CH) and has many low-end nodes. In the presented scheme, the nodes are deployed randomly in the given area of interest and do not have *a priori* knowledge about node deployment.

5.5. Clustering Approach

Initially, the given area of interest is partitioned into Voronoi diagrams. Voronoi diagrams use Euclidean distance to partition the polygon in a 2D plane, where one high-end node forms the cluster as shown in Figure 5.1 and there are a number of low-end nodes [12]. The data routing in sensor nets is generally multi-hopped instead of distance [15]. Therefore, to route data, it is important to use hop counts. Thus, a new Tabu search heuristic was applied to adjust some of the nodes on, or near, boundaries to be reassigned to adjacent clusters.

The Tabu search is a meta-heuristic based on neighborhood search, where adjustment of nodes takes place until a stopping criterion is met [2], [7]. The Tabu search locates such areas in the search space that are left undiscovered by typical search procedures. After every iteration, the

Tabu search changes the arrangement of the neighborhood. It also keeps a list, known as a Tabu list, that does not allow the reversal of neighborhood search moves.

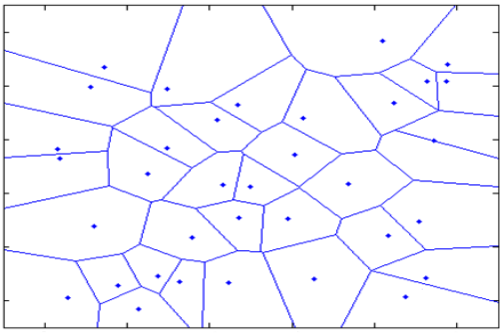


Figure 5.1. Voronoi clusters with just one high-end node acting as the cluster head.

In this topology, the Tabu search heuristic performs an extensive search to re-allocate the low-end nodes to adjacent clusters so that better clusters can be formed [5], [6]. A few nodes, those lying on or near the boundaries, are re-allocated to adjacent cluster(s) and are freed from the current cluster. For example, if low-end sensor node *x* is currently assigned to cluster *a*, then after re-allocation, it is freed from cluster *a*. Mathematically, one could say that, if $N_{xa}=1$, node *x* belongs to cluster *a* and that, if $N_{xb}=0$, node *x* is not in cluster *b*. After re-allocation, when node *x* is re-allocated to cluster *b*, then $N_{xa}=0$ means that node *x* is not in cluster *a*, and $N_{xb}=1$ means that node *x* now belongs to cluster *b*, where $a \neq b$, $a=b=1, 2,\ldots C$ clusters and $x=1, 2,\ldots n$ number of low-end sensor nodes.

In the intensification stage of the Tabu search, the hop count for nodes on or near the already established Voronoi cluster boundaries is measured from the adjacent cluster heads. Assign the node to the cluster that minimizes its hop count along the shortest-path tree. Cyclic transfer moves are made when more than two clusters are involved with node reassignment. These moves maintain the number of nodes in each cluster. Finally, pair-wise exchanges of nodes between two clusters are considered. The intensification phase stops when no improving

solutions are found in *n* iterations. The detailed implementation for the Tabu-Voronoi scheme is available in paper [6].

5.6. Addressing Scheme

Each node is allotted a binary address by the base station after deploying the network. Each node maintains a neighbor table which stores its node-ID, and the parent-child and neighbor links. The addressing scheme is based on the shortest-path tree, and nodes use the shortest-path tree to forward the data. The root node is the sink or the BS (Base Station) with a binary address of 00, and no node can have more than three children. Therefore, addresses are computed in a specific manner. Because a node cannot have more than 3 children, the root has three children with addresses of 01, 10, and 11 (all in binary). The integer values of the above-mentioned addresses are 1, 2, and 3, respectively. Addressing at the second level of this tree is explained with an example. To compute addresses for children of a node with address 10, prefix 01 to its address (10 in binary), making the address with the first child's address having integer value 6. To compute the second child's address, similarly, 10 is added to its address and gives 1010, having integer value of 10. The third and last child's address prefix, 11, becomes 1110, having an integer value of 14.

Prefixing 2 bits easily gives the children's addresses. Hence, it reduces computation [4]. A limitation about the number of offspring leads the tree to grow to a depth that reduces parent-child links, thus increasing the neighbor-finding probability, signifying fewer hops. Because the neighbor node is chosen to forward the information, it decreases the hop count instead of keeping the same hop count as in the spanning tree.

5.7. Securing the Network

In this section, we describe the key-management scheme. The network setup and addressing schemes have already been described in Sections 5.5 and 5.6, respectively.

5.7.1. The Network Security Model

A large Tabu-Voronoi, clustered, Heterogeneous Sensor Network has been considered. It is also assumed that the network does not have prior knowledge about the location of sensor nodes in the network because the sensor nodes are deployed randomly in the given interest area. The clustered HSN has numerous low-end sensor nodes, which are small in size, are low-cost, and have parameters stored in their memory as shown in Table 5.1. Every low-end node is affiliated with only one cluster, and gathered data are communicated to their corresponding high-end node that is a cluster head. These low-end sensor nodes are not tamper resistant. Once they are captured by eavesdroppers, they can acquire all the stored data on these tiny, low-end nodes.

Two security models have been considered. The first security model assumes that, immediately after deployment, the network is secure for a short amount of time while the bootstrapping phase is establishing pair-wise keys for sensor nodes in the network. The second model presents a secure scheme to set up pair-wise keys for nodes in the clustered HSN.

5.7.2. The Key-Management Scheme

This subsection presents the key-management scheme planned for the clustered heterogeneous sensor networks. Before deploying the network, high-end sensor nodes are loaded with two algorithms. Some parameters used in these algorithms are shown in Table 5.1. The ∂_1 represents the maximum number of low-end nodes that could be present in a cluster. The ∂_2 represents the maximum number of low-end nodes associated with another low-end node in the

cluster, and $\partial_2 << \partial_1$. The t_m represents the maximum bootstrapping time limit. ID is the node's unique identification number, and t_s is the transmission signal-strength radius of one sensor node.

The area of interest is a 2D plane, so the deployed nodes know of their location, i.e., to which cluster they belong, what their neighbor nodes are, who their parents and children are, etc. The two high-end neighbor sensor nodes can transmit data to one another if they are within the transmission signal strength of each other [13]. The transmission signal strength of one sensor node is the radius, t_s, it covers around itself [5].

Table 5.1. Parameters.

$\partial 1$	The maximum number of low-end nodes in a cluster
$\partial 2$	The maximum number of low-end nodes associated with another low-end node in the cluster, and $\partial 2 << \partial 1$
t_m	Maximum waiting time
ID	Unique sensor node ID
t_s	Transmission signal-strength radius of one sensor node

The security scheme is an asymmetric, post-distribution, key-management scheme. Originally, the keys are set up so that each sensor node shares a key with the *n* nodes closest to it. Attribute *n* is set up based on a sensor node's available memory. The first algorithm is executed for key space generation.

In the addressing scheme, as explained in Section 5.6, each sensor node is assigned one unique binary address known as the node-ID. Two nodes that want to share data have to establish a shared key for communication. The shared key is a combination of the node address (node-ID) plus the level (tree level) of any node in the network. Because the node-ID of all neighbor nodes is stored in the neighbor table, the level is computed by counting the number of bits in the node-ID divided by 2. For example, a node with the binary address 0110 is counted by the number of

bits as 4-bits. Dividing it by 2 gives the level of this node in the shortest-path tree, which is 2. Thus, nodes at the same level in the shortest-path tree share this level as part of the key. This shared key among neighbor nodes is known as a direct key. However, the base station assigns a shared key to the high-end neighbor nodes which have dissimilar level numbers. This kind of shared key establishment is known as an indirect key. Thus, $K_{x,y,p}$ denotes the shared key for two nodes *x, y,* and *p* represents the level number common to both *x, y* nodes or is allotted by respective high-end node if the level is not same for those two low-end nodes. An analogous method is applied for setting up shared keys between high-end nodes; however, the base station allocates the indirect keys to different-leveled high-end nodes.

This methodology is secure because the adversary would not be able to implant its own node when the shortest-path tree addressing scheme is followed, which is closed-knit, limiting offspring to three. Another reason the protocol is secure is because the adversary would not be able to figure out the level number if the address is not known.

In the literature [1], [8], [9], [14], [17], the shared key is chosen from the already-stored key pool in the sensor node. Because this process utilizes memory to keep direct/indirect keys, do computations, and use processing power to pick one key from the key pool. Thus, the security technique in sensor networks must be a technique characterized by limited calculations and smaller memory requirements. With the given approach, setting up the shared key is minimal because each shared key for a specific node is related to the node-ID. Thus, this technique does not waste memory to store the key pool, so it uses less computation, requiring less network energy.

As soon as the shared keys are assigned using an asymmetric post-distribution setup approach, nodes do two-way, hand-shaking for data exchange during the intra-cluster routing.

For example, node *x* could get ready to route data in case some event occurs in the network. Before routing data to node *y*, both nodes do two-way hand-shaking. Thus, *x* begins by sending a hello packet to *y*: *[x]* $K_{x,y,p}$ *+MAC*($K_{x,y,p}$, *). The *MAC*($K_{x,y,p}$, *) signifies the Message Authentication Code (MCA) and is engineered at network-layer level having shared key $K_{x,y,p}$. Afterwards, *y* replies to *x*: *[*$S_{y,x,p}$ *, y]* $K_{x,y,p}$ *+MAC*($S_{y,x,p}$, *), and $S_{y,x,p}$ is a shared key established by *y* and is utilized for the auxiliary data exchange between *x* and *y*.

Essentially, the data packet has two parts: the packet header and data. The header part has the packet-ID (*uid_*), packet type (*ptype_*), simulated packet size (*size_*), time stamp (*ts_*), and Message Authentication Code (*MAC*($K_{x,y,p}$, *)). The *uid_* is the unique identification number set for the packet to observe its route. The MAC part records any change to the packet since its inception. If node *x* times out and does not receive acknowledgement from the recipient, then re-transmission of the same packet takes place. The high-end nodes similarly initiate secure transmission to the base station.

The Sensor-node-key-setup (Table 5.2.) and Distribution-of-keys algorithms (Table 5.3.) are loaded to each node. As soon as a node initiates the bootstrapping phase, it establishes a key space as mentioned in the previous section. In the algorithm, the bootstrapping is shown on lines 3 and 4, and the nodes save these key in its neighbor table. If the direct key is setup, means the nodes are at the same level; that is, either the nodes are high-end only or low-end only. Otherwise, indirect keys are established if the nodes are at different levels. Once keys are established, the sensor nodes go into the distribution phase.

The distribution of the key method works in such a way that, as soon as a sensor node executes the Distribute-Keys method to distribute the keys, the method will transmit its ID to all nodes in its range (t_s) as represented in line 2 of the algorithm. The sensor then keeps accepting

requests for the keys from its neighbor sensors within its range. The shared keys are appended in the number-of-keys array (line 6). Then, the node starts sending each of the requesting nodes the ID, shared key, and MAC different for each (line 7). The messages are secured by encrypting the keying information requested by the neighboring sensor node.

