8-2009

QoS Provision for Wireless Sensor Networks

Dengfeng Yang
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Dengfeng Yang entitled "QoS Provision for Wireless Sensor Networks." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Major Professor

We have read this dissertation and recommend its acceptance:

Rapinder S. Sawhney, Denise F. Jackson, Xiaorui Wang

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Dengfeng Yang entitled "QoS Provision for Wireless Sensor Networks." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Major Professor

We have read this dissertation
and recommend its acceptance:

Rapinder S. Sawhney

Denise F. Jackson

Xiaorui Wang

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

QoS Provision for Wireless Sensor Networks

A Dissertation
Presented for the
Doctor of Philosophy
Degree

The University of Tennessee, Knoxville

Dengfeng Yang
August 2009

Dedication

To my daughter, Angelina An'an Yang

Acknowledgments

First and foremost, I am deeply indebted to my family, especially my mother, Yulin Wang. Where I am today is in no small part due to their love and support. I also owe a special thanks to my wife in the last five years, Miss Wei Xu. Without her unconditional support and continual inspiration to follow my dream, this work will go nowhere.

I additionally would like to thank my advisor, Dr. Xueping Li, my Dissertation Committee Chairperson and Mentor, for his patient guidance, constant encouragement, endless support and constructive criticism during my study. Also, I would like to thank, Dr. Xiaorui Wang. His advice and counsel over the years have been critical to this work. To Dr. Rapinder Sawhney, I say thank you as well for the many discussions with regard to this research and life in general. I also want to express my gratitude to Dr. Denise Jackson for serving as my supervisory committee member and for giving me wonderful ideas and advice.

Finally, I must express my appreciation to my officemates and my friends. In particular, to Laigang, Yuerong and Jojo, I express my sincerest gratitude for the many conversations that have had a tremendous impact on my research and myself as a person. To my friends, Zhen Liu, Shiying Si, Bo Zhang, Teng Wang and Jessie Luo, I am profoundly grateful for their friendships. Especially to Junxia Chang, her inspiration through my darkest time has always been a source of strength.

Abstract

Wireless sensor network is a fast growing area of research, receiving attention not only within the computer science and electrical engineering communities, but also in relation to network optimization, scheduling, risk and reliability analysis within industrial and system engineering. The availability of micro-sensors and low-power wireless communications will enable the deployment of densely distributed sensor/actuator networks. And an integration of such system plays critical roles in many facets of human life ranging from intelligent assistants in hospitals to manufacturing process, to rescue agents in large scale disaster response, to sensor networks tracking environment phenomena, and others.

The sensor nodes will perform significant signal processing, computation, and network self-configuration to achieve scalable, secure, robust and long-lived networks. More specifically, sensor nodes will do local processing to reduce energy costs, and key exchanges to ensure robust communications. These requirements pose interesting challenges for networking research. The most important technical challenge arises from the development of an integrated system which is 1)energy efficient because the system must be long-lived and operate without manual intervention, 2)reliable for data communication and robust to attackers because information security and system robustness are important in sensitive applications, such as military.

Based on the above challenges, this dissertation provides Quality of Service (QoS) implementation and evaluation for the wireless sensor networks. It includes the following 3

modules, 1) energy-efficient routing, 2) energy-efficient coverage, 3). communication security. Energy-efficient routing combines the features of minimum energy consumption routing protocols with minimum computational cost routing protocols. Energy-efficient coverage provides on-demand sensing and measurement. Information security needs a security key exchange scheme to ensure reliable and robust communication links. QoS evaluation metrics and results are presented based on the above requirements.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless sensors are autonomous sensing devices with wireless communication capability within short distance. A sensor node typically consists of a power unit, a sensing unit, a processing unit, a storage unit, and a wireless transmitter/receiver (Akyildiz et al., 2002; Kahn et al., 1999; Pottie and Kaiser, 2000). A Wireless Sensor Network (WSN) contains large number of sensor nodes (Akyildiz et al., 2002; Akkaya and Younis, 2005b) deployed in controlled and safe environments (such as home, office, warehouse) or uncontrolled and hostile environments (such as battlefields, toxic regions, et al.) and has wide applications including military sensing and tracking, real-time traffic and pollution monitoring, and wildlife monitoring, and so on.

Wireless Sensor Networks (WSN) are self-organized ad-hoc networks whose nodes are capable of sensing, gathering, processing and communicating data, especially the data pertaining to the physical medium in which they are embedded. This enables sensing and actuation at a fine grained level, both spatially and temporally. Though significant on their own, wireless sensor networks play a central role in achieving the goal of truly ubiquitous computing and smart environments. By interfacing the data collection and dissemination

infrastructure of WSN with external networks and devices, control and automation of physical entities like houses, factories, farms and so on can be achieved at a level that was not possible before.

1.1 Resource Constrained Wireless Sensor Networks

Unlike traditional networked systems, a sensor network is constrained by finite on-board battery power and limited network communication bandwidth. In addition, sensor networks are spatially aware and are more closely linked to geographic locations and the physical environments than centralized systems. A sensor node in a typical sensor network has a limited battery, a microprocessor, and a small amount of memory for signal processing and task scheduling (Akyildiz et al., 2002). Each node is equipped with one or more sensing devices, such as sensors for visible or infrared light, changing magnetic field, electrical resistance, acceleration or vibration, pH, humidity, or temperature; acoustic microphone arrays, and/or video or still cameras.

One of the most important constraints on sensor nodes is the low power consumption requirement. Sensors nodes carry limited, generally irreplaceable, power sources. Therefore, while traditional networks QoS evaluation aims to achieve high provisioning or high bandwidth, QoS evaluation of sensor network protocols must focus primarily on power conservation and security. They must have built-in trade-off mechanisms that give the end user the options of prolonging network lifetime at the cost of lower throughput or higher transmission delay.

Wireless devices must operate for a long period of time, relying on their battery power. Although many developers have looked at extending the life of a wireless system from a hardware point of view, such as directional antennas and improving battery life, energy efficient algorithms are still a hot topic for wireless sensor networks.

1.2 Vulnerable Wireless Sensor Networks

Usually sensors need to communicate with their neighboring sensors and base station, and the messages between them may contain sensitive information. Thus, it is crucial to enable encryption and authentication among sensor nodes to protect confidentiality, integrity, and availability of the communication and computation of a WSN. Some traditional key distribution algorithms used for wired networks, or wireless mesh networks are not suitable for WSN because of the unique properties of sensor networks, novel key distribution algorithms must be further investigated. Therefore, QoS evaluation of sensor networks must also need to focus on constructing secure communication links and obtaining confidential data.

It is challenging to secure WSN communications because of a WSN's inherent characteristics Camtepe and Yener (2005): (i) wireless nature of communication, (ii)resource limitation on sensor nodes, (iii) very large and dense WSN, (iv) lack of fixed infrastructure, (v) unknown network topology prior to deployment, (vi) high risk of physical attacks to unattended sensors. Moreover, in some deployment scenarios sensor nodes need to operate under adversarial conditions. The limited resources at sensor nodes rule out the use of public key cryptosystems such as RSA and Diffie-Hellman key agreement to serve this purpose (Rivest et al., 1978; Diffie and Hellman, 1976). A naive idea is to use a single shared key in the whole WSN, but an adversary can easily obtain the key and decipher the message. Another way is to customize public key cryptography and elliptic key cryptography for low-power devices, which are still considered as costly due to their high processing requirements (Huang et al., 2004; Malan et al., 2004).

However, these methods suffer from two major problems: high deployment density requirement and the degradation of resilience against nodes capture (Zhou et al., 2005, 2006). A more robust scheme is strongly needed for the resilience of wireless sensor networks under limited resource constraints.

1.3 Contribution and Scope

This dissertation focuses on designing a stand-alone wireless sensor network for providing QoS in terms of energy efficiency and robust communication. It primarily addresses three issues related to QoS improvement and evaluation. Firstly, it addresses an energy efficient routing protocol in sensor networks, where linear programming and heuristic optimization algorithms are presented Secondly, it presents an energy efficient coverage algorithm, where coverage and connectivity are considered at the same time to improve the system responsive time and efficiency. Thirdly, a group key distribution scheme is proposed to construct reliable and robust communication links between sensors.

1.3.1 Energy Efficient Routing

Energy efficient can be seen as one critical measure of QoS for wireless sensor network. Current energy-aware protocols are mainly categorized into 1) minimum energy routing protocols (Singh et al., 1998; Haque et al., 2005); 2) max-min routing protocols (Li et al., 2001b; Toh, 2001; Zhang and Mouftah, 2006); and 3) minimum cost routing protocols (Chang and Tassiulas, 1999, 2000; Zhang and Mouftah, 2006; Kalpakis et al., 2002). The minimum energy routing protocols minimize total consumed energy to reach the destination thus minimizing the unit energy consumption per packet. However, this protocol may not maximize the network lifetime since the residual energy is not taken into account. Consequently, some nodes on the minimum energy routes will easily fail due to the heavy forwarding load. The max-min routing protocols, such as conditional max-min battery capacity routing (CMMBCR) (Toh, 2001), max-min zP_{min} (Li et al., 2001b) and MREP (Zhang and Mouftah, 2006), avoid this problem by choosing a route that maximizes the minimal residual energy of some nodes in this route. But these protocols add the overhead of control packets for the on-demand version and it is difficult to decide the optimal threshold value that determines the operation modes.

Besides the above three categories of energy-aware protocols, there are many other efforts to extend the lifetime of a WSN. Linear programming (LP) formulations are proposed to maximize the network lifetime (Chang and Tassiulas, 1999, 2000; Zussman and Segall, 2003; Hou et al., 2005c; Kalpakis et al., 2002). Bhardwaj (Bhardwaj and Chandrakasan, 2002), Sadagopan (Sadagopan and Krishnamachari, 2004) and Sankar (Sankar and Liu, 2004) derive upper bounds on the lifetime of a sensor network that collects data from a targeted area under the energy constrained situation.

Most of the above energy-aware protocols are based on the flat network or hierarchical network and neglect the information that how the sensor nodes are deployed and distributed. The geographic location information is easy to obtain due to the very nature of a WSN and can be used to build routing algorithms to construct a scalable energy-saving communication infrastructure. An advantage of geographic routing is its stateless and localized nature since the packet forwarding depends only on the location information of the candidate nodes in the vicinity and the destination node. Hence, the geographic routing is scalable because it does not require additional control overhead.

In this dissertation, we present a two-layered sensor network model by incorporating the geographic deployment knowledge of the network, based on which we propose a Hybrid Energy-Efficient Routing Scheme (HERS) to extend the network lifetime by considering the max-min residual energy routing and the min-max cost routing. To the best of our knowledge, this is the first work to develop the hybrid energy-efficient routing protocol while minimizing the communication energy consumption using the nodes deployment knowledge.

1.3.2 Energy Efficient Coverage

Coverage is considered as the other measure of QoS of a sensor network. In coverage problems, the most significant factors are the ability of a network to observe a given area and

what are the changes that it detects in a given time frame. Energy-efficient sensing coverage extends system lifetime by leveraging on the redundant deployment of sensor nodes (Meguerdichian et al., 2001a). Within a couple of years, sensing coverage has become a well studied subject which provides either full coverage in both time and space (Cardei et al., 2005; Kumar et al., 2004; Tian and Georganas, 2003; Yan et al., 2003), coverage with guaranteed delay and connectivity (Cao et al., 2005; Hsin and Liu, 2004; Xing et al., 2005), or guaranteed intruders detection within a certain stealth distance (Gui and Mohapatra, 2004; Ren et al., 2005).

Effective coverage approaches for energy conservation in wireless sensor networks are scheduling sleep intervals for extraneous nodes while the remaining nodes stay active to provide continuous service. For the sensor network to operate successfully, the active nodes must maintain both sensing coverage and network connectivity. Furthermore, the network must be able to configure itself to any feasible degree of coverage and connectivity on a proportion or whole of the specified area in order to support different applications and environments with diverse requirements.

The work on sensing coverage can be broadly classified in terms of those that provide full coverage (single coverage and multiple coverage) and those that provide partial coverage. In full coverage, every point in the network is covered by at least one sensor. While such coverage is desirable in sensitive environments such as military surveillance, it requires a large number of sensors to be awake. In partial coverage, by contrast, only a subset of points in the sensor network are covered and, hence, the number of sensors that need to be awake is reduced. By degree of coverage, we mean the percentage of target area that is covered by working sensor nodes. In particular, an algorithm for partial coverage is especially desirable when it can provide a high degree of coverage while significantly increasing (more than doubling) network lifetime compared to the algorithms that provide full coverage.

Full sensing coverage is mandatory for sensor monitoring applications that require either

immediate response to detected events or information of all points in the sensing field. Full sensing coverage, however, is too expensive to support long-duration monitoring applications. More often those applications do not need zero responsive time or information at all points of the sensing field. Full sensing coverage provides over-qualified detection quality for these applications at the cost of exhausting network energy rapidly, who may be willing to sacrifice events detection probability or detection delay to some extent for increasing the network lifetime. A relaxed sensing coverage − partial coverage, where the sensing field is partially sensed by active sensors at any time − is a more appropriate approach to balancing intruders detection quality and battery power consumption.