Table 5.2. Sensor-node-key-setup algorithm.

```
1: method sensor-node-key-setup (∂1, ∂2, tm, ts)
2: if (level= =tree level) // generate key space for neighbor nodes, store one for self
3:       keys ←genDirectKey()
4:  else keys ← genIndirectKey()
5:       m ← random(0 < tm / 2) // compute random wait time
6:       while (m > 0) && (tss ≤ ts) do
7:            listen for transmissions from neighboring nodes
8:                if TRANSMISSION heard
9:                Request-Keying-Info //get keying info from other node
10:               end if
11:      finish-waiting-time (m)
12:      end while
13:      Distribute-Keys()
14: end method
```

Table 5.3. Distribution-of-keys algorithm.

```
1:method Distribute-Keys(∂1||∂2, ID, keys, ts)
2:   TRANSMIT(ID, ts) // transmit to nodes within range
3:   while (number-of-keys < ∂1-1||∂2) do
4:       if (receive ==Request-Keying-Info(ID))
5:           Share-Key //  share a key
6:            number-of-keys ← number-of-key∪ Share-key
7:            send(ID, Share-Key, MAC)
8:       end if
9:   end while
10: end method
```

5.8. Performance Evaluation

The performance is analyzed with respect to the communication overhead, memory used, and connectivity.

5.8.1. Communication Overhead

The communication overhead for the scheme presented in this paper is compared with the SBK scheme [11]. The complete communication path followed by any two sensor nodes in the scheme is shown in Figure 5.2. A sensor node transmits a Hello message to all the neighbor nodes within its transmission range. The sensor nodes that receive this Hello message then send a request for keying information. Finally, the sensor node sends the keying information to the requesting sensor, provided that the request is recognized.

The following equation (5.1) represents the communication overhead for the complete network of our scheme:

$$X(h+ \partial_1) + Y(h+ \partial_2) + h_{op}X \partial_1 + n \qquad (5.1)$$

The X represents the number of high-end nodes in the given network; h represents the number of Hello messages generated by the sensor nodes; and ∂_1 represents the number of sensor nodes requesting keying information from the high-end sensor nodes. The low-end sensor nodes also have similar communication overheads. Y represents the number of low-end nodes; h is the Hello message sent; and ∂_2 represents the maximum number of connected, low-end nodes in the cluster. The total number of keying-information requests from low-end nodes to high-end nodes is $X\partial_1$. Because communication from a low-end sensor to a high-end sensor could be multi-hopped, these transmissions are forwarded, on average, h_{op} hops. Moreover, each low-end sensor node sets up pair-wise keys with *n* neighboring nodes.

The SBK technique has the following communication overhead:

$$p(h+ \partial_2) + h_{op}\, \partial_1 + n \qquad (5.2)$$

In SBK, the network is homogeneous; that is, all nodes have similar capabilities, and *p* represents the number of elected nodes to serve as cluster heads. All additional variables of equation (5.2) represent similar values as given in equation (5.1).

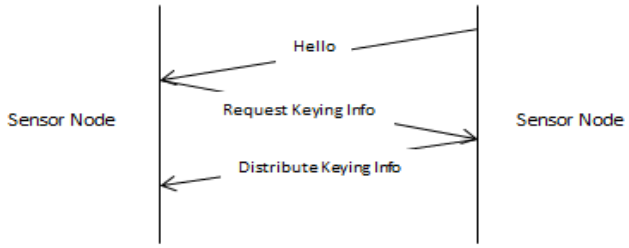


Figure 5.2. Communication path followed by any two nodes.

In the presented scheme, only fewer Hello messages are sent in the complete network because high-end nodes have a higher transmission range. The NS-3 network simulator is used for simulation. With the simulations, there are 800 nodes randomly distributed on a 650 m x 650 m area of interest. There are 80 high-end nodes and 720 low-end nodes to cover the complete area of interest. The transmission range of low-end nodes is set at 50 meters, and the transmission range of high-end nodes is set to 250 meters, which is 5 times that of the low-end nodes. Because the SBK scheme only utilizes low-end nodes, it would require 180 nodes to serve as cluster heads to distribute keying information to other low-end nodes in the given network. However, this scheme only requires 80 high-end nodes, which implies that X in equation (5.1) is much smaller than p in equation (5.2). Thus, the scheme is generating considerably fewer Hello packets than the SBK approach and, hence, has fewer communication overheads.

5.8.2. Storage and Connectivity

Table 5.4. represents the comparison of the given scheme with the SBK, E-G, and LEAP approaches with respect to the number of keys stored. The *n* represents the number of neighbor nodes for the given high-end node, and n_1 represents the keys that are stored on the nodes before network deployment.

Table 5.4. Number of keys.

Schemes	**keys**
SBK	*n*
LEAP	*n+3*
E-G	n_1
presented	*n*

Table 5.5. shows the degree of connectivity once the keys are established. For example, the number of keys (*n)* is 30. The presented scheme, LEAP, and SBK are all capable of setting up pair-wise keys with complete connectivity among the neighboring nodes. However, the E-G scheme has to store 225 keys in the memory and is capable of setting up pairwise keys with 90% connectivity.

Table 5.5. Degree of connectivity.

Schemes	**Number of keys**	**Degree of connectivity**
SBK	30	100%
LEAP	33	100%
E-G	225	90%
presented	30	100%

Finally, note that the presented scheme requires only *(n+1)* addition to set up pairwise keys with the neighbor nodes. The SBK scheme needs *(n+1)* prefabricated multiplications, whereas the matrix-based model requires *xy(n-1)* prefabricated multiplications to establish a matrix with *x* rows, *y* columns, and *n* neighbor nodes. Thus, this scheme needs less computation than the SBK scheme.

5.9. Conclusion

Setting up pairwise keys in heterogeneous sensor networks is very difficult. The existing pairwise key-establishment schemes are, generally, for a homogeneous sensor network. Many researchers have shown that clustered, heterogeneous sensor networks perform better. In the presented scheme, the clusters are initially formed by Voronoi diagrams; then, a new Tabu heuristic is applied to adjust the low-end nodes lying on or near the cluster boundaries. The presented scheme compares favorably with some existing schemes because it has less computation overhead, uses little memory to store its keys, and shows a high degree of connectivity with neighboring nodes. Therefore, this paper presents an effective and efficient key-management scheme that is specifically designed for clustered, heterogeneous sensor networks.

5.10. Acknowledgements

The research was done with grants W911NF-07-1-0250 and W911NF-08-1-0334 that were funded by the U.S. Army Research Office (ARO).

5.11. References

[1] Du, Wenliang, Deng Jing, Han Yunghsiang S., and Varshney Pramod K., "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," Proc. 10th ACM Conference on Computer and Communication Security (CCS '03), pages 42-51, 2003.

[2] El Rhazi, Abdelmorhit, and Pierre, Samuel, "A Tabu Search Algorithm for Cluster Building in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 8, Issue 4, pages 433-444, April 2009.

[3] Eschenauer, Laurent and Gligor, Virgil D. “A Key-Management Scheme for Distributed Sensor Networks,” Proc. Ninth ACM Conference on Computer and Communication Security (CCS ’02), pages 41-47, November 18-22, 2002.

[4] Gagneja, Kanwalinderjit K., Du, Xiaojiang, and Nygard, Kendall, “Enhanced Routing in Heterogeneous Sensor Networks,” ComputationWorld '09, pages 569-574, Nov. 15-20, 2009.

[5] Gagneja, Kanwalinderjit K., and Nygard, Kendall, “Energy Efficient Variant of Improved Tree Routing Protocol for HSNs,” IEEE CISSE 2010, University of Bridgeport, Bridgeport, CT, USA, Dec., 2010.

[6] Gagneja, Kanwalinderjit K., and Nygard, Kendall, “Adjusted Voronoi Clusters in Two-Tier Sensor Networks,” CATA’11, New Orleans, LA, USA, March 2011.

[7] Glover, Fred, and Laguna, Manuel, *Tabu Search*, Kluwer Academic Publishers, 1997.

[8] Karlof, Chris, and Wagner, David, “Secure Routing in Sensor Networks: Attacks and Countermeasures,” Proc. IEEE 1st Int. Workshop Sensor Network Protocols Applications, pages 113-127, May 2003.

[9] Liu, Donggang, and Ning, Peng, “Establishing Pairwise Keys in Distributed Sensor Networks,” Proc. 10th ACM Conf. Computer and Comm. Security (CCS ’03), 2003.

[10] Liu, Fang, Cheng, Xiuzhen, Ma, Liran, and Xing, Kai, “SBK: A Self-Configuring Framework for Bootstrapping Keys in Sensor Networks,” IEEE Transactions on Mobile Computing, Vol. 7, Issue 7, pages 858-868, July 2008.

[11] Misra, Satyajayant, and Xue, Guoliang, “SAS: A Simple Anonymity Scheme for Clustered Wireless Sensor Networks,” Proceeding of IEEE ICC, pages 3414-3419, 2006.

[12] Navarro, Alejandro, and Rudnick, Hugh, "Large-Scale Distribution Planning—Part II: Macro-Optimization with Voronoi's Diagram and Tabu Search," IEEE Transactions on Power Systems, Vol. 24, Issue 2, pages 752-758, May 2009.

[13] Stojmenovic, Ivan, *Handbook of Sensor Networks: Algorithms and Architectures*, Wiley Series on Parallel and Distributed Computing, 2005.

[14] Sun, Kun, Peng, Pai, Ning, Peng, and Wang, Cliff, "Secure Distributed Cluster Formation in Wireless Sensor Networks," Proceedings of the 22nd Annual Computer Security Applications Conference, pages 131-140, 2006.

[15] Yang, Yi, Wang, Xinran, Zhu, Sencun, and Cao, Guohong, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," ACM Transaction on Information and System Security, Vol. 11, Issue 4, pages 1-43, July 2008.

[16] Younis, Ossama, Krunz, Marwan, and Ramasubramanian, Srinivasan, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," Network, IEEE, Vol. 20, Issue 3, pages 20-25, May-June 2006.

[17] Zhu, Sencun, Setia, Sanjeev, and Jajodia Sushil, "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ACM Transactions on Sensor Networks, Vol. 2, pages 500-528, Nov. 2006.