Meguerdichian (Meguerdichian et al., 2001a) uses Voronoi diagram to propose algorithms in much favorable worst case running times scenarios in 2-D case. For identical nodes, $O(n \log n + nk^2)$ complexity by computing k_{th} order voronoi diagram; for communication range being changeable, runtime is $O(n \log n)$, which is much better than existing algorithms. Connectivity affects the robustness and achievable throughput of communication in a sensor network. CCP (Xing et al., 2005) provides geometric analysis of the fundamental relationship between coverage and connectivity, and presents a Coverage Configuration Protocol (CCP) that can dynamically configure the network to provide different feasible degrees of coverage requested by applications. This flexibility allows the network to self-configure for a wide range of applications and environments with diverse or changing coverage requirements.

Tian (Tian and Georganas, 2003) presents a 100% coverage algorithm by a node-scheduling scheme based on off-duty eligibility rules, which allows nodes to turn themselves off as long as the neighboring nodes can cover the area for them. Ye (Ye et al., 2003) achieves surveillance coverage by a probing mechanism. In this solution, after a sleeping node wakes up, it broadcasts a probing message within a certain range and waits for a reply. If no reply is received within a timeout, it takes the responsibility of surveillance until it depletes its energy. However, this probing-based approach has no guarantee on sensing coverage and

blind points can occur.

A partial coverage scheme allows sensor nodes to periodically wake up and go back to sleep. A node in sleep mode cannot sense events; its sensing capability is resumed after it wakes up. Therefore, the sensor network provides only a fraction of the maximal coverage of all the sensors. Battery power, however, is conserved for the nodes in sleep mode. Wang et al. (Wang and Kulkarni, 2005; Ren et al., 2007) present a partial coverage algorithm pCover for intruders detection by applying scheduling schemes. Gui et al. (Gui and Mohapatra, 2005) presents an on-demand coverage algorithm to track an intruder. In their work, the sensor nodes only be activated to cover the path the intruder travels. However, in their works, coverage degree problem is not considered or not fully addressed. Also, they assume the network is connected well.

In this dissertation, we present the energy-aware on-demand coverage and connectivity scheme for intruder tracking based on (Chen et al., 2001; Xing et al., 2005). We use the algorithm of SPAN (Chen et al., 2001) to build a whole network backbone for data communication, and we adopt the idea of CCP (Xing et al., 2005) to maintain a dynamic coverage degree on the area where the intruder presents, and do not consider other area. When the intruder moves to a new area, the old active nodes will go to sleep, while the new nodes which can cover the intruder will be activated to sense it and maintain a specified coverage degree.

1.3.3 Reliable and Robust Communication Links

Wireless sensor networks are vulnerable and easy to be attacked and compromised. *Security and responsiveness to attacks are also an important measure of QoS.* Currently, some pre-distribution key cryptography schemes are presented to establish pairwise keys between neighboring nodes in a hierarchical WSN (Perrig et al., 2002; Zhu et al., 2003; Perrig et al., 2001). These schemes offer nice scalability due to the nature of the hierarchical architecture.

However, they suffer from low resilience against nodes capture and high communication cost. Consequently, it is impractical to deploy in large scale networks with limited energy resources.

Other researchers present some key management schemes based on flat WSN. Random pairwise key scheme provides very good key resilience below a threshold, and is more scalable in the sense of efficient use of memory space of sensor nodes (Chan et al., 2003; Du et al., 2004a; Eschenaer and Gligor, 2002). It is based on Erdös and Rényi's random graph theory (Erdös and Rényi, 1959). Each sensor node stores a random set pairwise keys to achieve a probability p that two neighboring nodes are connected. If two nodes are not neighbors, they can use intermediary nodes to establish a path key. To obtain proper resilience, this scheme sacrifices key connectivity to decrease the storage usage. To further improve this scheme, location-based random pairwise key schemes are proposed (Du et al., 2003; Liu and Ning, 2003a,b; Liu et al., 2005), which take advantage of the location information to improve the key connectivity. The basic idea is to have each sensor to share pairwise keys with its closest neighbors. These schemes increase the resilience below a smaller threshold and incur small communication overhead for key establishment.

However, these schemes suffer from two major problems: high deployment density requirement and the degradation of resilience against nodes capture (Zhou et al., 2005, 2006). In the recent schemes, GKE (Zhou et al., 2005, 2006)and PIKE (Chan and Perrig, 2005) address the problem of high density requirement and obtain graceful resilience against nodes capture. In GKE, or Group-based Key Establishment scheme, the network is divided into small groups, and each sensor will be preloaded with unique pairwise keys shared with all other sensors in the same group. GKE assumes all the other groups are the neighbors of a randomly selected group, while is impractical, since one group should have at most eight geographically neighboring groups according to the groups definition and deployment. PIKE, or Peer Intermediaries for Key Establishment scheme, is a class of key-establishment protocols

that use one or more sensor nodes as trusted intermediary to facilitate key establishment. Since the intermediaries may not always be in the vicinity, PIKE requires network-wide communication to establish keys, which involves a relatively high communication overhead.

Here we propose **L**ayered **G**roup-based **K**ey **E**stablishment scheme (LGKE) to ensure the secured communication between the sensor nodes in a dynamic large scalable wireless sensor network. The sensor nodes are deployed in groups and each sensor node is preloaded with unique pairwise keys that are shared with each other within the same group. We construct the path keys via local communications for neighboring groups and build the keys via the group heads and the base station for non-neighboring groups.

Furthermore, we extend the Exclusion Basic System (EBS) technique (Eltoweissy et al., 2004) to achieve dynamic scalability. In a distributed environment, the existing EBS scheme can only add/delete nodes one by one. For a system containing $10,000$ nodes, adding or deleting one is trivial. While in our system, we can add/delete nodes as a group containing 100 to 500 nodes every time. Also, the base station can run an add/delete algorithm in real time, which means groups can be added/deleted after being deployed. The scalability is a compelling property of the LGKE scheme compared with other schemes.

1.4 Overview

Chapter 2 presents a two-layered sensor network model by incorporating the geographic deployment knowledge of the network, based on which we propose a Hybrid Energy-Efficient Routing Scheme (HERS) to extend the network lifetime by considering the max-min residual energy routing and the min-max cost routing. To the best of our knowledge, this is the first work to develop the hybrid energy-efficient routing protocol while minimizing the communication energy consumption using the nodes deployment knowledge. Chapter 3 presents the energy-aware on-demand coverage scheme. It only considers the interesting area and wakes

the necessary nodes to cover it. Other nodes still remain sleep, which decreases the unnecessary energy consumption. Chapter 4 proposes **L**ayered **G**roup-based **K**ey **E**stablishment scheme (LGKE) to ensure the secured communications between sensor nodes. The sensor nodes are deployed in groups and each sensor node is preloaded with unique pairwise keys that are shared with each other within the same group. We construct the path keys via local communications for neighboring groups and build the keys via the group heads and the base station for non-neighboring groups. Chapter 5 concludes the research presented in this dissertation, and points out the avenues for further research.

Chapter 2

Energy Efficient Routing for Wireless Sensor Networks

Transmission of data requires energy-aware QoS routing in order to ensure efficient usage of the sensors and effective access to the gathered measurements. This chapter presents a Hybrid Energy-Efficient QoS Routing Scheme (HERS) to extend the network lifetime by considering the max-min residual energy routing and the min-max cost routing. The organization is as follows: Section 2.1 describes the proposed two-layered network architecture and the deployment method. Section 2.2 provides the LP formulations of max-min residual energy and min-max cost optimization problems. Based on these, a hybrid energy-efficient QoS routing algorithm is developed. Section 2.3 shows the simulation results and comparisons with other similar protocols in terms of network lifetime. Finally, section 2.4 summarizes the chapter.

2.1 Network Architecture and Deployment Method

2.1.1 Scheme Architecture

It is critical to construct and maintain an efficient network topology. The sensor nodes in a multi-hop WSN can collaborate with each other to determine their transmission power and define the network topology through forming proper neighbor relations. This topology differs from the "traditional" network in which the nodes transmit with their maximal transmission power without considering the power efficiency.

We define a two-layered architecture of a WSN as shown in Fig. 2.1, which is different from traditional flat networks such as Fig. 2.2. In a flat network, the topology implicitly depends on the geographic locations of the sensors and each node transmits with its maximal transmission power. Although simple for small networks, this network architecture suffers from scalability. For example, adding a new node needs to inform the whole network to set up the keys for communication, which involves large communication energy consumption and is infeasible for the WSN because of its intrinsic resource limitation. On the other hand, a WSN is a scale free network (Barabási and Albert, 1999), in which there is a high probability that a node links to a node that has already a large number of connections. This means that for a group of sensor nodes, there could be one or two nodes that act as communicating switches. Hence, in this paper, we use the cluster heads as the switches and propose a two-payer WSN architecture Fig 2.1. The base station and cluster heads constitute a logical top layer while other sensor nodes constitute a logical lower layer. This two-layered group-based architecture assumption is reasonable and practical for the large-scale network. Each group contains 3 nodes acting as cluster heads communicating with the base station. If two groups are not neighbors, the sensors will communicate via cluster heads and the base station. More details of this two-layered architecture can be found at (Li et al., 2009).

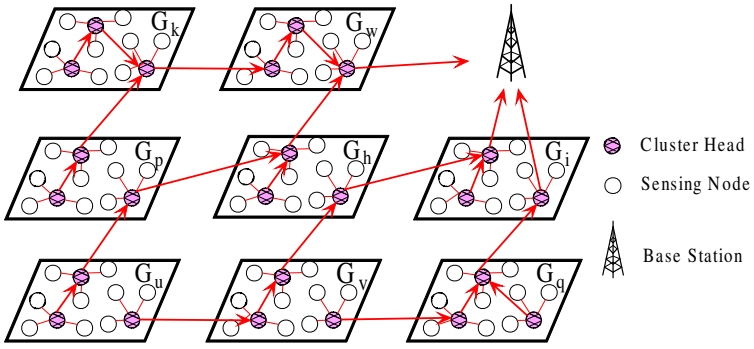

Figure 2.1: Architecture of two-layered sensor network. The base station and cluster heads constitute a logical top layer while other sensor nodes constitute a logical lower layer and is divided into groups.

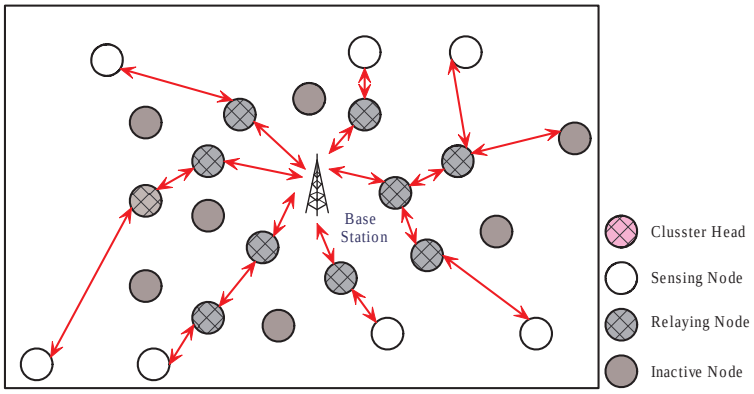

Figure 2.2: Traditional distributed sensor network without layers. The sensing nodes send the information to the base station via the relaying nodes. The relaying nodes just forward the information from their former nodes to their next nodes without processing it.

2.1.2 Deployment Knowledge

There are many methods to deploy sensor networks (Zhou et al., 2005; Du et al., 2004b). For example, an airplane can scatter sensor nodes over the battlefield. Based on the above two-layered architecture and the geographic locations of the sensor nodes, we can partition a large scale sensor network into groups. The following assumptions are made: 1) Nodes are deployed randomly and the groups are allocated according to the geographic location. 2) Once the sensor nodes are deployed, they are static. 3) the sensor nodes can be scattered in the targeted area with a certain probability density function (pdf), *e.g.*, the Gaussian distribution, and a deployment point refers to the location where a sensor node is deployed.

In practice, sensor nodes are usually deployed in groups to realize a specified function, *e.g.*, in (Werner-Allen et al., 2005a). Sensor nodes are deployed in different groups and each group is considered as a unit to sense temperature, acoustic pressure and seismic velocity. In the paper, we adopt such a group-based deployment approach, assuming that the target deployment region is a two-dimensional $ma \times na$ rectangular area. The target region is divided into $m \times n$ equal-sized subregions $R_{ij}(i = 1, \cdots m,$ and $j = 1, \cdots n)$ with each subregion $a \times a$ square meters. Each group, $G_{ij}(i = 1, \cdots m; j = 1, \cdots n)$, is deployed to the subregion R_{ij}. Let N be the total number of sensors. The N sensors are uniformly partitioned into N/g groups G_{ij} with the number of sensors in each group equal to g, and each group G_{ij} is randomly deployed into the subregions R_{ij}. The proposed deployment scheme relaxes the assumptions in (Du et al., 2004b) so that subregions are predefined and each sensor group is deployed into a specified subregion. Every sensor node k in group G_{ij} can be denoted as $[(i, j), k \in N]$. The deployment knowledge can be obtained as follows:

1. Divide the target region into subregions R_{ij};

2. N sensor nodes are partitioned into groups G_{ij} with g sensors in each group;

3. The node k in group G_{ij} has an identifier $[(i, j), k \in G_{ij}]$, and the distribution of the

sensors in each group follows a pdf $f(x, y|k \in G_{ij})$. A natural choice for the pdf is a two-dimensional Gaussian distribution which can be easily extended to other distributions such as the Weibull distribution or uniform distribution according to the application or the actually distribution of the sensors. In Section 2.3 we will investigate the influence of pdf on the maximal lifetime of the system;

4. Randomly distribute the groups G_{ij} into the subregions R_{ij}.

Since we model the sensor deployment distribution as a two-dimensional Gaussian distribution, the pdf for the sensor node k in the position $\mu = (x, y)$ is deployed in the group G_{ij} whose deployment coordination is (x_i, y_j). We have the probability density function:

$$f(x, y|k \in G_{ij}) = \frac{1}{2\pi\sigma^2} e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2} \tag{2.1}$$

where σ^2 is the variance parameter of the Gaussian distribution.