CHAPTER 6. PAPER 3: USING TABU-VORONOI CLUSTERING HEURISTICS WITH KEY-MANAGEMENT SCHEME FOR HETEROGENEOUS SENSOR NETWORKS

Kanwalinderjit Gagneja, Kendall Nygard
Dept. of Computer Science
North Dakota State University
Fargo, ND, USA

Navninderjit Singh
Dept. of Management
RIMT
Gobindgarh, Punjab, India

6.1. Abstract

Clustering, routing, and security are essential for better performance of any sensor network. Providing built-in security for routing algorithms in sensor networks is essential because some of these sensor networks have applications in hostile environments. Researchers work either on a routing algorithm or on security. However, the security should be embedded to the routing scheme's design. In this paper, we initially divide the given area of interest into Voronoi clusters and then apply a new Tabu heuristic to form more stable clusters. When some event occurs, a new, efficient, key-management technique is applied with the improved tree-routing algorithm for data routing in heterogeneous sensor networks. The simulation results show that this scheme offers security and uses less computation, affording substantial savings in memory requirements. Our simulation results show that when Tabu-Voronoi clustering and the secure-routing scheme are applied together, it performs better than some existing algorithms. This approach shows higher throughput, fewer network delays, and less energy utilization.

6.2. Keywords

Heterogeneous sensor networks, Routing, Security, Key management, and Clustering

6.3. Introduction and Related Work

Heterogeneous sensor networks (HSNs) with two-tier topology work best if clustering is applied. Various researchers have shown that clustering improves the sensor network's performance in terms of energy usage, coverage, etc. Hence, in this approach, node clusters using a combination of Voronoi and Tabu techniques are formed.

When some event occurs in the network, the data are routed from low-end nodes to the base station. The Improved Tree Routing (ITR) approach is used for data routing. A complete description of the ITR approach is available in [4]. Routed data devoid of security are susceptible to various attacks. Hence, for data security, it is important to embed security in the routing algorithms' design [1]. For securing the sensor networks, the key-management technique is preferred. To help manage resource-crunched sensor nodes, developers have invented various random, pair-wise key pre-distribution schemes, such as E-G [3]. The LEAP protocol stores distinct keys among various sensor nodes and the base station [10]. Liu et al .proposed the SBK approaches [8]. The neighbor nodes make use of key space to produce pairwise keys.

In this approach, the two-tier HSNs have two types of nodes, the low-end (LE) and high-end (HE) nodes, and high-end nodes work as cluster heads (CH). Initially, we partition the given area into Voronoi clusters. Voronoi diagrams generate polygonal clusters using Euclidian distance [9]. Because sensor network routing is multi-hopped instead of distance based, we apply the Tabu search [2], [9] to adjust some nodes in the Voronoi clusters.

Therefore, data routing is multi-hopped instead of distance based. In case of some event in the network, the low-end nodes gather data and forward data to their CH using the two-way, hand-shaking, secure ITR route. Empirical evaluations show the described algorithm's performance with respect to the delivery ratio, end-to-end delays, and energy usage.

The paper organization is as follows: Section 6.4 describes the implementation specifications of clustering with the new Tabu heuristic approach. Section 6.5 describes the ITR routing scheme. Section 6.6 discusses the implemented security scheme on the ITR.Simulation setting and result analysis is discussed in Section 6.7. Section 6.8 presents the conclusions.

6.4. Clustering Approach

Initially, a given 2D area of interest is divided into Voronoi diagrams using Euclidean distance. The sensor nets use multi hopping for data routing instead of distance; thus, it is more appropriate to use hop counts. Hence, to make hop-count based clusters, we use a new Tabu search heuristic to reassign nodes to different clusters. Tabu search is a meta-strategy based on the neighborhood search in which iterative moves (adjustments) among solutions are made until a stopping criterion is satisfied. The Tabu search maintains a Tabu list that prevents the reversal of neighborhood search moves.

In this approach, the Tabu search heuristic does an extensive search to reassign the sensor nodes to make better clusters. Specifically, if some sensor node *a* belongs to cluster *q*, reassignment frees it from cluster *q*. Mathematically, it could be presented as $N_{aq}=1$ (node *a* in cluster *q*) and $N_{ar}=0$ (node *a* not in cluster *r*). When node *a* is reassigned to cluster *r*, then

$N_{aq} = 0$ (node a not in cluster q) and $N_{ar}=1$ (node a in cluster r), where $r \neq q$ q, $r=1, 2,\ldots C$, and $a=1, 2,\ldots m$.

During the intensification stage**,** three different types of node reassignments take place. The very first node reassignment is for those specific nodes on or close to the Voronoi cluster boundaries. If the hop counts of the node from the adjacent cluster head is less, those nodes are assigned to an adjacent cluster. Second-cyclic transfer moves are performed on adjacent clusters where each cluster exchanges a number of nodes but keeps the same number of nodes as the original. Last, pair-wise exchanges among two adjacent clusters are performed. The intensification stage stops in n recursions when no refining solutions are found.

The Tabu list helps avoid cycles and return to the same solution. The moves set as Tabu in the Tabu list are not considered for further solutions, but in some situations, this could be a good move. Hence, if the Tabu list satisfies the aspiration condition**,** that move is considered for the next solution. In the diversification phase, each node is freed from its present cluster at its respective iteration.

6.4.1. System Assumptions

Every node, either low end or high end, has a unique binary node-ID. Because node-IDs are crucial to this protocol's functioning, they are identifiable only by their neighbors, the CH and the BS.

- The local position of the low-end nodes is identifiable to the CHs in their respective cluster and to the BS.
- CSThresh1 and CSThresh2 are the carrier-sense threshold settings for all the low-end nodes and all CHs, respectively. RXThresh1 and RXThresh2 are the receiving

threshold settings for all the low-end nodes and all CHs, respectively. The important point to note here is that the carrier-sense threshold setting, CSThresh_, should always be set lower than the receiving threshold setting, RXThresh_; otherwise, the data could not be sent up the physical layer.

- The BS has infinite resources, and CHs are temper resilient.

Following is the acceptably model that makes sure the given area of interest is fully covered by the ITR and is connecting all CHs with the low-end nodes in their respective clusters. The following notations are defined to analyze the optimality of this approach:

Defined Sets:

S ={1,…,m}, the set of LE sensor nodes

C = {1,…,.p}, the set of clusters

Indexes of Sensors:

i= 1..m for the low-end sensor nodes

j= 1..p for cluster-head sensor nodes

Defined Input Parameters:

A_i = The A_i are the LE sensor nodes in the given network.

B_i = The B_i are the HE sensor nodes in the given network, where one cluster has only one HE sensor node.

e_n = energy used by the given network

e_r = remaining energy of the given network at any time

Decision Variables:

L_{ij} = 1, if given, low-end node *i* could reach another low-end node *j* in one hop, else 0

H_{ij} = 1, if given, high-end node *i* could reach another high-end node *j* in one hop, else 0

X_{ij} = 1, if given LE node *i* is connected to HE *j* in ITR, else 0

Y_{ij} = 1, if given HE node *i* is connected to HE node *j* in ITR, else 0

P_{ni} = Power/Energy of the given network, whereas it is given as the total energies of the LE and HE sensor nodes.

$P_{ni} = \log (1+ e_n / e_r)$

P_{ri} = the residual energy of the given network

Figure 6.1 represents the logarithmic nature of a network's total energies vs. the residual energy for the complete network and for all HE nodes. The new Tabu search heuristic generates an approximate solution for the given optimization problem by decreasing the energy usage for the LE and HE nodes in the given network. The energy usage could be decreased by minimizing the following objective function (6.1) that is a linear combination of HE nodes' energy (P_{hi}) and LE nodes' energy (P_{ai}).

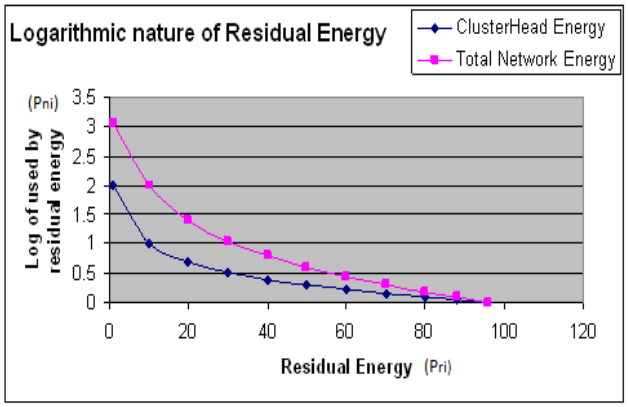


Figure 6.1. Logarithmic nature for cluster heads and complete network of used energies versus residual energy.

Minimize

$$\sum_{i=1}^{p} Bi.\log\left(1+\frac{Phi}{Pri}\right)+\sum_{i=1}^{m} Ai.\log\left(1+\frac{Pai}{Pri}\right) \quad (6.1)$$

Subject to

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Bj \le Ai)\Big) \quad (1)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Xij \le Ai)\Big) \quad (2)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Xij \le Bj)\Big) \quad (3)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Xij \le Lij)\Big) \quad (4)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Xij + Bj = Ai)\Big) \quad (5)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,j\neq i}^{p}(Yij \le Bi)\Big) \quad (6)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Yij \le Bj)\Big) \quad (7)$$

$$\sum_{i=1}^{m}\Big(\sum_{j=1,\,p<m}^{p}(Yij \le Hij)\Big) \quad (8)$$

$$\sum_{j=1}^{p} Xij \le 1 \quad (9)$$

The inequalities, (1)-(9), are the sensor-network constraints. The first constraint (1) makes sure that the HE nodes are smaller in number than the LE nodes. Constraints (2)-(5) check to see if each LE sensor node is associated with at least one HE node within its threshold transmission range using the ITR scheme. Constraints (6)-(8) check to see if every CH is accurately associated to other CHs as per ITR within its given threshold transmission range. Constraint (9) checks if given cluster *j* is being covered properly by CH node *i* and if this constraint is true for all *C* clusters in the given network. Table 6.1. shows the pseudo code of new Tabu heuristics.

Table 6.1. New Tabu heuristic pseudo code.

Step 1. Initialization:
Generate an initial solution according to the Voronoi diagram method and calculate the hop count for each low end sensor node in its respective cluster. Initialize this current solution to the best solution.
Step 2. Neighborhood Search:
2.1 Loop through clusters i=1,2,3,…C
2.1.1 For cluster i, loop through all adjacent clusters $j \neq i$
2.1.1.1 For all nodes in cluster j, evaluate if the distance from the node N_{pj} being evaluated to some node N_{ki} in cluster i is less than equal to low threshold and improves the hop count if moved to cluster i, then move node N_{pj} to cluster i, such that N_{pj} is now N_{pi} node reassigned to cluster i
End loop
2.1.2 For cluster i, Loop through all adjacent clusters j
2.1.2.1 For all nodes in cluster j, evaluate if the distance from node N_{pj} being evaluated to some node N_{ki} in cluster i is less than equal to low threshold and improves the hop count if moved to cluster i, and for all nodes in cluster i, evaluate if the distance from the node N_{mi} being evaluated to some node N_{qj} in cluster j is less than equal to low threshold and improves the hop count if moved to cluster j
then move (pairwise exchange) node N_{pj} to cluster i and move node N_{mi} to cluster j, such that N_{pj} is now N_{pi} (node reassigned to cluster i) and N_{mi} is now N_{mj} (node reassigned to cluster j)
End loop
2.1.3 While ($j \neq i$), Loop through all adjacent clusters j of i
2.1.3.1 Loop through all nodes in clusters j, evaluate if distance from node N_{pj} being evaluated to some node $N_{k(j+1)}$ in cluster $j+1$ is less than equal to low threshold and improves the hop count if moved to adjacent cluster $j+1$, and for all nodes in cluster $j+1$, evaluate if the distance from the node $N_{m(j+1)}$ being evaluated to some node $N_{q(j+2)}$ in cluster $j+2$ is less than equal to low threshold and improves the hop count if moved to cluster $j+2$
then for all adjacent clusters move (cyclic exchange) node N_{pj} to cluster $j+1$ and move node $N_{k(j+1)}$[j+1] to cluster $j+2$.
End loop
2.2 Update the Tabu list
2.3 Update the neighbor table
2.4 Update the best solution
Step 3: Check if all the clusters are evaluated in sequential manner and if i=C, go to step 4 else to step 2.1
Step 4. Stop and report the results.

In the pseudo code, it is given that the initial solution is the Voronoi clusters. A table is maintained by calculating the hop counts for each LE node from its CH in the given cluster. This initial solution is set as the best solution. In a sequential manner, all clusters are checked for possible node reassignment to other clusters. In this algorithm, starting from the first cluster, the Tabu heuristic is applied on the adjacent clusters to see if there are some LE nodes on or near the boundaries of the existing Voronoi clusters. If it finds such LE nodes, they are reassigned to the cluster; this process is called a move. Hence, we have defined three such types of moves.

First, if cluster *i* has adjacent cluster *j* and cluster *j* has LE nodes, *a*, within the threshold radio range of cluster *i*. If node *a* is moved to cluster *i* and the hop count decreases more than it is now from the CH of cluster *j*, LE node *a* is assigned to cluster *i*. The second type of move is a pair-wise exchange. When two clusters are evaluated and it is found that each have one node, those nodes satisfy the threshold radio range and improve the hop count if moved to another cluster. The two nodes are exchanged between two clusters. The third type of move is the cyclic exchange as shown in Figure 6.2. When more than two adjacent clusters are evaluated and each are found to have at least one node that satisfies the threshold radio range and improves the hop count if moved to the adjacent cluster, then the nodes are exchanged among clusters.

As soon as the arrangement of nodes is redefined, it is assessed with the already-established best solution until now that was determined over the complete search process. Immediately after these LE nodes are reassigned to adjacent CHs and at respective iterations, these LE nodes are released from their current clusters. Once a move is made, that move is

added to the Tabu list, and the neighbor table is updated. Now, this solution is set as the best solution. The algorithm stops when all clusters are visited once.

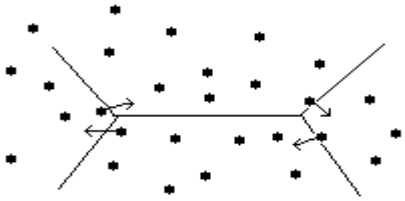


Figure 6.2. Tabu heuristic performing cyclic exchange of nodes.

6.5. The Routing Approach

The Improved Tree Routing (ITR) [4] approach helps decrease the number of hop counts. In the ITR approach, a special tree with only three offspring is devised. This type of tree could use an alternative route resulting in fewer hops. In case of node failures, the spanning tree cannot route data. However, ITR is resilient in such situations because the ITR approach uses its neighbor table to discover an alternative route to forward the data packets.

Each node is allotted a binary address. Nodes maintain a neighbor table which stores the parent, child, and neighbor links. A node could use a neighbor to forward the data instead of setting up the shortest path, provided this neighbor is leading to fewer hops. The root node is the BS with a binary address of 00; all other addresses are computed in a specific manner.

A node cannot have more than 3 children, thus the root has children with address 01, 10, and 11 (all binary). The integer values of the above-mentioned addresses are 1, 2, and 3, respectively. We explain the address at the second level of this tree with an example. To compute addresses for children of a node with address 10, prefix 01 is added to its address (10 in binary) to make the first child's address 0110, with an integer value of 6. Similarly, include

prefix 10 with its address to get the second child's address, that is, 1010 (integer value 10). For the third and last child's address, prefix 11, that is, 1110 (integer value 14).

Easily, only prefixing 2 bits gives the children's addresses. Hence, it reduces a lot of computation. A limitation on the number of offspring leads the tree to grow deeper which reduces parent-child links, thus increasing the neighbor-finding probability, signifying fewer hops. Because the neighbor node is chosen to forward the information in the ITR, it decreases the hop count, instead, as in a spanning tree.

The clustering and ITR routing algorithms are realized in the network simulator (NS-3). Primarily, two agents are designed and implemented: one for splitting the given area into clusters, the VoronoiTabuClusteringAgent, and the other for ITR routing, the ITRAgent. The ITRAgent receives input generated by the VoronoiTabuClusteringAgent as clusters and sets up the ITR routes for the nodes. The implementation details of the ITRAgent are available in [4].

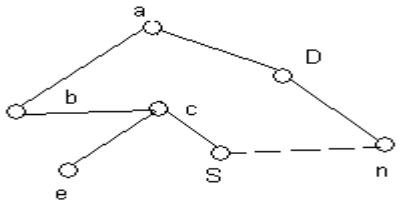


Figure 6.3. Source node S picks a neighbor node, n, which is neither a sibling, nor a grandparent or grandchild of destination node D [4].

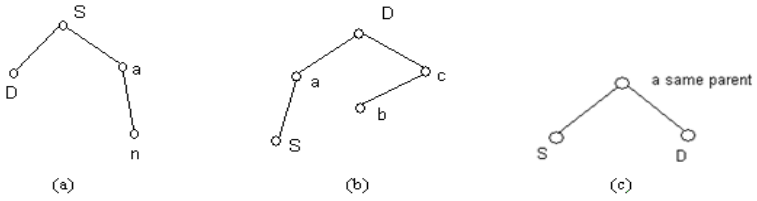


Figure 6.4. Destination, D, and Source, S, nodes having a) child-parent link, b) grandchild-grandparent link, and c) sibling link [4**].**

If the source node, S, and destination node, D, do not have a grandparent/ sibling/ grandchild association, the source node could select a neighbor node, *n*, to route the data [4]. Because node *n* is the neighbor of sender node *S*, it creates a temporary association with *n* to route the data. As shown in Figure 6.3, the spanning-tree path is *S->c->b->a->D*, having a four-hop count, $H_{ST} = 4$. If neighbor node *n* is selected, *S->n->D*, the hop count is two, $H_{ITR} = 2$. Thus, we can say that, if $H_{ST} - H_{ITR} > 0$, $H_{ITR} < H_{ST.}$ Selecting neighbor node *n* to forward the data would not decrease the hop count if a source/destination pair shares some association of type grandparent/grandchild or parent/child or sibling (Figure 6.4).

6.6. Security

Embedding the routing algorithm with security is relatively challenging. Two dissimilar node types are utilized with two-tier HSNs: the LE and the HE nodes. The HEs are smaller in number than the LE nodes. Every LE node is affiliated with only one cluster, and gathered data are communicated to their corresponding CH. The two sensor nodes can transmit data to one another if they and the neighbors are within each other's transmission signal strength, where the transmission signal strength of one sensor node is the radius, t_s, it covers around itself.

The methodology used for adding security in ITR routing is asymmetric, post-distribution key management. Originally, the keys are set up so that each sensor node shares a key with the *n* nodes closest to it. Recall the addressing scheme of ITR, where each node is assigned one unique binary address, node-ID. For sharing data, two nodes set up a shared key between them. The combination of the node's address + level forms the shared key. The addresses for all neighbors are stored in the neighbor table. The level could be calculated by

finding the number of bits in the address divided by 2. For example, a node has an address of 0110. There are 4 bits in its address; dividing by 2 will give us the level of this node in the ITR tree, 2. Thus, nodes at the same level in the ITR tree share this level as part of the key, and this shared key among neighbor nodes is known as a direct key. If the neighbor nodes have dissimilar level numbers, the CH assigns a shared key to them. This kind of shared-key establishment is known as indirect key. Thus, $K_{x,y,p}$ denotes the shared key for two nodes, *x* and *y*, and *p* represents the level number common for both *x* and *y* nodes or is allotted by the respective CH if the level is not the same for those two nodes. An analogous method is applied to set up shared keys between the CH and the BS.

This methodology is secure because the adversary would not be able to implant its own node and because ITR addressing scheme is followed, which is closed-knit limit of only three offspring. Also, the adversary would not be able to figure out the level number if the address is not known.

In the literature [3], [8], [10], the shared key is chosen from the already-stored key pool in the sensor node. Because this process utilizes memory to keep those keys, it also does computations and uses processing power to pick one key from the key pool. Thus, the sensor networks' security technique must be one that does not consume considerable network energy and that has fewer calculations and memory requirements. In our approach, the algorithm used to set up the shared key is trivial because each shared key for a specific node is related to the node-ID. Thus, this technique is not wasting memory to store the key pool, so it uses less computation, thus utilizing less network energy.

After the shared keys are assigned using the asymmetric post-distribution setup approach, nodes do two-way, hand shaking for data exchange during intra-cluster routing. Any node, say *x*, could get ready to route data in case some event occurs in the network. Before routing data to the next node, say *y*, both nodes do two-way hand shaking. Thus, *x* begins by sending a hello packet to *y*: $[x]\ K_{x,y,p} + MAC(K_{x,y,p}, *)$. The $MAC(K_{x,y,p}, *)$ signifies the Message Authentication Code and is engineered at the network layer level having shared key $K_{x,y,p}$. Afterwards, *y* replies to *x*: $[S_{y,x,p}, y]\ K_{x,y,p} + MAC(S_{y,x,p}, *)$, and $S_{y,x,p}$ is a shared key established by *y* and is utilized for the auxiliary data exchange between *x* and *y*.

6.6.1. Additional Security Issues

Even though the secure-ITR scheme does solve some of the security-associated difficulties, this scheme does not address all security issues. Mainly, it does not solve the information-leakage problem via some hidden channels. Another issue is that it does not take care of compromised nodes because we assume that CH nodes are tamper resilient and that LE nodes would not disclose the keys because the keys are computed. Finally, the protocol does not guard against Denial of Service attacks. It is assumed that, for a good number of applications, it is satisfactory to just go for authentication.

Embedding security in ITR comes at a cost; and is represented in Figure 6.5. The extra energy utilization is because of extra computations and extra transmissions required by the Secure-ITR routing protocol. For a 40-byte packet, MAC uses 5 bytes. Shared keying has 2 bytes of overhead with respect to the packet without encryption, so in total, it has around 8 bytes of overhead.