An example is shown in Fig. 2.3, where the target deployment region is divided into 4 sub-regions $R_{ij}(1 \leq i, j \leq 4)$. The N sensors are partitioned into 16 subgroups $G_{ij}(1 \leq i, j \leq 4)$, with each group containing g nodes and 3 cluster heads as a whole, and uniformly distributed into the sub-regions. In this deployment method, each group of sensors has eight randomly determined adjacent groups, *i.e.*, sensors in group G_{32} have neighbors either in group G_{22} or in groups G_{31}, G_{22}, G_{14}, G_{11}, G_{41}, G_{13}, G_{42}, G_{34}, and the sub-region R_{ij} is a square with the edge $a \times a$. For simplicity, we assume each group has 8 neighbors and ignore the groups locating at the edge and the corner of the targeted area, where they actually have 5 and 3 neighbor groups, respectively.

When the pdf $f(x, y|k \in G_{ij})$ of sensor nodes in the sub region R_{ij} is as in Eq. 2.1, the probability that the sensor deployment point will be out of the range of the preferred targeted sub-region is less than 0.03% if we empirically choose σ to ensure $6\sigma = a$, where σ is the standard deviation of the Gaussian distribution.

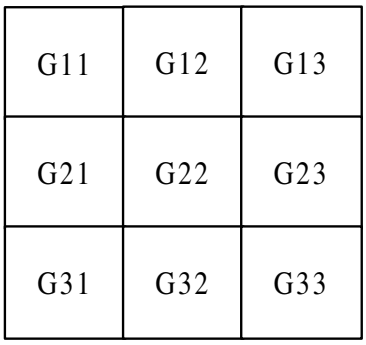

Figure 2.3: Random Group-based Deployment Method. The left figure is the sub-region structure, and the right figure shows the groups that are randomly arranged to the sub-regions, and the indexes ($1 \leq i, j \leq 4$).

2.2 Maximization of the Lifetime of Network

A sensor network can be modeled as a directed graph $G(N, L)$, where N is the set of all nodes plus the base station, and L is the set of all directed links (i, j), $i, j \in N$. Link (i, j) exists if and only if $j \in S_i$, where S_i is the set of all nodes that can be directly reached by node i with a certain transmit power level in its dynamic range. $SUBR_t(g, e)$ represents the directed subgraph of the t^{th} group SR_t, where g is the group size, or the number of sensor nodes in the group SR_t, e is the set of all the directed links (p, q), and $0 \leq t \leq N/g$, $p, q \in n$. The directed graph $CB(M, CL)$ is the set of all cluster heads and the base station, where M is the number of all the cluster heads and the base station, CL is the set of all the directed links (x, y), $x, y \in M$. Let CH be the set of cluster heads. Each node has an initial battery energy E_i. The transmission energy consumed at node i to transmit a data unit to its neighboring node j is denoted by e_{ij}^t and the energy consumed by the receiver j is denoted by e_{ij}^r.

There will be multiple commodities C in this network $G(N, L)$, where a commodity is defined as a group containing the source and destination nodes. For example, $SUBR_t(g, e)$ can be seen as a commodity, and the source nodes are the sensing nodes which will send the sensed information to the destination nodes. The destination nodes are the cluster heads,

which receive the sensed information, process or relay it. Another example is $CB(M, L)$, where source nodes are the cluster heads and the destination node is the base station.

We assume that each sensor node generates one data packet per time unit, which is termed as one round, and transmits the packet to the base station via intermediate nodes. For simplicity, every packet has a fixed size of z bits. At each round, each function unit, or group, will sense the necessary information and send it to the cluster heads for data aggregation and processing; then, the cluster heads will send the extracted information to the base station. The base station is assumed to be resource rich.

We define the system lifetime as the duration of time where all the function units perform properly. We assume that all the groups have to coordinate with each other to perform their designed functions. Hence, the lifetime of a WSN is equal to the time of the death of the first group after time zero. In our model, when all the 3 cluster heads use up the energy, the group will lose its function, and consequently, the system will die. The objective is to find the most energy efficient algorithm that can maximize the lifetime of the network.

2.2.1 Energy Model

Our general energy model follows the first order radio model (Heinzelman et al., 2000a). The energy expenditure per unit of information transmission from node i to j with the distance d_{ij} is given by:

$$e_{ij}^t = e^T + \epsilon_{amp} d_{ij}^\alpha \tag{2.2}$$

and

$$e_{ij}^r = e^R \tag{2.3}$$

where $e^T = e^R = 50nJ/bit$ is the energy consumed in the transceiver circuitry at the transmitter and the receiver respectively. The constant $\epsilon_{amp} = 100pJ/bit$ is the energy consumed

coefficient at the output transmitter antenna for transmitting one meter. The distance exponent α ranges from 2 to 4, with 2 being a short distance and 4 a long distance. We investigate the influence of α on the system's lifetime in a later section. Notice that we ignore the sensing energy consumption and computation energy consumption, because they are much smaller than the communication consumption $(1:10 \sim 1:300)$(Heinzelman et al., 2000a).

2.2.2 General Energy Efficient Modeling

Let $q_{ij}^{(c)}$ be the transmission rate of the commodity c from node i to node j, or a flow variable indicating the flow that c sends to the base station over the link (i,j), and let $\bar{q}_{ij}^{(c)}$ be the amount of information of commodity c transmitted from node i to node j until time T, i.e., $\bar{q}_{ij}^{(c)} = Tq_{ij}^{(c)}$. Let CH_t be the set of the cluster heads u, v, w in the t^{th} group $SUBR_t$ where $0 \leq t \leq N/g$ and u, v, w denote the 3 cluster head in the t^{th} group. Let r_i be the information requirement of node i.

Based on the lifetime definition and the system model, the lifetime of the cluster head $k \in CH_t$ under the given flow $q^{(c)}$ is given by:

$$T_k(q) = \frac{E_k}{\sum\limits_{j \in SR_t} e_{kj}^t \sum\limits_{c \in C} q_{kj}^{(c)} + \sum\limits_{j:k \in S_k} e_{jk}^r \sum\limits_{c \in C} q_{jk}^{(c)}} \tag{2.4}$$

where $\sum\limits_{j \in SR_t} e_{kj}^t \sum\limits_{c \in C} q_{kj}^{(c)} + \sum\limits_{j:k \in S_k} e_{jk}^r \sum\limits_{c \in C} q_{jk}^{(c)}$ is the energy consumption including transmitting and receiving consumptions per unit of time. Now the lifetime of our model can be defined as the minimal maximal lifetime of the cluster heads in some group, i.e.:

$$T_{sys} = min \max_{k \in CH_t} T_k(q), \quad t \in 1, \ldots, N/g \tag{2.5}$$

The objective is to find the flow that maximizes the system lifetime under the flow

conservation condition. Accordingly, the energy efficiency problem can be formulated into the following linear programming problem:

$$\texttt{Maximize } T \tag{2.6}$$

Subject to

$$\bar{q}_{ij}^{(c)} \geq 0, \forall j = 1, 2, \ldots, N, \forall c \in C \tag{2.7}$$

$$\sum_{j=1}^{N+1} e_{ij}^{t} \sum_{c \in C} \bar{q}_{ij}^{(c)} + \sum_{j=1}^{N} e_{ij}^{r} \sum_{c \in C} \bar{q}_{ij}^{(c)} \leq E_i, \forall i \in c \tag{2.8}$$

$$\sum_{j=1}^{N} \bar{q}_{ji}^{(c)} + r_i \times T = \sum_{j=1}^{N+1} \bar{q}_{ij}^{(c)}, \forall i \in c, i \neq j, \forall c \in C \tag{2.9}$$

where $N+1$ means that we consider the base station as a packet receiver, but it is not thought of as a sensor node. Eq.(2.7) conserves the feature that sensors have the ability to send packets to other nodes. Eq. (2.8) means that any energy consumption in data sending and receiving should respect the energy constraints. Eq. (2.9) means that every sensor generates the same amount flow as it takes in. As long as the variable T in (2.6) is considered as an independent variable, the above equations can be seen as a linear programming problem (Chang and Tassiulas, 2000). The solution to the above LP problem gives energy-aware QoS paths for the sensor nodes.

2.2.3 Energy Optimization Method

In this section we first present two traditional energy optimization routing algorithms for the energy-efficient communications: one is to maximize the minimal residual energy routing path (Li et al., 2001b; Toh, 2001; Zhang and Mouftah, 2006) and the other is to minimize the maximal energy consuming path (Chang and Tassiulas, 1999, 2000; Zhang and Mouftah,

2006). Then we provide our improved Hybrid Energy-efficient Routing Scheme (HERS) as a comparison.

- Max-min Residual Energy Routing Scheme (MMRERS)

To extend the lifetime of the network, MMRERS selects a routing path that contains the maximal residual energy among all the possible paths. This maximal residual energy routing path is determined in quantity by the node whose residual energy is minimal along the path. That is why the MMRERS is designed to address this max-min problem. Alternatively, the purpose of MMRERS is to find the path and maximize its remaining energy at a low communication overhead.

Initially,within the group SR_t, where $0 \leq t \leq N/g$, the set A contains only the cluster heads u, v, w. Let $SR_t - A$ denote the set of sensor nodes in SR_t that are not included in A. We define the *residual energy* of a pair (i, j) as $min(E_r[i] - e^t_{ij} * \bar{q}^{(k)}_{ij}, E_r[j] - e^r_{ji} * \bar{q}^{(k)}_{ji})$ where $i \in SR_t - A$ and $j \in A$. Intuitively, on adding a directed link (i, j) to A, the residual energy at the sensor node i is reduced by the energy consumed in transmitting a data packet from i to j. Consequently, the residual energy of node j will be reduced by the energy consumed in receiving a data packet. Among all pairs (i, j) such that $i \in SR_t - A$ and $j \in A$, the procedure chooses one with the maximum residual energy and includes the link (i, j) in A. The process is repeated until all sensor nodes in SR_t are included in A, and for other groups, the process will repeat in the same procedure.

For the top layer, a similar algorithm is adopted. Firstly, the set B contains only the base station, and set CH denotes the set of cluster heads. We use the same definition of residual energy, and add the maximal residual energy link (m, n) to the set B until all the cluster heads are included in B.

- Min-max Link Energy Consuming Routing Scheme (MMLERS)

To maximize the lifetime of a network, a direct method is to obtain a path which consumes minimal energy in communication. Consequently, the goal of the design is to minimize the energy depleting rate at individual nodes by selecting paths constituted by links with energy as low as possible.

According to our network architecture, every group contains 3 cluster heads, denoted by u, v, and w. Each cluster head needs to receive, process and transmit the information coming from every non-cluster head node within the group, which would consume the energy denoted by E. Therefore, in order to extend the lifetime T for each cluster head, our objective is to minimize the maximum energy E_{max} consumed by each non-cluster head node within a group. Let SR_t, $0 \leq t \leq N/g$, denote the t^{th} group. The following is the linear programming problem to minimize the maximal energy consumption:

$$\texttt{Minimize } E_{max} \tag{2.10}$$

Energy constraints:

$$\sum_{j \in SR_t} e_{ij}^t \sum_{c \in C} \bar{q}_{ij}^{(c)} + \sum_{j:i \in S_i} e_{ij}^r \sum_{c \in C} \bar{q}_{ij}^{(c)} + \sum_{k \in CH_t} \sum_{j \neq i \in SR_t} e_{jk}^t \sum_{c \in C} \bar{q}_{jk}^{(c)} + \sum_{k \in CH_t} \sum_{j \neq i \in S_i} e_{kj}^r \sum_{c \in C} \bar{q}_{kj}^{(c)} \leq E_{max}, \quad \forall i \in SR_t, \quad 0 \leq t \leq N/g \tag{2.11}$$

Flow constraints:

$$\sum_{j \in SR_t} \bar{q}_{ji}^{(c)} + r_i * T = \sum_{j \in S_i} \bar{q}_{ij}^{(c)} + \sum_{k \in CH_t} \sum_{j \neq i \in S_i} \bar{q}_{ik}^{(c)}, \forall i \in SR_t \tag{2.12}$$

where i is a sensor node within the group SR_t, and k is a cluster head in any other groups except SR_t where node i is in. For every group in the network G, the linear program minimizes the maximum energy consumed by any sensor node of this group, subjecting to

the energy constraint (2.11) and the flow constraint (2.12). Eq. (2.12) defines the flow conservation condition, where flow in equals to flow out.

For the top layer max-min problem, all the cluster heads construct another group, denoted by CH. It can be easily formulated in a similar linear equation format like Eqs. (2.6) - (2.9).

- Hybrid Energy-efficient Routing Scheme (HERS)

The MMRERS scheme suffers from the fact that the max-min residual energy links may consume more communication energy. Under this condition, the system will use up its energy soon and die. The MMLERS LP formulation can be solved polynomially by using source-based algorithms, such as Dijkstra's algorithm or Bellman-Ford algorithm (Ahuja et al., 1993). However, its max-min paths usually contain nodes whose communication consumption to other nodes is minimal or near to minimal. Therefore, these nodes have a high probability to be selected by these optimal paths and run out of battery energy quickly due to the heavy forwarding load while other nodes contain almost full battery energy. Thus, the MMLERS is energy inefficient also.

In order to find a QoS path for sending data, we present the hybrid energy-efficient routing scheme (HERS), which considers both the max-min residual energy as the link cost and the min-max communication energy consumption. The minimum energy path will be recalculated and redirected to other paths after every residual energy broadcast, and therefore, makes good use of all the possible nodes to construct optimal paths. Also, because the algorithm runs within the divided groups, the death of a non-cluster head sensor node won't affect the lifetime of the rest of the network. Moreover, the complexity of the HERS algorithm is low since it involves in the group size, denoted as g, and it considers the threshold values of residual energies of cluster heads. Multiple cluster heads extend the system's lifetime also.

To further investigate the impacts of residual energy broadcasting on the network lifetime, we adopt dynamic HERS, or D-HERS, which adjusts the broadcast interval, denoted by m,

dynamically, and static HERS, or S-HERS, which uses a fix broadcast interval.

We denote $E_r[i]$ to be the residual energy at the sensor nodes i. Initially, $E_r[i] = E$ for each node i in the network. We define the cost function $cost_{ij}$ that combines the unit data transmission energy consumption e^t_{ij} and e^r_{ji}, initial energy E_i and E_j, and the residual energy $E_r[i]$ and $E_r[j]$ as follows:

$$cost_{ij} = e^t_{ij}(E_i/E_r[i])^\beta + e^r_{ij}(E_j/E_r[j])^\beta \tag{2.13}$$

The link cost is denoted by the energy consumption from the sending node i to the receiving node j. We use the exponent index β to emphasize that residual energy does have impact. We will investigate the influence of the exponent index β on the system's lifetime in section 2.3.1.

The HERS algorithm, hence D-HERS and S-HERS which differ from each other by different residual energy broadcast policies, is describe as follows. HERS takes the deployment knowledge of N sensors and initial energy in each sensor as inputs. Output is the system's lifetime. Let T be the system's lifetime counted by the total rounds.

During the simulation, since each group resides within a relatively small region and each contains only g nodes, we treat each group as a sensing unit which executes the same functions. Thus, we take a random selection policy to select a node to act as an information source that sends the information to the destination, or the cluster head along the optimal path. After m rounds, another random selected node is selected as the source. Similarly, another cluster head is randomly selected as the destination. For D-HERS, the broadcast period m increments every p rounds, or time slots, where p is fixed value and is decided after the network is deployed. If m reaches the predefined threshold, it will not change any more.

Algorithm 1 HERS

1: Partition the N sensor network into $t \Leftarrow \lceil \frac{N}{g} \rceil$ groups $g_i, \ldots, g_t$, such that each group has g nodes and 3 cluster heads.
2: for each group, calculate the distances between each sensor node, and send the information to the cluster heads.
3: for each cluster head, calculate the distance from other cluster heads in other groups.
4: let initial broadcast period $m \Leftarrow 5$ and lifetime $T \Leftarrow 0$.
5: **if** $mod(T, m) == 0$ **then**
6: for each group, broadcast the residual energy to every other node.
7: for each cluster head, broadcast the residual energy to other cluster heads.
8: for each group, calculate the shortest link cost paths from sensor nodes to cluster heads using Eq. 2.13.
9: for each cluster head, calculate the shortest link cost paths to other cluster heads using Eq. 2.13.
10: based on the new paths, for each group, send packages from sensors to cluster heads, update the residual energy.
11: for each cluster head, send packages to other cluster heads and finally to the base station, update the residual energy.
12: $T \Leftarrow T + 1$.
13: **else**
14: based on the old paths, for each group, send packages from sensors to cluster heads, update the residual energy.
15: for each cluster head, send packages to other cluster heads and finally to the base station, update the residual energy.
16: **for each group**
17: **if** the residual energy of all the 3 cluster heads less than a threshold **then**
18: return T.
19: **else**
20: $T \Leftarrow T + 1$.
21: goto 5.
22: **end if**
23: **end if**

2.3 Simulation and Results

In this section, we compare the performance of the HERS algorithm with three other algorithms: the MLDA (maximum lifetime data gathering) and CMLDA (clustering-based MLDA) which incorporate the MMRERS and MMLERS schemes and are proposed by

(Kalpakis et al., 2002), and the LRS Leach (Heinzelman et al., 2000a; Lindsey and Raghavendra, 2002) in terms of network lifetime, since they outperform other existing energy efficiency schemes (Kalpakis et al., 2002).

We build the simulation environment by choosing a $50m \times 50m$ target area and vary the number of sensors in the network, i.e. the network size, N, excluding the cluster heads, from 40, 60, 80, to 100 respectively. Each group contains 10 non-cluster head sensor nodes and 3 cluster heads. Each sensor (including non-cluster heads or cluster heads) has an initial energy of 1 unit and generates packets of size 1000 bits per round, which is consistent with the settings in (Kalpakis et al., 2002). For the lower layer, each group has an optimal path whose information source is a non-cluster-head sensor node, and destination a cluster head. For the other nodes and cluster heads within this group, they all act as information relay units and just forward the whole packets they receive to the next hop. If the next hop is a cluster head which acts as the destination, it will process these packets and extract key information and then forward them to the base station via other cluster heads. This is the top layer's communication scheme, see next paragraph. Here we assume the extraction ratio is $1 : 10$, which means in each round, a cluster head receives 1000 bits from the source node, and the key information 100 bits are extracted and forwarded to the next hop.

As for the top layer, we use a similar scheme. Since N/g groups in the low layer has N/g optimal paths, and therefore has N/g information destinations, which are considered as information sources in the top layer, denoted by $P_t, t \in 1, \cdots, N/g$. The destination is the base station in the top layer. According to the extraction ratio, each path will transmit 100 bits to the base station. We process these transmission in queue in our simulation, although in fact, these N/g sources may send information to the base station simultaneously. This simplification will not affect the evaluation of the algorithms since practically the base station will process incoming information in the queue, otherwise, information congest will occur. The Bellman-Ford algorithm (Ahuja et al., 1993) is used to find the optimal path, and the

Boost Graph Library (Siek et al., 2001) is used in our Matlab simulation code.

We assume the broadcast message contains 100 bits in this simulation. The base station is located at location (25m, 150m), as (Kalpakis et al., 2002) set. We can use optimal position selection techniques to locate the base station such as in (Pan et al., 2005b), but it is out of our paper's scope. The energy model for the sensors is based on the first order radio model described in Eqs. 2.2 and 2.3.

We distribute the g sensor nodes and 3 cluster heads into each group according to the Gaussian distribution, as shown in Fig. 2.4, where the total number of sensor is $N = 40$, and group size is $g = 10$. Therefore, there are 4 groups and each group has a $25m \times 25m$ targeted area. We also simulate the lifetime of the network based on the uniform distribution of the sensor node in the targeted area when the network size N is 40. As shown in the Figs. 2.5, 2.6, and 2.7, the Gaussian distribution has a slightly better performance in the lifetime than that of the uniform distribution.

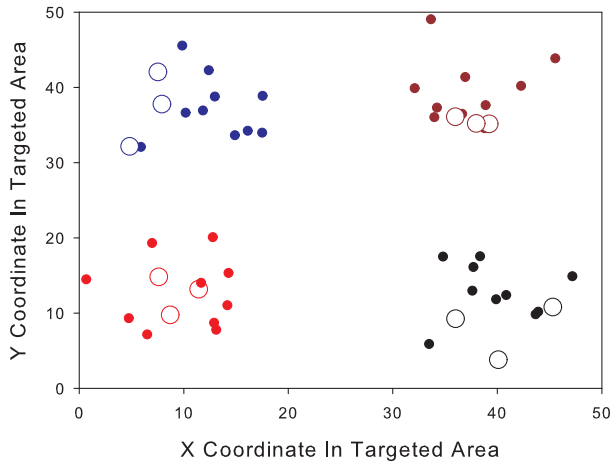

Figure 2.4: Sensor Deployment in the Gaussian distribution. Here solid dots represent sensor nodes and hollow circles represent cluster heads. In this target area $50m \times 50m$, 40 sensor nodes are deployed and are divided into 4 groups, each group contains 3 cluster heads and 10 sensor nodes within the $25m \times 25m$ targeted area. Within each sub region, the 10 sensor nodes and 3 cluster heads are distributed in Gaussian distribution with the central point of the sub region as mean point, the variance sigma is set to ensure that $\sigma = 25m/6$, under this situation, the probability that the sensor nodes are out of the sub region is less than 0.03%, the cluster heads are deployed randomly.

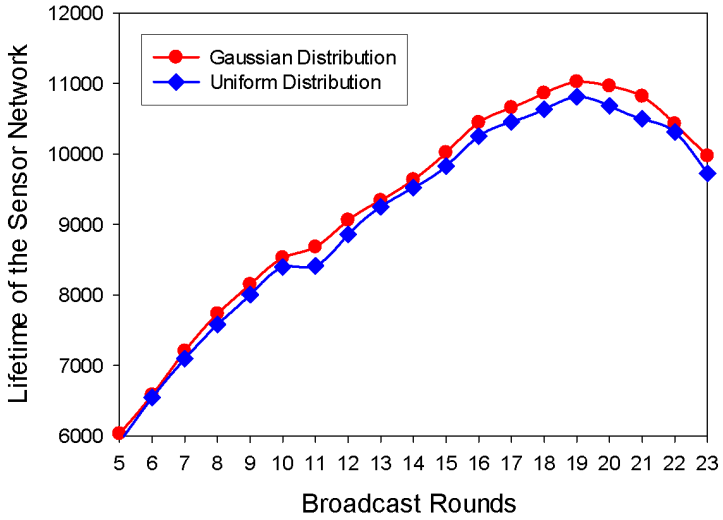

Figure 2.5: Rounds vs. lifetime for the Uniform distribution and the Gaussian distribution. Here the network size is $N = 40$, and is divided into 4 groups, with each groups 10 nodes. The distribution of the nodes will follow Fig. 2.4. Uniform distribution means within each group, the nodes are distributed in the targeted area with even probability.

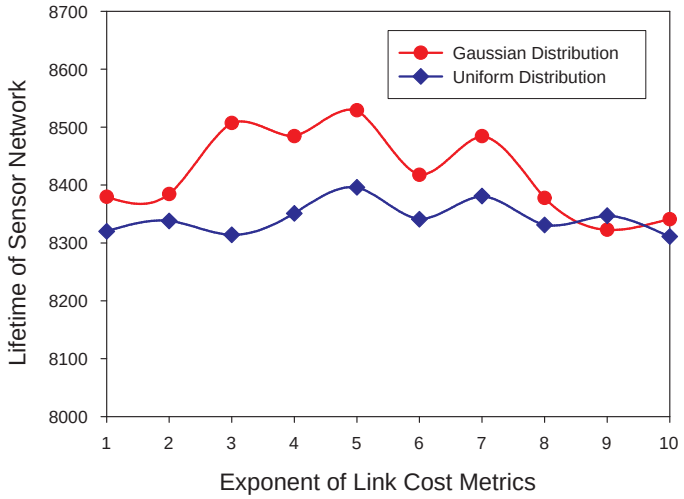

Figure 2.6: Exponent of Link Cost Metrics β vs. network lifetime. Here the network size is $N = 40$, and is divided into 4 groups, with each groups 10 nodes. The distributions of the nodes will follow Fig. 2.4. The broadcast period m is chosen to be 10. The key observation is that when $\beta = 5$, the mean lifetime is relatively higher and variance is smaller.

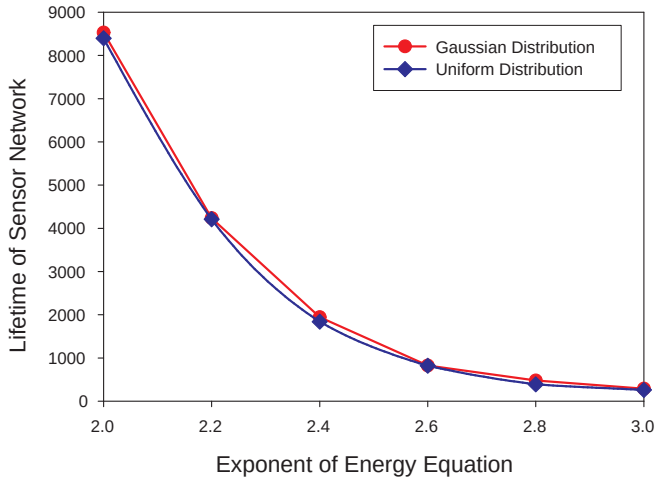

Figure 2.7: Exponent of energy equation α vs. network lifetime. Here the network size is $N = 40$, and is divided into 4 groups, with each groups 10 nodes, the distribution of the nodes will follow Fig. 2.4. The broadcast period m is chosen to be 10, and the exponent β is set as 5. When α changes from 2 to 3, the lifetime decreases dramatically.