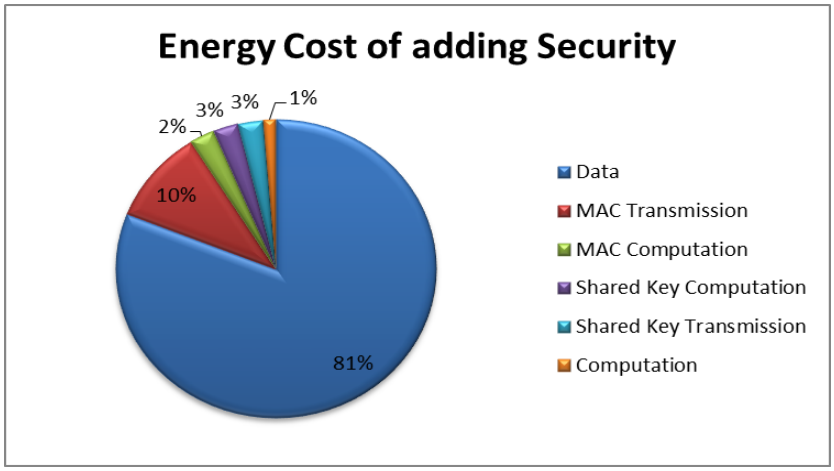


Figure 6.5. Energy usage of embedding security in ITR protocol.

6.7. Evaluation

The performance evaluation of the described Tabu-Voronoi clustering; ITR routing; and asymmetric, post-distribution, key-management technique is done on the NS-3 networks simulator [6]. The TabuVoronoi-SecureImprovedTreeRouting (TV-SITR) protocol has been compared to TabuVoronoi-ImprovedTreeRouting (TV-ITR), Voronoi-ImprovedTreeRouting (V-ITR), Improved Tree Routing (ITR), Low-Energy Adaptive Clustering Hierarchy (LEACH) [5], and Directed Diffusion (DD) [7]. The complete problem is solved in a number of steps. First, the DD and ITR [4] are implemented and compared. Second, the given area of interest is divided into Voronoi clusters, and data are routed using ITR routing. Third, new Tabu search heuristic is applied. Finally, the two-way, hand-shaking security is embedded in the ITR routing.

6.7.1. Simulation Environment

The Secure ITR routing algorithm executes to forward data from LE nodes to the HE node and executes again for the HE node to BS routing. In the above-mentioned six schemes,

TV-SITR, TV-ITR, V-ITR, ITR, LEACH and DD, the nodes are set up in a square field of 650 m x 650 m. Just one BS is operational with address 00; the rest of the sensor nodes are source nodes. Depending on the network size, there could be various numbers of HE and LE nodes. IEEE 802.11 wireless modus operandi is used. It executes at the Media Access Control (MAC) layer. The clustering of nodes in TV-SITR, TV-ITR, V-ITR, ITR, and LEACH takes place at the physical-layer level. Routing for all six schemes takes place at the network layer. The size of the packet is 40 bytes. The simulation start time is 0.1 of a simulated second, and it ends at 150 simulated seconds. The omni-directional radio range is assigned to each node. The data-sharing interface for BS is the Direct Sequenced Spread Spectrum (DSSS) Wave LAN. Initial energies are 100.0 J for HE nodes and are 10.0 J for LE nodes. The HE nodes have a Carrier Sense Threshold of 220 m, and LE nodes have a Carrier Sense Threshold of 65 m. The HE nodes have a Receiving Frequency of 10.0 MHz, and the LE nodes have a Receiving Frequency of 3.0 MHz. The radio range is 914 KHz for HE nodes and is 9140 Hz for LE nodes. The results are the average of 5 different node-deployment orders. For reliability, uniformity, and evenness, various parameters and the energy model for all six schemes are kept the same.

6.7.2. Performance Analysis

Three different features have been chosen for performance analysis. All six schemes are implemented in the NS-3 network simulator for the comparison. Figure 6.6(a) represents the throughput for the six schemes; it could be observed that TV-SITR gives the best throughput of data in comparison to the V-ITR, ITR, LEACH, and DD schemes, but not TV-ITR. The reason could be that TV-SITR has added security overheads while forwarding the

packets for inter-cluster and intra-cluster routing, but TV-ITR does not. Because TV-ITR does not do two-way hand shaking at all, it could forward more data than TV-SITR; moreover, the Tabu heuristic has decreased the hop counts for the entire network, so TV-ITR performs better than any other mentioned scheme. V-ITR performs better than ITR, LEACH, and DD because the given area is divided into Voronoi clusters. Moreover, clustering and ITR routing approaches alone reduce the number of hop counts, making bandwidth available for more data to travel, whereas DD does not use clustering, only goes for the flooding of packets, and uses Publish and Subscribe API, resulting in lower network throughput.

It is fairly evident from Figure 6.6(c) that all six schemes are exhausting some energy for the network functioning. The simulation results show that TV-SITR loses a smaller amount of energy than its counterparts, V-ITR, ITR, LEACH and DD, but uses more energy than TV-ITR. Because TV-SITR is securing the routed data by using two-way hand shaking and computing the keys for data sharing, it is consuming more energy than the TV-ITR algorithm. However, TV-ITR is routing unsecured data, but the new Tabu heuristic is adjusting the Voronoi cluster nodes to minimize the hop count with ITR routing. Thus, TV-ITR utilizes fewer hops and saves substantial amounts of energy. The TV-SITR uses less energy than V-ITR, ITR, LEACH, and DD. This could be because TV-SITR is using the Tabu heuristic for node adjustment, whereas nodes in the LEACH, DD, V-ITR, and ITR schemes use their energy to overhear their neighbor nodes. The TV-SITR, TV-ITR, V-ITR, and ITR schemes are running on HSNs that have two kinds of nodes with different resources and capabilities. However, the LEACH and DD protocols have just one kind of node for the entire

network. In the LEACH protocol for some time frame, certain nodes act as CHs and, consequently, consume additional energy to act as CHs.

Figure 6.6(b) shows the network delays. The network size in all six given schemes combining LE and HE nodes is 13 nodes, 40 nodes, 121 nodes, 200 nodes, 400 nodes, 600 nodes, and 800 nodes. For the total network size, approximately 10% of the network nodes are HE nodes, and the other 90% are LE nodes. Hence, a network of 800 nodes has 720 LE nodes and 80 HE nodes. The network with 600 nodes has 540 LE and 60 HE nodes. The 400-node network has 360 LE and 40 HE nodes. The network with 200 nodes has 20 HE and 180 LE nodes. The 121-node network has 11 HE and 110 LE nodes. The network with 40 nodes has 4 HE and 36 LE nodes. The 13-node network has 3 HE and 10 LE nodes.

From Figure 6.6(b), we can observe that the TV-ITR scheme has the least network delays (in milliseconds). The network delays increase as the size of the network increases because, with the increase in network size, the number of intermediate nodes also increases, directly increasing the number of hop counts and resulting in more delays. The TV-SITR scheme shows less delay than the LEACH, DD, V-ITR, and ITR schemes, but TV-SITR has more delays than the TV-ITR scheme.

6.7.3. Security Analysis of Storage and Connectivity

Table 6.2. shows the security comparison of our scheme with the SBK, LEAP and E-G schemes relating to the stored number of keys (m), and n represents the keys stored before deploying the network on the nodes. If m is set to 30, all the given approaches could set up pair-wise keys that show 100% connectivity for the neighbor sensor nodes, but the E-G approach requires 225 keys and shows just 90% connectivity.

Table 6.2. Security analysis of storage and connectivity.

Schemes	Number of Keys	Example m=30	Percentage of Connectivity
SBK [8]	*m*	30	100%
LEAP [10]	*m+3*	33	100%
E-G [3]	*n*	225	90%
Presented	*m*	30	100%

Last, the presented scheme needs just *(m+1)* additions to set up pair-wise keys for neighbor nodes. The SBK approach requires *(m+1)* assembled exponentiations. The matrix-based technique needs *xy(m-1)* pre-fabricated multiplications for the matrix that has *x* rows, *y* columns, and *m* neighbors. Therefore, the presented approach requires fewer calculations than the SBK technique.

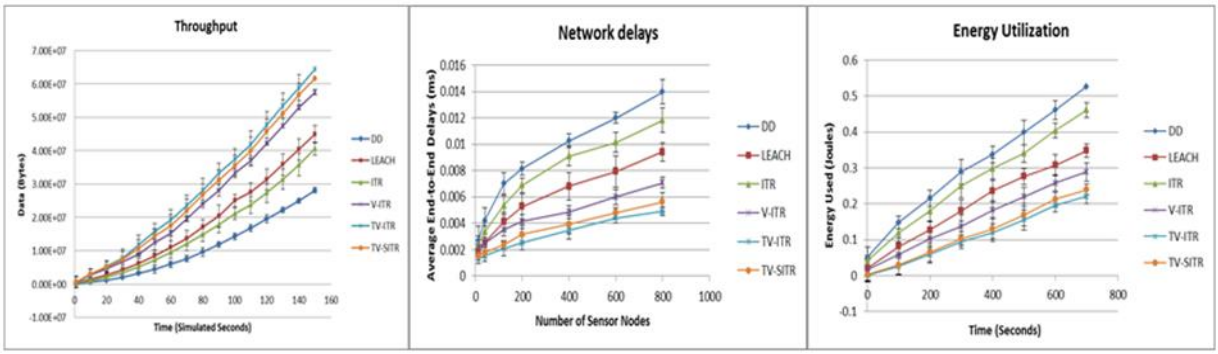


Figure 6.6. (a) Throughput (b) Network delays (c) Energy usage.

6.8. Conclusions

In this paper, we presented a new approach for HSNs, TV-SITR. Clusters are initially formed by Voronoi diagrams, and then, a new Tabu heuristic is applied to reassign the low-end nodes to adjacent clusters. A two-way, hand-shaking scheme is applied to secure the data routed by the ITR algorithm. Through empirical evaluations, we have gained the fact that TV-SITR performs better than the V-ITR, ITR, LEACH, and DD protocols, although added security does

consume certain energy and increases some network delays because of the overheads, which is why TV-SITR is not performing better than TV-ITR. However, the embedded security scheme consumes the least resources and performs better than the SBK, E-G, and LEAP protocols. Overall, the performance of TV-SITR is good.

6.9. Acknowledgments

The research was done with grants W911NF-07-1-0250 and W911NF-08-1-0334 funded by the U.S. Army Research Office.

6.10. References

[1] Du, Xiaojiang, Guizani, Mohsen, Xiao, Yang, and Chen, Hsiao-Hwa, "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks," IEEE Transactions on Wireless Communications, Vol. 6, Issue 9, pages 3395-3401, Sept 2007.

[2] El Rhazi, Abdelmorhit, and Pierre, Samuel, "A Tabu Search Algorithm for Cluster Building in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 8, Issue 4, pages 433-444, April 2009.

[3] Eschenauer, Laurent and Gligor, Virgil D. "A Key-Management Scheme for Distributed Sensor Networks," Proc. Ninth ACM Conference on Computer and Communication Security (CCS '02), pages 41-47, November 18-22, 2002.

[4] Gagneja, Kanwalinderjit K., Du, Xiaojiang, and Nygard, Kendall, "Enhanced Routing in Heterogeneous Sensor Networks," ComputationWorld '09, pages 569-574, Nov. 15-20, 2009.

[5] Heinzelman, Wendi Rabiner, Chandrakasan, Anantha, and Balakrishnan, Hari, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," HICSS, Jan 4-7, 2000.

[6] Henderson, Tom, Lacage, Mathieu, http://www.nsnam.org/ last accessed on July, 2008.

[7] Intanagonwiwat, Chalermek, Govindan, Ramesh, and Estrin, Deborah, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proceedings of ACM Mobicom'00, pages 56-67, Aug 2000.

[8] Liu, Fang, Cheng, Xiuzhen, Ma, Liran, and Xing, Kai, "SBK: A Self-Configuring Framework for Bootstrapping Keys in Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 7, Issue 7, pages 858-868, July 2008.

[9] Navarro, Alejandro, and Rudnick, Hugh, "Large-Scale Distribution Planning—Part II: Macro-Optimization with Voronoi's Diagram and Tabu Search," IEEE Transactions on Power Systems, Vol. 24, Issue 2, pages 752-758, May 2009.

[10] Zhu, Sencun, Setia, Sanjeev, and Jajodia Sushil, "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ACM Transactions on Sensor Networks, Vol. 2, pages 500-528, Nov. 2006.

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

This study addresses the management of wireless sensor networks deployed in an area of interest. It is a known fact that the wireless sensor networks have limited resources, such as processing power, transmitting range, bandwidth availability, and memory. Moreover, routing, security, and clustering of sensors are handled separately by the researchers. Because shortest path routing directly depends on the node position and sets of resources may change dynamically, cumulative and coordinated activities are essential to maintain the sensor nodes' organizational structure. For the longer life of wireless sensor networks, it is better if we apply secure routing to the data in the sensor network, taking care of clustering and positioning the sensors.

We consider the two-tier, heterogeneous sensor networks approach in which we partition the nodes into clusters. Voronoi diagram clusters are initially employed and are adjusted using a Tabu search meta-heuristic. A secure routing technique called Secure Improved Tree Routing (Secure-ITR) is applied to route the data through cluster heads to a base station as appropriate. Through simulation results, we found that Tabu-Voronoi clustering with Secure ITR routing performs better than just routing with ITR or using only Voronoi clusters. Voronoi clusters with ITR routing perform better than ITR routing without clustering. Simulation results also show that Tabu-Voronoi clustering with the Secure ITR routing method increases coverage and throughput, and reduces energy utilization and network delays. However, some overheads are attached while using security keys.

The overall strategy that we employ to conserve energy is summarized as follows:

1. Divide the area of interest into clusters to assist in limiting the necessary transmission range of the low-end and high-end nodes.

2. Utilize an energy-efficient secure routing strategy within clusters and among the cluster heads. Because the transmission range of the sensors is adjustable, the approach supports the determination of high-performance settings.

7.1. Improved Routing

Improved Tree Routing (ITR) is a special tree where the number of offspring is limited to three. The main idea behind such a tree is to use alternative paths with decreased hops for efficient routing. The spanning tree cannot forward data if some node on the route fails. In such situations, the ITR scheme uses a neighbor table to find an alternative path to forward the packets. The sensors record data about their parent, child, and all neighbor nodes in their neighbor table upon deployment. A binary address is assigned to each node in the network. Using the neighbor table, nodes generate parent-child links. A node can use an alternative path to forward that data if it finds that this path is shorter. With the empirical evaluations, we found that the ITR scheme decreases the hop counts for the data to be routed, increases the throughput, and reduces the network delays in comparison to the Directed Diffusion protocol. Therefore, our first objective to develop an efficient routing protocol is fulfilled.

7.2. Embedding Security in ITR Routing

In sensor networks, routing the data does not make sense until the data are secured. Thus, routing among the low-end nodes, high-end nodes, and the base station has been made secure by applying the two-way handshaking security scheme to the Improved Tree Routing (ITR) technique. The embedded security in ITR is asymmetric, post-distribution cryptography. Initially, keys are established. The low-end and CH nodes use their address plus their level in IT routing as their keys. Then, nodes discover their shared keys. A distributed or centralized approach can be used to find a shared key among the nodes. The addresses of the CH and low-end nodes are known to all nodes in their neighborhood in one cluster. By checking the number of bits in the

address, the nodes can know the level of the sensor node in the IT routing tree. (Please refer to the addressing scheme of the ITR in Paper 1.) The level of the sensor node in the IT routing tree equals half the number of bits in the address. The sensor nodes with the same level share their level as the key. However, for nodes with different level numbers, the CH could allocate them with one shared key. Therefore, the shared key, $S_{k,y,x}$, is used between two nodes, x and y, where k is either the level number shared by the x and y nodes or is assigned by the CH if they have different levels. The Secure-ITR data-routing procedure potentially improves the network's energy efficiency by reducing the number of hops utilized to reach to the base station. Although embedding security comes with the price of overheads, almost 20% of the overheads are there because of keys that are an additional data-security feature. Thus, we meet our second objective to embed security in the routed data.

7.3. Clustering Methodology

In our application, we begin with the assignment of nodes to clusters given by the Voronoi diagram; then, we apply the new Tabu search to adjust nodes in clusters to minimize the number of hops. The Voronoi scheme uses Euclidean distance to partition nodes in a two-dimensional plane. However, for data-routing purposes, it is appropriate to use hop counts rather than a direct-distance measure. The use of hop counts implies that Voronoi clustering can lead to routing anomalies attributable to holes, wrap-around regions, nodes that lie near the boundaries, and varying node density in the different clusters.

We apply the Tabu search to adjust the node assignment to clusters, overcoming the limitations of the Voronoi clusters and resulting in improved routing. The Tabu search is a meta-strategy based on a neighborhood search in which iterative moves (adjustments) among solutions are made until a stopping criterion has been satisfied. Three different types of node

reassignments (moves) take place, first for those nodes on or close to the boundaries. If the hop counts from the adjacent CH are less, then assign that node to the adjacent cluster. Second-cyclic transfer moves are performed on clusters where each cluster exchanges a number of nodes but keeps the same number of nodes as the original. Last, pair-wise exchanges among clusters are performed. The reassignment stops in *n* recursions when no refining solutions are found. Next, the reassigned nodes are freed from their present cluster at their respective iteration. The rich set of Tabu search moves results in cluster adjustments that greatly improve network performance, leading to better coverage and reduced energy dissipation for routing data to cluster heads.

In our simulation, the Secure ITR routing algorithm runs first within the Tabu-Voronoi clusters for low-end nodes, and then, it runs for data transmission to the base station from cluster heads. Forming clusters in this manner helps to lower the energy consumption, leading to more network coverage because hop counts decrease with this technique. Hence, we meet our third objective to form high-performance clusters.

7.4. Concluding Remarks

Three different features have been chosen for performance analysis. All six schemes are implemented in the NS-3 network simulator for comparisons. Figure 6.6.(a) represents the throughput for the six schemes; it is observed that TV-SITR gives the best throughput of data in comparison to the V-ITR, ITR, LEACH, and DD schemes, but not TV-ITR. The reason is likely that TV-SITR has added security overheads while forwarding the packets with inter-cluster and intra-cluster routing, but TV-ITR does not. Because TV-ITR does not do two-way hand shaking, it can forward more data than TV-SITR; moreover, the Tabu heuristic has decreased the hop counts for the entire network, so TV-ITR performs better than any other evaluated scheme. The V-ITR performs better than ITR, LEACH, and DD because the given area is divided into

Voronoi clusters. Moreover, clustering and the ITR routing approaches alone reduce the number of hop counts, making bandwidth available for more data to travel, whereas DD does not use clustering; it only goes for flooding packets and a uses Publish and Subscribe API, resulting in lower network throughput.

It is evident from Figure 6.6.(c) that all six schemes are exhausting some energy for the network functioning. The simulation results show that TV-SITR loses a smaller amount of energy than its counterparts, V-ITR, ITR, LEACH, and DD, but uses more energy than TV-ITR. Because TV-SITR secures the routed data by using two-way hand shaking and computing the keys for data sharing, it consumes more energy than the TV-ITR algorithm. The TV-ITR method routes unsecured data, but the new Tabu heuristic adjusts the Voronoi cluster nodes to minimize the hop count with ITR routing. Thus, TV-ITR utilizes fewer hops and saves substantial amounts of energy. The TV-SITR uses less energy than V-ITR, ITR, LEACH, and DD. This could be because TV-SITR is using the Tabu heuristic for node adjustment, whereas nodes in the LEACH, DD, V-ITR, and ITR schemes use their energy to overhear their neighbor nodes. The TV-SITR, TV-ITR, V-ITR, and ITR schemes run on HSNs that have two kinds of nodes with different resources and capabilities. However, the LEACH and DD protocols have just one kind of node for the entire network. In the LEACH protocol for a given time frame, certain nodes act as CHs and, consequently, consume additional energy to act as CHs.

Figure 6.6.(b) shows the network delays. The network size in the given six schemes, combining LE and HE nodes, is 13 nodes, 40 nodes, 121 nodes, 200 nodes, 400 nodes, 600 nodes, and 800 nodes. For the total network size, approximately 10% of the network nodes are HE nodes, and the other 90% are LE nodes. Hence, a network of 800 nodes has 720 LE nodes and 80 HE nodes. The network with 600 nodes has 540 LE and 60 HE nodes. The 400-node

network has 360 LE and 40 HE nodes. The network with 200 nodes has 20 HE and 180 LE nodes. The 121-node network has 11 HE and 110 LE nodes. The network with 40 nodes has 4 HE and 36 LE nodes. The 13-node network has 3 HE and 10 LE nodes.

From Figure 6.6.(b), we observe that the TV-ITR scheme has the least network delays in milliseconds. The network delays are increasing as the size of the network increases because, with the larger network size, the number of intermediate nodes also increases, directly increasing the number of hop counts and resulting in more delays. The TV-SITR scheme shows less delay than the LEACH, DD, V-ITR, and ITR schemes, but it has more delays than the TV-ITR scheme.