2.3.1 Optimal Selection of Parameters

Since each node cannot know each other's residual energy level, a periodic broadcast containing the battery energy information should be sent to the neighbors. Obviously a smaller interval will provide more accurate battery information but will also consume more communication energy and incur information congestion. A longer broadcast interval, on the other hand, will consume less energy but leads to a non-optimal path selection. Fig. 2.5 shows the tradeoff between the lifetime of the network and the residual energy broadcast interval. The broadcast interval ranges from 5 to 23. When $m = 19$, the lifetime will approach to its maximum, 11025, which is less than $16,667$, since the optimal lifetime obtained from the MIP optimization is $16,667$. However, when the broadcast interval m increases, the computation time increases. We choose the broadcast interval $m = 10$ for S-HERS in the later discussion.

Fig. 2.6 shows the relation of the exponent index β in the link cost equation 2.13 with the lifetime of the network when the network size is 40. As we can see, the Gaussian distribution

has a better performance than the Uniform distribution, and the network has longer lifetime when β is chosen between 3 to 5. We select β to be 5 since we want the ratio of initial energy to residual energy to play a more significant role during the node selection, and when $\beta = 5$, the system has a relatively higher mean lifetime and smaller variance.

Fig. 2.7 shows the impact of the exponent α in the energy equation 2.2-2.3 to the lifetime of the network with the network size 40. When α increases from 2 to 3, the lifetime decreases dramatically since the energy exponent change will incur changes exponentially. To compare with (Kalpakis et al., 2002), the same energy exponent $\alpha = 2$ is chosen.

2.3.2 Performance of HERS

In this section, we will compare the performance of HERS with CMLDA, a clustering-based 2-level hierarchical protocol proposed by Kalpakis, Dasgupta and Namjoshi (Kalpakis et al., 2002), and LRS, a chain-based 3-level hierarchical protocol proposed by Lindsey, Raghavendra and Sivalingam (Lindsey et al., 2001) since these protocols have the similar hierarchical network structure and outperforms other protocols in terms of system lifetime. More details about the protocols can be found in (Lindsey et al., 2001), (Lindsey and Raghavendra, 2002) and (Kalpakis et al., 2002).

Table 2.1 is based on the fixed broadcast interval $m = 10$ for S-HERS, $\beta = 5$ and $\alpha = 2$,

Table 2.1: Simulation results for a $50m \times 50m$ sensor network

N	**S-HERS**	**D-HERS**	**MLDA**	**CMLDA**	**LRS**
40	8529	8529	6610	6512	5592
60	7912	8468	7174	7084	5872
80	7397	8471	7945	7809	6002
100	6804	8545	8290	8121	5526

S-HERS broadcast interval $m = 10$

D-HERS Broadcast interval $m = 10$ for $N = 40$, $m = 12$ for $N = 60$, $m = 14$ for $N = 80$, $m = 16$ for $N = 100$

for D-HERS, Broadcast intervals are chosen as $m = 10$ for $N = 40$, $m = 12$ for $N = 60$, $m = 14$ for $N = 80$, $m = 16$ for $N = 100$ separately. Some key observations can be obtained as follows.

The lifetime of a network obtained using the static broadcast interval (S-HERS) decreases when the network size increases. It is inferior to MLDA and CMLDA when network size is 80 or more, since when the network size increases, the number of the groups increases. Therefore, the communication within the top layer consumes more energy. S-HERS performs 1.10-1.31 times better than MLDA and CMLDA with a smaller network size. The lifetime obtained using S-HERS is significantly longer than that of LRS for all the network sizes, which is 1.23-1.53 times better than LRS.

D-HERS outperforms MLDA, CMLDA, and LRS at a relatively stable level of about 8500. The lifetime of the network using D-HERS is about 1.03-1.29 times longer than that of MLDA and CMLDA, and about 1.50 times longer than that of LRS.

Next we conduct a series of experiments with larger network sizes. The sensors are deployed in a $100m \times 100m$ area with Gaussian distribution. The network size, N, varies from 100 to 500. Each sensor has an initial energy $1J$, and the base station is located at $(50, 300)$. Each group has 10 sensor nodes. All the settings are the same as (Kalpakis et al., 2002). In this series experiments, we adopt D-HERS scheme since D-HERS has a better performance than S-HERS in large scale networks. The results are shown in Table 2.2. As we can see, D-HERS has a stable lifetime level about 7200, which is about 1.5 times longer than that of CMLDA, and about 2.5 times longer than that of LRS.

2.4 Summary

In this chapter, we presented a new energy-aware QoS routing protocol for sensor networks. We apply the deployment model to a two-layered sensor network, provide a general linear

Table 2.2: Simulation results for a $100m \times 100m$ sensor network

N	**D-HERS**	**CMLDA**	**LRS**
100	7212	3611	2458
200	7106	4512	2854
300	7535	5560	3212
400	7468	6142	3654
500	7194	6577	3596

programming formulation for the network lifetime maximization problem, and present our heuristics algorithms S-HERS and D-HERS to achieve an optimal network lifetime by finding the QoS paths for every sending node i to receiving node j. The simulation results showed that 1) for a small scale sensor network, the S-HERS can achieve network lifetime $1.1 - 1.3$ times better than other existing protocols; 2) for any size network, the D-HERS can obtain a factor of $1.1 - 2.0$ increase in network lifetime when compared to the same protocols; 3) the Gaussian distribution performs 1.20 times better than uniform distribution does.

We consider only a relatively large network size for computing the QoS routing in a distributed manner. However, in practical, there might be extremely large scale sensor networks, see, $\geq 10,000$. Our HERS algorithms based on a two-layered network have high complexity. A solution to decrease the computation complexity is to divide the network into more layers. The nodes number of each layer is confined in a specified range. Also, the algorithms presented are based on a centralized environment. However, a decentralized algorithm will be more practical since every node will make decisions based on the information it can obtain from its neighbors. These will be my next research topics.

Chapter 3

Energy Efficient Coverage for Wireless Sensor Networks

3.1 Introduction

Acquisition of data requires energy-aware QoS coverage and connectivity in order to ensure efficient coverage of the targets and effective data transmission for energy saving. In this chapter, we present the energy-aware QoS coverage and connectivity scheme for intruder tracking, as shown in Figure 3.1. The algorithm of SPAN (Chen et al., 2001) is used to build the network connectivity backbone for data communication. And CCP (Xing et al., 2005) is adopted to maintain a dynamic coverage degree.

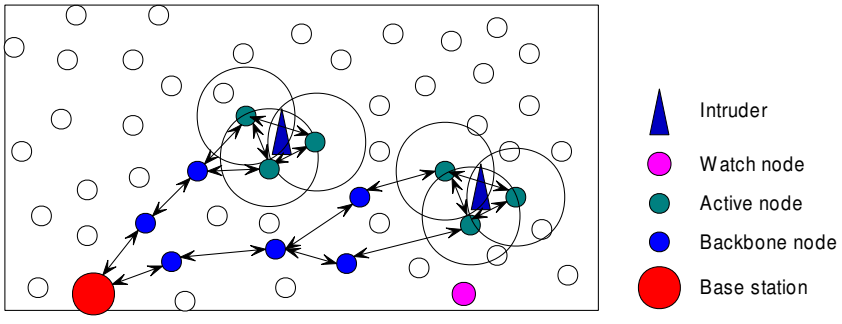

Figure 3.1: On-demand Tracking

This chapter is organized as follows: section 2 will briefly introduce the SPAN (Chen et al., 2001) algorithm of build communication backbones, section 3 addresses Coverage Configuration Protocol (Xing et al., 2005), the theoretic foundationary of our revised method to maintain coverage degree, section 4 will introduce our prediction model, section 5 will present our on-demand algorithm for intruders tracking, firstly single intruder algorithm is introduced, then an improved algorithm for multi-intruder detection algorithm is presented; section 6 will give the simulation settings and results, finally we give our summary.

3.2 Building Communication Backbones

SPAN (Chen et al., 2001) is a power saving technique for multi-hop ad hoc wireless networks that reduces energy consumption without significantly diminishing the capacity or connectivity of the network. SPAN builds on the observation that when a region of a shared-channel wireless network has a sufficient density of nodes, only a small number of them need be on at any time to forward traffic for active connections. It makes periodic local decisions on whether to sleep, or to join a forwarding backbone as a coordinator and participate in the forwarding backbone topology.

To build and maintain backbones, SPAN combines two basic algorithms: coordinator eligibility rule and coordinator withdrawal rule. According to the coordinator eligibility rule, a non-coordinator node should become a coordinator if it discovers, using only information gathered from local broadcast messages, that two of its neighbors cannot reach each other either directly or via one or two coordinators. This election algorithm does not yield the minimum number of coordinators required to merely maintain connectedness. However, it roughly ensures that every populated radio range in the entire network contains at least one coordinator. Also, nodes with more energy should volunteer as a coordinator more quickly.

SPAN also implements coordinator withdrawal rule. According to the rule, each coordinator periodically checks if it should withdraw. If every pair of its neighbors can reach other either directly or via some other coordinators or neighbors, the coordinator should withdraw to save energy consumption. In order to also rotate the coordinators among all nodes fairly, after a node has been a coordinator for some period of time, it marks itself as a tentative coordinator if every pair of neighbor nodes can reach each other via one or two other neighbors, even if those neighbors are not currently coordinators. A tentative coordinator can still be used to forward packets. However, the coordinator announcement algorithm treats a tentative coordinator as a non-coordinator. Thus, by marking itself as tentative, a coordinator gives its neighbors a chance to become coordinators.

SPAN improves 802.11 ad hoc power-saving mode to save energy and decrease packet delivery latency. To ensure that SPAN does not provide incorrect information because of topology changing, the MAC maintains a separate neighbor table. The MAC adds one bit to the MAC header of each packet to notify neighbors of its power saving status. SPAN modifies the MAC so each broadcast message must be explicitly advertised because most traffic in SPAN would be broadcast messages. SPAN also introduces a new small *advertised traffic window* in the MAC to allow a node in power saving mode to turn itself off at the end of the advertised traffic window until the next beacon period. A more detailed introduction about SPAN can be referred to (Chen et al., 2001).

3.3 Coverage Configuration Protocol

For the sensor network to operate successfully, the active nodes must maintain both sensing coverage and network connectivity. Furthermore, the network must be able to configure itself to any feasible degree of coverage and connectivity in order to support different applications and environments with diverse requirements. We implement our on-demand coverage scheme

based on CCP (Xing et al., 2005). A more detailed introduction about CCP can be referred to (Xing et al., 2005), here we only present the important theorems about coverage and connectivity.

Theorem 3.3.1. *For a set of sensors that at least 1-cover a convex region A, the communication graph is connected if* $R_c \geq 2R_s$.

Theorem 3.3.1 is the corresponding Theorem 1 in (Xing et al., 2005), where R_c is the sensor's communication radius and R_s is the sensor's sensing radius. Theorem 3.3.1 establishes a sufficient condition for a 1-covered network to guarantee 1-connectivity.

Theorem 3.3.2. *For a set of nodes* K_s*-cover a convex region A forms a* K_s *connected communication graph if* $R_c \geq 2R_s$.

Theorem 3.3.2 is the corresponding Theorem 2 in (Xing et al., 2005), where K_s is the desired coverage degree. Theorem 3.3.2 builds a relationship between the degree of coverage and connectivity. This result is important for applications that require degree of coverage of connectivity larger than one.

Based on the theorems 3.3.1 and 3.3.2, Wang *et. al.*(Xing et al., 2005) present coverage configuration protocol if $R_c \geq 2R_s$. Each node executes an eligibility algorithm to determine whether it is necessary to become active or not. Given a requested coverage degree K_s, a node is ineligible if every location within its coverage range is already K_s-covered by other active nodes in its neighborhood. For example, assume the nodes covering the shaded circles in Figure 3.2 are active, the node with the bold sensing circle is ineligible to be activated for $K_s = 1$, but eligible for $K_s > 1$.

Before presenting the eligibility algorithm, we defined the following notations.

1. A point $p \in$ coverage region A is called an intersection point between nodes u and v, that is, $p \in u \cap v$, if p is an intersection point of the sensing circles of u and v.;

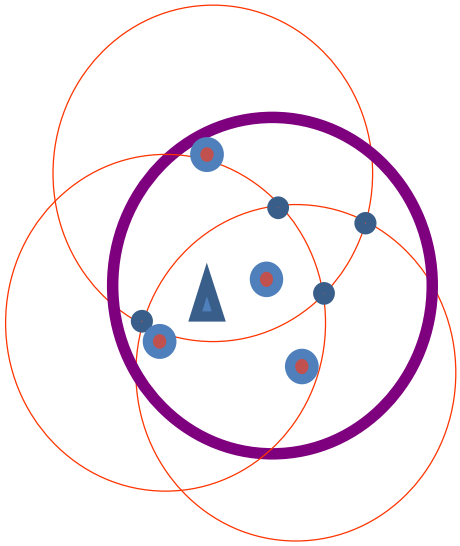

Figure 3.2: K_s Coverage. An example of 1-coverage eligibility. The node with the bold sensing circle is ineligible since every point in its sensing range is covered by other nodes.