Figure 6.6. shows the performance comparison of the Directed Diffusion, LEACH, ITR, V-ITR, TV-ITR, and TV-SITR protocols in terms of throughput (a), network delays (b), and energy usage (c). Five runs are averaged, so the figure also shows the standard deviation from the mean at each point. The table 7.1. shows the throughput variances and shows the throughput of data in bytes for the Directed Diffusion, LEACH, ITR, V-ITR, TV-ITR, and TV-SITR protocols with respect to time in simulated seconds. Overall, the TV-ITR scheme performs best; TV-SITR performs better than the V-ITR, ITR, LEACH, and DD protocols. This could be attributed to the fact that Tabu has reduced the hop counts for the nodes lying on or near the Voronoi cluster boundaries by using new Tabu heuristics. The V-ITR performs better than ITR because V-ITR is partitioned into Voronoi clusters and because the nodes hear only those nodes that are in a similar cluster. Therefore, the ITR protocol loses more energy than V-ITR. ITR performs better than LEACH and DD. The LEACH protocol uses clustering, and DD does not use node clustering. DD also uses flooding for data forwarding, and uses the Publish and Subscribe interfaces. However, the performance of TV-ITR is better than TV-SITR. This could be because of overheads attached to the data in the TV-SITR with added security. For a packet of

40 bytes, 8 bytes are used for security in TV-SITR. Thus, TV-SITR has 20% overhead compared to TV-ITR that directly affects the network traffic.

Table 7.1. Throughput.

Time in Simulated Seconds	DD (bytes)	LEACH (bytes)	ITR (bytes)	V-ITR (bytes)	TV-ITR (bytes)	TV-SITR (bytes)
1	1.67E+05	3.12E+05	2.76E+05	5.01E+05	5.50E+05	5.69E+05
10	6.21E+05	1.39E+06	1.01E+06	2.66E+06	2.78E+06	3.08E+06
20	1.17E+06	2.62E+06	2.04E+06	4.57E+06	5.28E+06	5.08E+06
30	2.06E+06	4.20E+06	3.44E+06	6.77E+06	7.70E+06	7.39E+06
40	3.20E+06	6.19E+06	5.21E+06	8.99E+06	1.17E+07	1.07E+07
50	4.43E+06	8.46E+06	7.29E+06	1.24E+07	1.58E+07	1.43E+07
60	5.92E+06	1.10E+07	9.52E+06	1.53E+07	1.92E+07	1.75E+07
70	7.62E+06	1.37E+07	1.20E+07	1.95E+07	2.36E+07	2.21E+07
80	9.57E+06	1.71E+07	1.48E+07	2.41E+07	2.80E+07	2.68E+07
90	1.18E+07	2.04E+07	1.78E+07	2.81E+07	3.34E+07	3.13E+07
100	1.43E+07	2.51E+07	2.12E+07	3.32E+07	3.76E+07	3.56E+07
110	1.68E+07	2.76E+07	2.37E+07	3.70E+07	4.19E+07	3.99E+07
120	1.96E+07	3.14E+07	2.73E+07	4.22E+07	4.77E+07	4.57E+07
130	2.22E+07	3.59E+07	3.15E+07	4.74E+07	5.35E+07	5.10E+07
140	2.50E+07	4.04E+07	3.58E+07	5.30E+07	5.89E+07	5.68E+07
150	2.81E+07	4.50E+07	4.06E+07	5.75E+07	6.44E+07	6.17E+07
Variance	6.69E+11	5.26E+12	4.28E+12	1.63E+12	9.13E+12	8.43E+12

Table 7.2. shows the average end-to-end delays for the routed data in milliseconds (ms) for the Directed Diffusion, LEACH, ITR, V-ITR, TV-ITR, and TV-SITR protocols with respect to the network sizes as represented in Table 7.3.

Table 7.2. Network delays.

Number of Nodes	DD (ms)	LEACH (ms)	ITR (ms)	V-ITR (ms)	TV-ITR (ms)	TV-SITR (ms)
13	2.69E-03	1.85E-03	2.34E-03	2.19E-03	1.33E-03	1.53E-03
40	4.20E-03	2.52E-03	3.34E-03	2.46E-03	1.53E-03	1.76E-03
121	7.00E-03	4.12E-03	5.36E-03	3.49E-03	2.08E-03	2.38E-03
200	8.15E-03	5.26E-03	6.88E-03	4.14E-03	2.51E-03	3.15E-03
400	1.02E-02	6.80E-03	9.10E-03	4.85E-03	3.43E-03	3.89E-03
600	1.20E-02	7.94E-03	1.01E-02	5.99E-03	4.40E-03	4.80E-03
800	1.40E-02	9.40E-03	1.18E-02	7.07E-03	4.90E-03	5.60E-03
Variance	6.09025E-07	8.1543E-07	4.98952E-07	1.92092E-07	2.5712E-07	2.02901E-07

Table 7.3. shows the network sizes. The network size in all six given schemes combining LE and HE nodes is 13 nodes, 40 nodes, 121 nodes, 200 nodes, 400 nodes, 600 nodes, and 800 nodes. For the total network size, approximately 10% of the network nodes are HE nodes, and the other 90% are LE nodes. Hence, a network of 800 nodes has 720 LE nodes and 80 HE nodes. The network with 600 nodes has 540 LE and 60 HE nodes. The 400-node network has 360 LE and 40 HE nodes. The network with 200 nodes has 20 HE and 180 LE nodes. The 121-node network has 11 HE and 110 LE nodes. The network with 40 nodes has 4 HE and 36 LE nodes. The 13-node network has 3 HE and 10 LE nodes. From Figure 6.6.(b), it is clear that, with the increased number of sensor nodes in the network, the delays also increase for all schemes. The reason is that, with increased network size, intermediate nodes also increase for the data-routing path, directly increasing the network delays. However, the TV-SITR has more network delays than TV-ITR. This could be because the TV-SITR performs two-way hand shaking while data routing so adds to more delays.

Table 7.3. Network sizes.

Network Size=Low End nodes + Cluster Heads		
13	=	10 + 3
40	=	36 + 4
121	=	110 + 11
200	=	180 + 20
400	=	360 + 40
600	=	540 + 60
800	=	720 + 80

Table 7.4. shows the energy usage by the network in joules (J) for the Directed Diffusion, LEACH, ITR, V-ITR, TV-ITR, and TV-SITR protocols with respect to time in seconds.

Table 7.4. Energy usage.

Time (Sec.)	DD (J)	LEACH (J)	ITR (J)	V-ITR (J)	TV-ITR (J)	TV-SITR (J)
1	5.07E-02	2.09E-02	4.25E-02	1.65E-02	1.90E-03	2.34E-03
100	1.48E-01	8.18E-02	1.20E-01	5.87E-02	2.61E-02	2.96E-02
200	2.16E-01	1.28E-01	1.80E-01	1.02E-01	5.93E-02	6.56E-02
300	2.89E-01	1.81E-01	2.50E-01	1.37E-01	9.38E-02	1.02E-01
400	3.37E-01	2.36E-01	2.98E-01	1.83E-01	1.20E-01	1.30E-01
500	3.98E-01	2.76E-01	3.40E-01	2.19E-01	1.55E-01	1.69E-01
600	4.61E-01	3.08E-01	4.04E-01	2.58E-01	1.96E-01	2.12E-01
700	5.27E-01	3.49E-01	4.61E-01	2.89E-01	2.21E-01	2.39E-01
Variance	7.60E-04	6.27E-04	6.59E-04	7.22E-04	4.88E-04	5.46E-04

The hierarchical routing that is used to forward sensed data reduces hop counts for data to reach the base station, directly enhancing the security of the deployed sensor network system. Using this routing technique, energy utilization is reduced because energy expanded in the hop count is an important factor. More efficient routes also mean that an adversary will have a smaller amount of time to intercept the data, increasing network security. The Sybil attack is defended with our scheme because nodes store their addresses in their neighbor tables and use

these addresses with shared keys for authentication, so one node cannot act as if it is another node, completely preventing a Sybil attack. The given topology uses intra-cluster routing and inter-cluster routing. Only the low-end sensors send data to their neighbor using their neighbor table, and a parent-child association is established by the cluster head (a high-end node). With inter-cluster routing, the cluster heads send data to the base station; no other node can route data, so the wormhole and sinkhole attacks will not occur. This topology also avoids the Hello Flood Attack because it is a collaborative protocol, using an address-based hierarchy to send data from low-end nodes to a high-end node and, finally, to the base station. Adjusting the transmission power helps low-energy nodes reach neighbors up to certain distance. By minimizing the radio range to some threshold level, fewer neighbors are reached, resulting in a reduced breach of information and lower energy consumption.

7.5. Lessons Learned

In this research, we developed an efficient key-management approach embedded in an improved tree-routing algorithm for clustered, heterogeneous sensor networks. This research was funded by the Army Research Office. The simulation results show that this scheme offers good security and uses less computation with substantial savings in memory requirements, when compared with some other key-management, clustering, and routing techniques. The simulations are performed on the NS-3 network simulator. One can write the protocol, and that protocol should be added to the NS-3. It is the most difficult part because there can easily be subtle errors while doing so; the errors could take days to fix. First, we wrote the Improved Tree Routing protocol and added it to NS-3. We then set up the nodes in the area of interest so that a performance analysis could be done for the ITR routing. This was working poorly because no data transmission took place. After a detailed investigation, it was determined that some low-end

nodes were not within the proper transmission range with high-end nodes and other low-end nodes. Moreover, the high-end nodes were not within each other's radio range, so no data transmission took place. Hence, the question arose about how to fix this radio-range problem. To address this problem, we first calculated the threshold radio ranges for low-end and high-end nodes. By assigning radio ranges to the nodes, the ITR routing protocol was enabled. A second step was to embed security in the routing protocol. The security part was mathematics-based and went well, but increased the traffic in the network. Finally, we dealt with the challenge of dividing the area of interest into high-performance, Tabu-Voronoi clusters. Initially, the area was partitioned into Voronoi clusters. We then applied the Tabu search on neighbor nodes that were on or near the boundaries of adjacent clusters to determine if that would reduce the number of hop counts. Other moves were pair-wise exchange and cyclic transfers. These moves improved the Tabu-Voronoi clusters, but in some cases, overwrote previously made moves. The simulator provided output files that were analyzed using statistical tools. The simulation results showed good performance for our scheme compared with existing schemes.

7.6. Future Work

The algorithm presented in this study does not solve the problem when the sensor nodes are mobile. A number of researchers have worked on mobile sensors in a wireless sensor network. This work could be enhanced when a network only has cluster heads as mobile nodes and when a network has some cluster heads as mobile and some low-end nodes as mobile. Then, we can see how the ITR algorithm behaves when the clustering nodes move from one point to another.

REFERENCES

[1] Ai, Chunyu, Hou, Hailong, Li, Yingshu, and Beyah, Raheem, "Authentic Delay Bounded Event Detection in Heterogeneous Wireless Sensor Networks," Journal Ad Hoc Networks, Vol. 7, Issue 3, pages 599-613, May 2009.

[2] Al-Sultan, Khaled S., "A Tabu Search Approach to the Clustering Problem," Pattern Recognition, Vol. 28, Issue 9, pages 1443-1451, Sept 1995.