2. A point p on the boundary of the coverage region A is called an intersection point between node v and A, that is, $p \in v \cap A$ if $|pv| = R_s$, where R_s is the sensing range of nodes v and u.

Theorem 3.3.3. *A convex region A is K_s-covered by a set of nodes if (1) there exist in region A intersection points between nodes or between nodes and A's boundary; (2) all intersection points between any nodes are at least K_s-covered; and (3) all intersections points between any node and A's boundary are at least K_s-covered.*

Theorem 3.3.3 is the corresponding Theorem 4 in (Xing et al., 2005). According to this theorem, we can transform the problem of determining the coverage degree of a region to the simpler problem of determining the coverage degrees of all the intersection points in the same region. A node is ineligible for turning active if all the intersection points inside its sensing circle are at least K_s-covered. To find all the intersection points inside its sensing circle, a node v needs to consider all the nodes in its sensing neighbor set, $SN(v)$. $SN(v)$ includes all the active nodes whose sensing circles intersect the sensing circle of v, that is,

$SN(v) = \{$active node u$\}$, where $|uv| < 2 * R_s, u \neq v$. If there is no intersection point inside the sensing circle of node v, v is ineligible when there are K_s or more nodes that are located at node $v's$ position.

3.4 Prediction Model

In order not to lose the track of intruders, some nodes in the future path of intruders need to be activated in advance. Also, some nodes which are sensing the intruders need to set a time to sleep for energy saving if the future positions of the intruders are not within the nodes' sensing range any more. Here we present a linear prediction model to roughly estimate the future positions of the objects. Some more accurate but complicated prediction models, such as ARIMA (Box and Jenkins, 1991; Li et al., 2006), are presented. However, considering the limited computation ability and resource constraints of sensor nodes, we adopt this simple but efficient model instead. Figure 3.3 A shows actual movement of the intruder (solid curve) and the predicted position (arrow head) using linear prediction model. As long as we set a proper time span, the curve can be approached by some piecewise lines.

As shown in Fig. 3.3-A, we know the previous position of an intruder (x_1, y_1), the current intruder's position (x_2, y_2), we can get the future position (x_3, y_3) using Eq. 3.1

$$\begin{aligned} x_3 &= 2 * x_2 - x_1 \\ y_3 &= 2 * y_2 - y_1 \end{aligned} \tag{3.1}$$

Eq. 3.1 means the time span of sampling (x_1, y_1) and (x_2, y_2) is the same as the time span of sampling (x_2, y_2) and (x_3, y_3), and the intruder will not make a weird movement, such as turning around suddenly, and continue moving at the old direction.

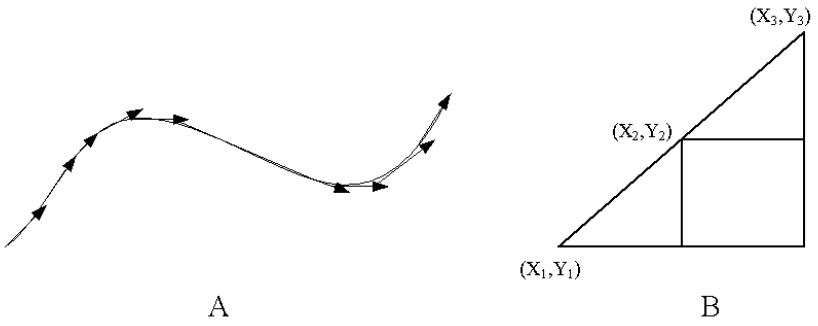

Figure 3.3: Prediction Model.

3.4.1 Activation in Advance

To track a fast moving intruder, node v needs to be activated in advance in order not to losing the tracking. We will broadcast an activation packet in advance containing the intruder's future position if the intruder runs out of node $v's$ sensing range. As shown in Figure 3.4, node v knows its central position (x_1, y_1), and the intruder's current position (x_2, y_2), which is out of node $v's$ sensing range R_s, and the intruder's future position (x_3, y_3), which is within its sensing range. Node v needs to determine a delay time t_{da}. After t_{da}, node v will be activated. We use the distance of $d_1/cos(\alpha)$ to estimate the actual distance the intruder needs to go from its current position, where α is the angel between the line $\overline{(x_1, y_1)(x_2, y_2)}$ and the line $\overline{(x_2, y_2)(x_3, y_3)}$ and can be calculated by simply geometry knowledge. The node knows the speed p the intruder moves at. Therefore, we have the delay time in Eq. 3.2.

$$t_{delay} = (||(x_1, y_1)(x_2, y_2)|| - R_s)/(cos(\alpha) * speed) \tag{3.2}$$

To avoid the situation that more than one nodes have the same delay time to be activated and then violate the coverage degree maintenance, we add a random time t_r to the delay time. Finally t_{da} is calculated in Eq. 3.3.

$$t_{da} = t_{delay} + t_r \tag{3.3}$$

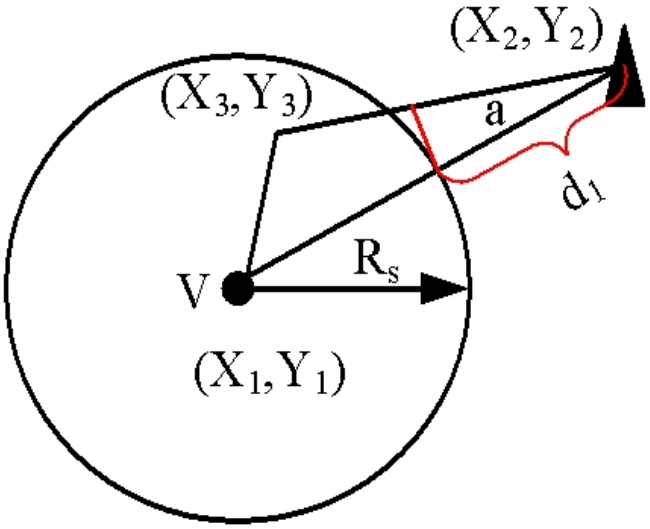

Figure 3.4: Prediction Model in Activation

3.4.2 Deactivation for Energy Saving

When the intruders move out of node $v's$ sensing range, node v needs to arrange a delay time to be deactivated and go to sleep. As shown in Figure 3.5, node v knows its central position (x_1, y_1), and the intruder's current position (x_2, y_2), which is within node $v's$ sensing range R_s, and the intruder's future position (x_3, y_3), which is out of $v's$ sensing range. Node v needs to determine a delay time t_{dd}. After t_{dd}, node v will be deactivated. Using the similar method in Section 3.4.1, we estimate the distance that the intruder needs to go out of node $v's$ sensing range as $||(x_2, y_2)(x_3, y_3)|| - d_1/cos(\alpha)$, where α is the angel between the line $\overline{(x_1, y_1)(x_2, y_2)}$ and the line $\overline{(x_2, y_2)(x_3, y_3)}$ and can be calculated by simply geometry knowledge. And $d_1 = ||(x_1, y_1)(x_3, y_3)|| - R_s$. Therefore, we have the delay time t_{dd} is:

$$t_{dd} = (||(x_2, y_2)(x_3, y_3)|| - d_1/cos(\alpha))/(speed) + t_r \tag{3.4}$$

where t_r is a random time, which is added to avoid that more than two nodes are deactivated at the same time. At that situation, the coverage degree maintenance is violated.

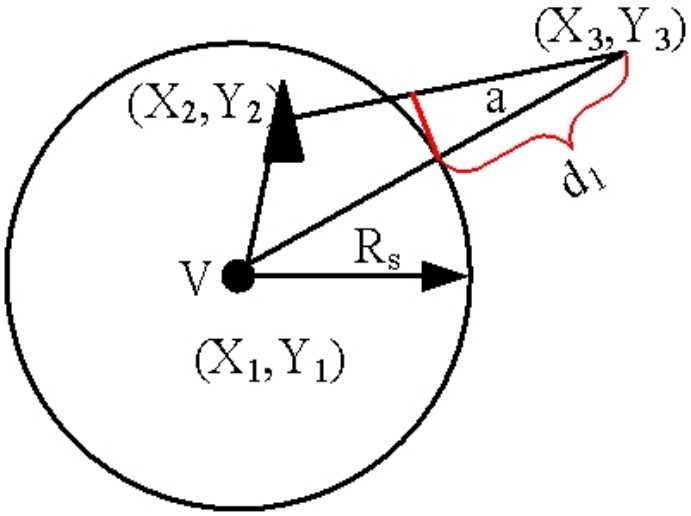

Figure 3.5: Prediction Model in Deactivation

3.5 On-demand Algorithm Implementation

Our algorithm includes communication backbones construction, local coverage maintenance, intruder track and data report. Since the communication backbone algorithm is based on SPAN (Chen et al., 2001) and introduced in section 3.2, we will skip it and focus on the last three parts in detail. Firstly a single intruder detection algorithm is introduced, and followed by an improved multiple intruders detection algorithm.

3.5.1 Single Intruder Detection Algorithm

This algorithm directly uses the prediction model introduced in Section 3.4, and includes coverage degree maintenance algorithms and intruder tracking.

Firstly we need to check the coverage degree for each sensor. As introduced in Section 3.3, to obtain the coverage degree of each sensor. We only need to check the coverage degree of the intersection points within its sensing range.

Check Coverage

Algorithm 2 is the pseudo-code to run in every node, or in node u.

Algorithm 2 intersection-calculation

```
1: for all nodes v, and v ≠ u do
2:    if ||vu|| ≤ R_s then
3:       store-neighbor(v,neiborlist)
4:    end if
5: end for
6: for all nodes v, w ∈ neiborlist do
7:    calculate-intersection(v, w, sec)
8:    if ||sec, u|| ≤ R_s then
9:       store-intersection(sec, seclist)
10:   end if
11: end for
12: for all nodes v ∈ neiborlist do
13:   bound-intersection(v,sec)
14:   if ||sec, u|| ≤ R_s then
15:      store-intersection(sec,seclist)
16:   end if
17: end for
18: return seclist
```

Every node u has a neighborhood list called *neiborlist* which stores $u's$ neighbors who have intersection points with u. Every node u has also an intersection points list called *seclist*, which stores the intersection points between $u's$ neighbors, or $u's$ neighbors and boundary.

Every intersection point s has a member called *msr* representing its current coverage degree. Algorithm 3 is the pseudo-code to check coverage degree. Node u knows the position p of the intruder, and p is in $u's$ sensing range, otherwise u will not run the algorithm.

Algorithm 3 check-coverage

```
1: for all s ∈ seclist do
2:    if s.msr ≤ k_s then
3:       return 1
4:    end if
5: end for
6: return 0
```

Return 1 means the coverage degree is less than k_s, and node u needs to be activated. Return 0 means the intruder is already k_s-covered, node u needs to do nothing.

Coverage Degree Updating

Whenever a node v is activated or deactivated, the coverage degree of the intersection points within $v's$ sensing range will change correspondingly. However, $v's$ neighbors do not know the coverage degree changes, so the node v needs to send packets to its neighbors about its status change. For example, node u receives a packet from node v indicating node $v's$ status is changed. Algorithm 4 is the pseudo-code to update node $u's$ coverage degree.

Algorithm 4 update-coverage

```
1: for all sec ∈ seclist do
2:   if ‖sec, u‖ ≤ R_s && ‖sec, v‖ ≤ R_s then
3:     if v is activated then
4:       sec.msr ++
5:     end if
6:     if v is deactivated then
7:       sec.msr - -
8:       if sec.msr≤ 0 then
9:         sec.msr = 0
10:      end if
11:    end if
12:  end if
13: end for
```

Intruder Tracking

Here we introduce the whole picture of our single intruder tracking algorithm. Every node, *i.e.*, u will periodically check the position of the intruder, and based on this information, u will do some operations such as making predictions, becoming activated or deactivated. Variable denoted by *-is-sensing* indicates that node u is activated or not. Following is the algorithm.