[3] Al-Yamani, Ahmad, Sait, Sadiq M., Youssef, Habib, and Barada, Hassan, "Parallelizing Tabu Search on a Cluster of Heterogeneous Workstations," Journal of Heuristics, Vol. 8, Issue 3, pages 277-304, May 2002.

[4] Barbarosoglu, Gulay, and Ozgur, Demet, "A Tabu Search Algorithm for the Vehicle Routing Problem," Computers and Operations Research, Vol. 26, Issue 3, pages 255-270, March 1999.

[5] Bonivento, Alvise, Fischione, Carlo, Necchi, Luca, Pianegiani, Fernando, and Sangiovanni-Vincentelli, Alberto, "System Level Design for Clustered Wireless Sensor Networks," IEEE Transactions on Industrial Informatics, Vol. 3, Issue 3, pages 202-214, Aug 2007.

[6] Brusco, Michael J., Cradit, Dennis, and Stahl, Stephanie, "A Simulated Annealing Heuristic for a Bicriterion Partitioning Problem in Market Segmentation," Journal of Marketing Research, Vol. 39, Issue 1, pages 99-109, 2002.

[7] Cheng, Wei, Xing, Kai, Cheng, Xiuzhen, Lu, Xicheng, Lu, Zexin, Su, Jinshu, Wang, Baosheng, and Liu, Yujun, "Route Recovery in Vertex-Disjoint Multipath Routing for Many-to-One Sensor Networks," MobiHoc'08, pages 209-219, May 26-30, 2008.

[8] de Werra, Dominique, and Hertz, Alain, "Tabu Search Techniques: A Tutorial and an Application to Neural Networks," OR Spektrum, Vol. 11, pages 131-141, 1989.

[9] Du, Xiaojiang, Guizani, Mohsen, Xiao, Yang, and Chen, Hsiao-Hwa, "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks," IEEE Transactions on Wireless Communications, Vol. 6, Issue 9, pages 3395-3401, Sept 2007.

[10] El Rhazi, Abdelmorhit, and Pierre, Samuel, "A Tabu Search Algorithm for Cluster Building in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 8, Issue 4, pages 433-444, April 2009.

[11] Faigle, Ulrich, and Kern, Walter, "Some Convergence Results for Probabilistic Tabu Search," ORSA Journal on Computing, Issue 4, pages 32-37, 1992.

[12] Fall, Kevin, and Varadhan, Kannan, "NS Notes and Documentation," available from http://www-mash.cs.berkeley.edu/ns/, accessed on May 2009.

[13] Fisher, Marshall L., and Jaikumar, Ramchandran, "A Generalized Assignment Heuristic for Vehicle Routing," Networks, Vol. 11, Issue 2, pages 109-124, 1981.

[14] Fox, Bennett L., "Integrating and Accelerating Tabu Search, Simulated Annealing and Genetic Algorithms," Annals of Operations Research, Issue 41, pages 47-67, 1993.

[15] Glover, Fred, and Dyer, James, "A Barge Sequencing Heuristic," Transportation Science, Vol. 4, Issue 3, pages x-y, August 1970.

[16] Glover, Fred, and Laguna, Manuel, *Tabu Search*, Kluwer Academic Publishers, 1997.

[17] Glover, Fred, "Tabu Search, Part I," ORSA Journal on Computing, Issue 1, pages 190-206, 1989.

[18] Glover, Fred, "Tabu Search, Part II," ORSA Journal on Computing, Issue 2, pages 4-32, 1990.

[19] Glover, Fred, Taillard, Eric, Laguna, Manuel, and de Werra, Dominique, "Tabu Search," Annals of Operations Research, Vol. 41, pages x-y, 1993.

[20] Heinzelman, Wendi Rabiner, Chandrakasan, Anantha, and Balakrishnan, Hari, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," HICSS, Jan 4-7, 2000.

[21] Intanagonwiwat, Chalermek, Govindan, Ramesh, and Estrin, Deborah, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proceedings of ACM Mobicom'00, pages 56-67, Aug 2000.

[22] Intanagonwiwat, Chalermek, Govindan, Ramesh, Estrin, Deborah, Heidemann, John, and Silva, Fabio, "Directed Diffusion for Wireless Sensor Networking," Networking, IEEE/ACM Transactions, Vol. 11, Issue 1, pages 2-16, Feb 2003.

[23] Karlof, Chris, and Wagner, David, "Secure Routing in Sensor Networks: Attacks and Countermeasures," Ad Hoc Networks, Vol. 1, Issue 2-3, pages 293-315, Sept 2003.

[24] Karp, Brad, and Kung, H.T., "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proceedings of ACM Mobicom'00, pages 243-254, Aug 2000.

[25] Krajewska, Marta Anna, and Kopfer, Herbert, "Transportation Planning in Freight Forwarding Companies: Tabu Search Algorithm for the Integrated Operational Transportation Planning Problem," European Journal of Operational Research, Vol. 197, Issue 2, pages 741-751, 2009.

[26] Lee, Chae Y., and Kang, Hyon G., "Cell Planning with Capacity Expansion in Mobile Communications: A Tabu Search Approach," IEEE Transactions on Vehicular Technology, Vol. 49, No. 5, pages 1678-1691, Sept 2000.

[27] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, pages 281-297, 1967.

[28] Madria, Sanjay, and Yin, Jian, "SeRWA: A Secure Routing Protocol Against Wormhole Attacks in Sensor Networks," Ad Hoc Networks, Vol. 7, Issue 6, pages 1051-1063, Aug 2009.

[29] Mhatre, Vivek, Rosenberg, Catherine, Kofman, Daniel, Mazumdar, Ravi, and Shroff, Ness, "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," IEEE Transactions on Mobile Computing, Vol. 4, Issue 1, pages 4-15, Jan/Feb 2005.

[30] Misra, Satyajayant, and Xue, Guoliang, "SAS: A Simple Anonymity Scheme for Clustered Wireless Sensor Networks," Proceeding os IEEE ICC, pages 3414-3419, 2006.

[31] Misra, Satyajayant, and Xue, Guoliang, "CluRoL: Clustering Based Robust Localization in Wireless Sensor Networks," MILCOM 2007 IEEE, pages 1-7, Oct 29-31, 2007.

[32] Montané, Fermín, Alfredo, Tang, and Galvao, Roberto Diéguez, "A Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery Service," Computers and Operations Research, Vol. 33, Issue 3, pages 595-619, March 2006.

[33] Navarro, Alejandro, and Rudnick, Hugh, "Large-Scale Distribution Planning—Part II: Macro-Optimization with Voronoi's Diagram and Tabu Search," IEEE Transactions on Power Systems, Vol. 24, Issue 2, pages 752-758, May 2009.

[34] Pacheco, Joaquín A., "A Scatter Search Approach for the Minimum Sum-of-Squares Clustering Problem," Computers and Operations Research, Vol. 32, Issue 5, pages 1325-1335, May 2005.

[35] Perrig, Adrian, Szewczyk, Robert, Tygar J. D., Victorwen, and Culler, David E., "SPINS: Security Protocols for Sensor Networks," Journal of Wireless Networks, Vol. 8, Issue 5, pages 521-534, Sept 2002.

[36] Pham, D. T., and Karaboga, D., *Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural networks*, Springer, 2000.

[37] Punj, Girish, and Stewart, David, "Cluster Analysis in Marketing Research: Review and Suggestions for Application," Journal of Marketing Research, Vol. 20, pages 134-148, 1983.

[38] Qiu ,Wanzhi, Skafidas, Efstratios, and Hao, Peng, "Enhanced Tree Routing for Wireless Sensor Networks," Ad Hoc Networks, Vol. 3, pages 638-650, July 2008.

[39] Scheuerer, Stephan, "A Tabu Search Heuristic for the Truck and Trailer Routing Problem," Computers and Operations Research, Vol. 33, Issue 4, pages 894-909, April 2006.

[40] Sohraby, Kazem, Minoli, Daniel, and Znati, Taieb F., *Wireless Sensor Networks: Technology, Protocols and Applications* (3rd edition), Wiley, 2007.

[41] Sun, Kun, Peng, Pai, Ning, Peng, and Wang, Cliff, "Secure Distributed Cluster Formation in Wireless Sensor Networks," Proceedings of the 22nd Annual Computer Security Applications Conference, pages 131-140, 2006.

[42] Sung, Chang S., and Jin, Hyun Woong, "A Tabu-Search-Based Heuristic for Clustering," Pattern Recognition, Issue 33, pages 849-858, 2000.

[43] Thanos, Stathopoulos, Martin Lukac, Dustin McIntire, Heidemann, John, Estrin, Deborah, and Kaiser W., "End-to-End Routing for Dual-Radio Sensor Networks," Infocom, pages 2252-2260, May 2007.

[44] Thompson, Paul, and Orlin, James, "The Theory of Cyclic Transfers," Operations Research Center Working Paper, MIT, Cambridge MA, Aug 1989.

[45] Thulasiraman, Preetha, Ramasubramanian, Srinivasan, and Krunz, Marwan, "Disjoint Multipath Routing to Two Distinct Drains in a Multi-drain Sensor Network," Infocom, pages 643-651, May 2007.

[46] Tian, Shourui, Shatz, Sol M., Yu, Yang, and Li, Juzheng, "Querying Sensor Networks Using Ad Hoc Mobile Devices: A Two-Layer Networking Approach," Ad Hoc Networks, Vol. 7, Issue 5, pages 1014-1034, July 2009.

[47] Wang, Li-Chun, Wang, Chung-Wei, and Liu, Chuan-Ming, "Optimal Number of Clusters in Dense Wireless Sensor Networks: A Cross-Layer Approach," IEEE Transactions on Vehicular Technology, Vol. 58, Issue 2, pages 966-976, Feb 2009.

[48] Ward, Joe, "Hierarchical Grouping to Optimize an Objective Function," Journal of the American Statistical Association, pages 236-244, March 1963.

[49] Wedel, Michel, and Kamakura, Wagner, *Market Segmentation: Conceptual and Methodological Foundations* (2nd edition), International Series in Quantitative Marketing, Kluwer Academic Publishers, 2000.

[50] Yang, Yi, Wang, Xinran, Zhu, Sencun, and Cao, Guohong, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," ACM Transaction on Information and System Security, Vol. 11, Issue 4, pages 1-43, July 2008.

[51] Yongguo, Liu, Zhang, Yi , Hong, Wu, Mao, Ye, and Kefei, Chen, "A Tabu Search Approach for the Minimum Sum-of-Squares Clustering Problem," Information Sciences: an International Journal, Vol. 178, Issue 12, pages 2680-2704, June 2008.

[52] Younis, Ossama, Krunz, Marwan, and Ramasubramanian, Srinivasan, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," Network, IEEE, Vol. 20, Issue 3, pages 20-25, May-June 2006.

[53] Yu, Yang, Prasanna, Viktor, and Krishnamachari, Bhaskar, *Information Processing and Routing in Wireless Sensor Networks*, World Scientific Publishing Co. Ltd., 2006.