Algorithm 5 intruder-track

```
1: get-intruder-position(p)
2: if ‖p, u‖ ≤ R_s && -is-sensing then
3:    get-future-position(f)
4:    if ‖f, u‖ ≥ R_s then
5:       get-deactivation-delay(dd)
6:       after dd, go to sleep
7:       send packets to every neighbor v, v ∈ neiborlist
8:       -is-sensing = false
9:    end if
10: else if ‖p, u‖ ≤ R_s && !-is-sensing && check-coverage() then
11:   get-activation-delay(da)
12:   after da, wake up
13:   send packets to every neighbor v, v ∈ neiborlist
14:   -is-sensing = true
15: else if ‖p, u‖ ≥ R_s && -is-sensing &&!-is-coordinator then
16:   get-deactivation-delay(dd)
17:   after dd, go to sleep
18:   send packets to every neighbor v, v ∈ neiborlist
19:   -is-sensing = false
20: else
21:   get-future-position(f)
22:   if ‖f, u‖ ≤ R_s then
23:      get-activation-delay(da)
24:      after da, wake up
25:      send packets to every neighbor v, v ∈ neiborlist
26:      -is-sensing = true
27:   end if
28: end if
```

For the current intruder's position p, every node u has four kinds of conditions: 1). u is activated and the intruder is within $u's$ sensing range. In this case, sensor u predicts the intruder's next possible position f, if f is out of sensor $u's$ sensing range, sensor u calculates its delay time dd. After time dd, sensor u will go to sleep. 2). sensor u is not activated, if the coverage degree $\leq k_s$, then sensor u calculates a delay time da, after time da, sensor u is activated. 3). u is activated but the intruder is out of $u's$ sensing range, and the intruder is within $u's$ sensing range, then sensor u calculates a delay time dd, after time dd, sensor u will go to sleep. 4). sensor u is not activated and the intruder is out of $u's$ sensing range. In

this case, sensor u predicts the intruder's next possible position f, if f is within sensor $u's$ sensing range, sensor u calculates its delay time da. After time da, sensor u will be activated.

Note that NS2 has an internal timer to handle the delay time. When the delay time eclipses, NS2 automatically calls functions to activate or deactivate the corresponding sensor.

3.5.2 Multiple Intruders Detection Algorithm

This algorithm involves in more sensor nodes' cooperation since the intruders may enter the sensor network from different positions. For example, in single intruder detection algorithm, if the intruder i_1 moves out of one non-coordinator sensor's (n_1) sensing range and will not return back based on the prediction model, n_1 will go back to sleep. However, for multiple intruders detection algorithm, n_1 may be monitoring another intruder i_2 and will last for some time. In this case, simply let n_1 sleep will violate the coverage degree of i_2. Thus we need to adjust the corresponding algorithms for coverage degree maintenance and intruder tracking. Also, in the intruder tracking algorithm, some data specifying the intruders must be added to the transferred packages from the sensor n_1 to the base station for identification, otherwise the base station cannot distinguish the intruders. Note that all these adjustments increase the energy consumption for this sensor network.

For simplicity, we assume the coverage degree for every intruder is the same. As discussed in section 3.5.1, we do not need to change the algorithms for coverage checking, coverage maintaining and coverage updating, the only changed algorithm is the whole picture of intruder tracking algorithm.

Intruder Tracking

As we said above, if a sensor u is monitoring more than one intruders at the same time, and one intruder moves out of $u's$ sensing range, the sensor u cannot go to sleep since it is monitoring other intruders. In this situation, the intruders tracking algorithm is changed

correspondingly.

Every node periodically scans its sensing range to check whether there are intruders within its sensing range, and based on this information, it will do some operations such as making predictions and becoming activated or deactivated. Variable denoted as *-is-sensing* indicates that node u is activated or not.

Node u has a position list called *plist* to store the intruders' positions. For each position p in *plist*, u has four kinds of conditions: 1). u is activated and the intruder is within $u's$ sensing range. In this case, sensor u predicts the intruder's next possible position f, if f is out of sensor $u's$ sensing range, sensor u calculates its delay time dd and add it to delay time list $ddlist1$, else, the function returns since node u must keep active to monitor this intruder. 2). sensor u is not activated, and the intruder is within $u's$ sensing range, and the coverage degree $\leq k_s$, then sensor u calculates a delay time da, after time da, sensor u is activated, then return. 3). u is activated but the intruder is out of $u's$ sensing range, if u is not a coordinator, then sensor u calculates a delay time dd and add it to the delay time list $ddlist1$. 4). sensor u is not activated and the intruder is out of $u's$ sensing range. In this case, sensor u predicts the intruder's next possible position f, if f is within sensor $u's$ sensing range, sensor u calculates its delay time da. After time da, sensor u will be activated, then return.

If sensor u is not needed to be deactivated, the delay time list $ddlist1$ must be empty. Otherwise sensor u finds the maximal delay time T_{delay} from $ddlist1$, after this time, u will be deactivated.

3.5.3 Data Report to Base Station

When there are intruders within node $u's$ sensing range, and u is activated, u will periodically send the intruders' positions to the base station. For multiple intruders algorithm, we assume each sensor can know every intruder's identification and therefore, a global ID identifying

Algorithm 6 Intruders tracking

```
1: get-intruders-position(plist)
2: for all p ∈ plist do
3:    if ‖p, u‖ ≤ R_s && -is-sensing then
4:       get-future-position(f)
5:       if ‖f, u‖ ≥ R_s then
6:          get-deactivation-delay(dd1, ddlist1)
7:       else
8:          return
9:       end if
10:   else if ‖p, u‖ ≤ R_s && !-is-sensing && check-coverage() then
11:      get-activation-delay(da)
12:      after da, wake up
13:      send packets to every neighbor v, v ∈ neiborlist
14:      -is-sensing = true
15:      return
16:   else if ‖p, u‖ ≥ R_s && -is-sensing && !-is-coordinator then
17:      get-deactivation-delay(dd2, ddlist1)
18:   else
19:      get-future-position(f)
20:      if ‖f, u‖ ≤ R_s then
21:         get-activation-delay(da)
22:         after da, wakeup
23:         send packets to every neighbor v, v ∈ neiborlist
24:         -is-sensing = true
25:         return
26:      end if
27:   end if
28: end for
29: if ddlist1 not empty then
30:   find the maximal T_delay ∈ ddlist1
31:   after T_delay , go to sleep
32:   send packets to every neighbor v, v ∈ neiborlist
33:   -is-sensing = false
34: end if
35: return
```

an intruder is added to the packet. Note that, this assumption is reasonable since once one intruder enters the network, at least one sensor can detect and identify this intrusion. This information is sent back to the base station through communication backbones and other noncoordinator sensors, therefore, the whole network can identify this intruder and attach

an ID to it.

3.6 Environment Setting and Results

We simulate our scheme in the NS2 network simulator using CMU wireless extensions. We use a $750m * 750m$ simulation network with 120 nodes. Nodes in our simulations use radios with a 2Mbps bandwidth and 250 meters nominal radio range, and 50 meters sensing range. Twenty nodes send and receive traffic. Each of these nodes send a CBR flow to another node. and each CBR flow sends 128 byte packets.

All the sensor nodes never move except the intruders. Source and destination at all times so they send and receive packets at higher throughput. However, they do not participate in coordinator elections. The ID of the base station is set to 5. The 120^{th} node is set to be the intruder and active at all times for single intruder tracking simulation, while the 197^{th} to 120^{th} nodes are set to be intruders for multiple intruders tracking simulation. As Fig. 3.6 shows, the coordinators rotate during a 600 second simulation.

3.6.1 Path Tracking Results

As shown in Fig. 3.7, the base station receives the intruder's positions when it moves with a coverage degree 2. Actually, the intruder moves from $(746, 735)$ to $(236, 157)$ to $(189, 510)$. When it arrives at the destination, it will stop 60 seconds and moves on. So there are more points at the area where it stops.

3.6.2 Energy Comparison

The main purpose of our on-demand coverage problem is to provide energy efficient QoS coverage. Here we present two baselines for our comparison: Virtual Patrol (Gui and Mohapatra, 2005) and CCP (Xing et al., 2005).

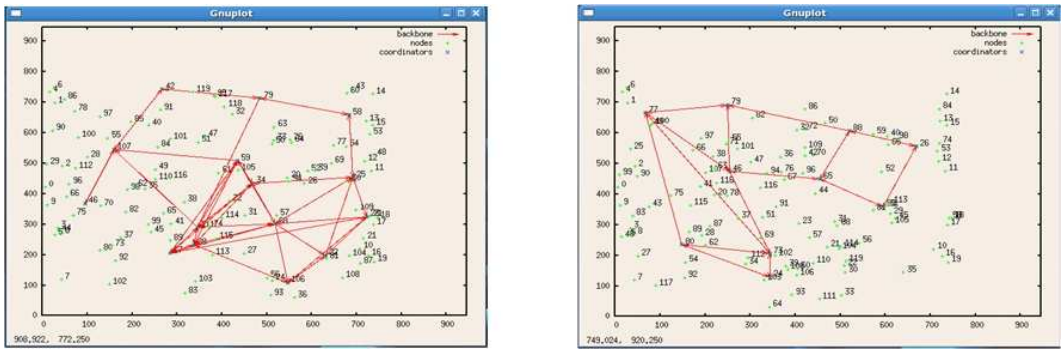

Figure 3.6: Communication Backbone Based on SPAN

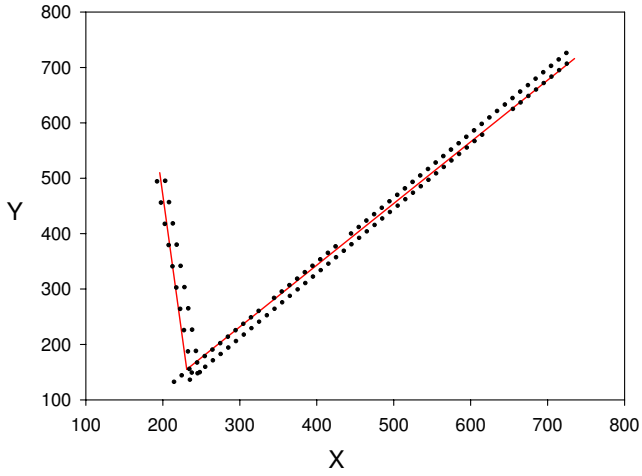

Figure 3.7: Intruder Tracking

Virtual Patrol

Virtual Patrol (Gui and Mohapatra, 2005) is another on-demand coverage algorithm. As shown in Fig 3.8, the main idea is to activate all the nodes along the single intruder moving path and report data to the base station during different duties. But they did not consider the communication connectivity. Also, when the intruder moves out of the activated nodes, these nodes will still remain active.

We make some improvements for this algorithm. Firstly, we add the communication backbones using SPAN, secondly, we deactivate the nodes when the intruder moves out of their sensing ranges.

CCP

As introduced in Section 3, CCP (Xing et al., 2005) is another algorithm considering both the connectivity and coverage for wireless sensor networks. CCP is designed for general applications to cover the *full* deployed area. Our algorithm is only interested in the area where the intruder presents, and therefore is a *partial* coverage algorithm. Naturally our algorithm can save more energy, as shown in Fig. 3.9.

As we can see, the remain energy of our algorithm is higher with coverage degree MSR 1. When MSR is 6, there are not so much difference since our algorithm activates all the nodes which can cover the intruder. The original VP and improved VP have not much difference. Because the wakened nodes do not need to send packets to the base station, while the communication energy consumption play a huge role in wireless sensor networks.

3.6.3 Mean Delay Time until Detection

Mean delay time is one QoS evaluation metric for energy-efficient QoS coverage scheme. As we can see, when the intruder moves out of the current scope of the sensing nodes, and the new nodes need some random time to be activated based on the prediction model. The

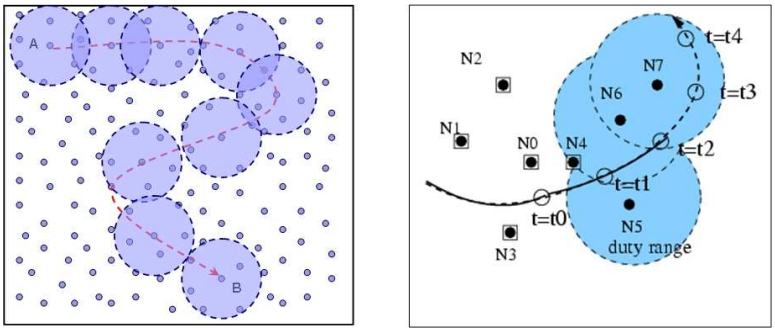

Figure 3.8: Virtual Patrol Algorithm

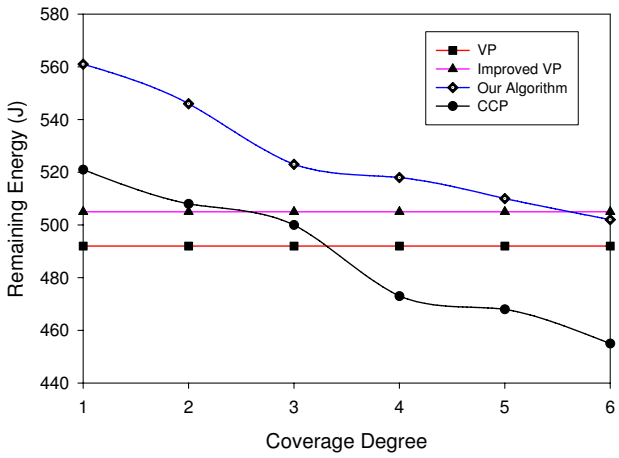

Figure 3.9: Energy Comparison with Virtual Patrol and CCP

definition of mean delay time until detection is the time span from the intruder moves out of one node $u's$ sensing scope to the nearest sensor v which will be activated and ready to sense the intruder. This metric is critical since if this delay is too long, we may lose the track of the intruder. We run 3 simulation scenarios with single intruder and 4 intruders at the speed of $20m/s$. In this case, we exclude the scenarios that a node is assigned a delay time to be activated, however, the intruder does NOT actually move into its sensing range. Because in the case, this node goes back to sleep in stead of being activated.

Fig. 3.10 shows the average delay time at different simulation scenarios. As we can see, at the same condition, in multiple intruders scenario, the sensors have smaller mean delay time. The reason is that when one intruder i_1 runs out of a sensor $u's$ scope and is moving toward to the sensor v, while another intruder i_2 already run into the sensor $v's$ sensing scope and activates v. This feature is critical in battle field when a sensor network needs to monitor huge amount of enemies. Under this situation, the sensor network is more efficient.

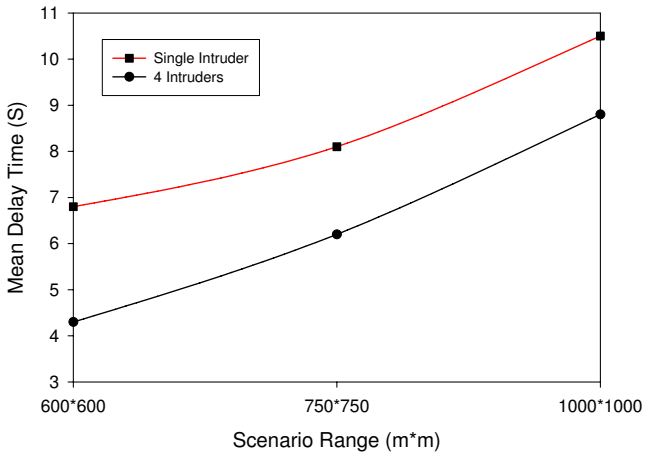

Figure 3.10: Mean Delay Time (S) at different scenarios. Where we have 4 intruders and each intruder has the same speed $20m/s$. The single intruder also runs at this speed.

3.6.4 Responsive Time

Responsive time is the other QoS evaluation metric for energy-efficient QoS coverage scheme. It is because in several applications, such as battlefield, energy efficient is not a critical concern while the responsive time between an intruder is detected and the base station receives the intruder's position is more important. In our algorithm, we take the average of all the responsive time obtained from all the nodes which detects the intruders. Fig. 3.11 shows the responsive time v.s. the coverage degree for single intruder and multiple intruders at different scenarios $600mm$, $750mm$ and $1000mm$. As we can see that with the increase of coverage degree and decrease of the simulation area, the average responsive time increases. It is reasonable because a smaller area means a dense sensors deployment and quick information transfer. The average responsive time of multiple intruders detection is a little smaller than that of single intruder detection. That is because in multiple intruders detection, more sensors are involved in intruders detection, which enables a faster information transfer path, and therefore, is more efficient.

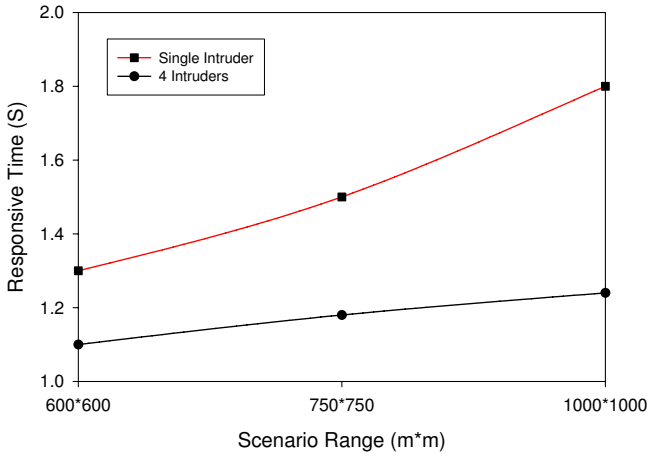

Figure 3.11: Responsive Time (S) at different scenarios. Where we have 4 intruders and each intruder has the same speed $20m/s$. The single intruder also runs at this speed.

3.6.5 Probability of Detection

Probability of Detection is the other QoS evaluation metric for energy-efficient QoS coverage scheme. Here we present an evaluation model that describes the probability of an intruder can be detected by the k sensor nodes when the intruder moves out of the scope of current k sensor nodes. We map the intruder detection problem to a line-set intersection problem. We use tools from Integral Geometry and Geometric Probability to analytically evaluate this probability of detecting the intruder moving at a random direction to the scope of the next nodes which are eligible to be activated but only k nodes are activated, where k is the desired coverage degree. A more detail presentation can refer to (Lazos et al., 2007).

In this model, we assume the trajectories of the mobile intruder are straight lines, with all the trajectories crossing the next scope being equiprobable. Although such an assumption constraints the space of all possible trajectories, our assumption gives a low bound of detection probability. Given any arbitrary entry and exit point in the next scope, moving on a straight line minimizes the length of the trajectory of the target within the next scope (minimizes the time that the intruder can be detected). Hence, the intruder detection probability assuming line trajectories is the worst case probability compared to the detection of any other possible trajectory. The worst case analysis allows us to compute network parameters such as sensor density and length of the perimeters of the sensing areas, so that intruder detection is guaranteed with a minimum probability.

The problem of mobile intruder detection under stochastic deployment can be mapped to a line-set intersection problem in the following way. Let the next scope be mapped to a bounded set S_0, defined as a collection of points in the plane with perimeter length L_0. Let the sensing area of sensor s_i be mapped to a bounded set s_i with perimeter length L_i. Let the trajectory of the intruder X be mapped to a straight line $A(\xi, \theta)$ in the plane, with parameters ξ and θ be the shortest distance of A to the origin of a coordinate system, and θ be the angle of the line perpendicular to A with respect to the x axis. Then, the mobile

intruder detection problem for stochastic sensor network is equivalent to the following line-set intersection problem.

Line-set intersection problem: Given a bounded set S_0 of perimeter length L_0 and N sets S_i of perimeter length L_i, randomly and independently placed inside S_0, compute the probability $P_D(k)$ that a random line A intersecting S_0, also intersects *at least* k out of the N sets $S_i, i = 1 \cdots N$, where k is the coverage degree.

Table 3.1 summarizes the mapping from the mobile intruder detection problem to the line-set intersection problem. Let A_0 be a bounded next scope of perimeter length L_0 monitored by N sensors randomly deployed within A_0, with sensor $s_i, i = 1 \cdots N$ having a sensing area of perimeter length L_i. The probability $P_D(k)$ that at least $k \geq 1$ sensors detect an intruder X moving on a random straight line trajectory is given by:

$$P_D(k) = 1 - \sum_{w=0}^{k-1} \sum_{j=1}^{|Z_{N,w}|} \prod_{i=1}^{|z_j|} q_{z_j(i)} \prod_{v=1}^{|\bar{z}_j|} (1 - q_{\bar{z}_n(v)}) \tag{3.5}$$

where $Z_{N,w}$ denotes the $\binom{N}{w}$ w-tuples z_j of vector $[1, \cdots, N]$. The $\bar{z}_j$ denotes the complement

Table 3.1: Mapping the mobile intruder detection problem to the line-set intersection problem

Mobile Intruder Detection	$\leftrightarrow$	Line-set Intersection
Number of sensors N	$\leftrightarrow$	Number of sets N
Next activated scope	$\leftrightarrow$	Set S_0
Sensing area A_i of perimeter L_i	$\leftrightarrow$	Set S_i of perimeter L_i
Random sensor deployment	$\leftrightarrow$	Random set placement
Trajectory of intruder X	$\leftrightarrow$	Random line l crossing S_0
Probability of intruder detection by K_s sensors $P_D(k)$	$\leftrightarrow$	Probability of l intersecting K_s sets

(N-w)-tuples of z_j with respect to vector $[1, \cdots, N]$, and q_i is given by $q_i = L_i/L_0$.

We now compute the probability $P(z_j)$ that a line A intersects *exactly* k sets denoted by the k-tuple z_j. Since the sets A_i are randomly and independently deployed, the probability of the intersection of events becomes equal to the product of the probability of the individual events.

$$\begin{aligned} P(z_j) &= Pr(\text{exactly k sets intersect the line}) \times Pr(\text{rest N-k sets not intersect the line}) \\ &= \prod_{i=1}^{|z_j|} q_{z_j(i)} \prod_{v=1}^{|\bar{z}_j|} (1 - q_{\bar{z}_j(v)}) \end{aligned} \tag{3.6}$$

where $\bar{z}_j$ denotes the complement of z_j. To compute the probability of a random line intersecting *any* k sets, $P(z_j)$ must be summed over all possible k-tuples z_j.

$$P(Z_{N,k}) = \sum_{Z_{N,k}} \prod_{i=1}^{|z_j|} q_{z_j(i)} \prod_{v=1}^{|\bar{z}_j|} (1 - q_{\bar{z}_j(v)}) \tag{3.7}$$

Therefore $P_D(k)$ can be expressed by:

$$P_D(k) = 1 - \sum_{w=0}^{k-1} P(Z_{N,w}) \tag{3.8}$$

In Fig. 3.12, we show $P_D(k)$ as a function of the coverage degree k changing from 1 to 6. The simulation scenario is $600m * 600m$. We observe that with the increasing of the coverage degree k, the probability is decreasing. We can select a k such that the $P_D(k)$ can be above a threshold, *i.e.*, if $P_D(k) \geq 95\%$, the coverage degree k can not be larger than 4. We also show the probability $P(z_j)$ of the intruder detected by exact k sensors as a function of the coverage degree k sensor nodes in Fig. 3.13. As we can see, to gain an optimal detection probability, the coverage degree is chosen as 8.

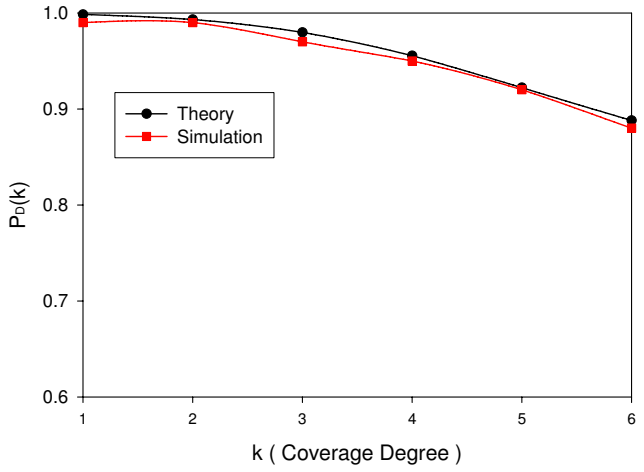

Figure 3.12: Probability of detection $P_D(k)$ for N = 100.

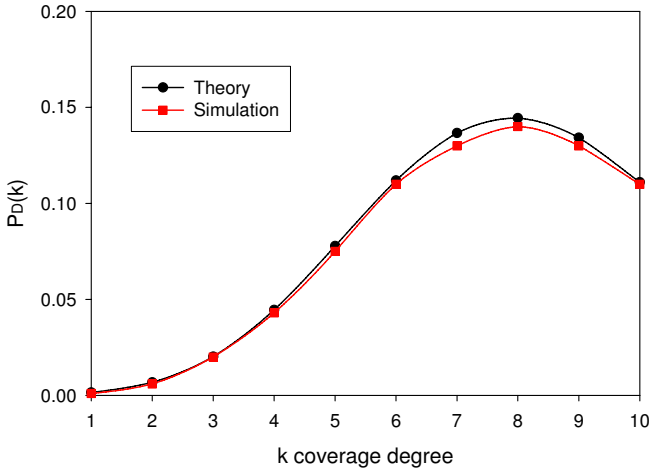

Figure 3.13: Probability of detection $P(Z_{N,k})$ for N = 100.

3.7 Summary

In this chapter, we designed on-demand intruders tracking algorithms using modified SPAN algorithm and CCP protocol. Initial sleeping nodes were activated to build backbones. As the intruders moved, nearby nodes were activated and sent intruders' positions information to the base station. When the intruders moved away, past nodes went to sleep again to save energy and future nodes were activated using the linear prediction model to estimate the intruders' future positions. Some QoS evaluation metrics in term of energy consumption, mean delay time until detection, responsive time and probability detection were introduced to measure our algorithms. Results showed that our algorithm is energy efficient compared with other on-demand algorithms such as VP and CCP. Also, in the mean delay time until detection and responsive time, our algorithm validated that sensor networks are more efficient to detect the enemy in term of lower delay time and quicker responsive time when sensor networks are used to detect huge amount of intruders. For the probability of detection, our algorithm can provide more than 90% intruders detection, which is important for realistic applications.

Chapter 4

Key Establishment for Layered Group-based Wireless Sensor Networks

4.1 Introduction

Information security is also a key QoS evaluation metric for the wireless sensor networks. It is because at some critical applications, such as battlefield, sensor nodes are captured and information channels can be eavesdropped easily. When these situation happen, wireless sensor networks must be robust enough to continue providing reliable service. This chapter presents a key establishment scheme for providing QoS to wireless sensor networks.

This chapter is organized as follows. Section 4.2 describes the architecture and properties of LGKE scheme in detail. The QoS analysis and quantitative metrics are given to evaluate LGKE scheme in Section 4.3. We summarize the chapter in Section 4.4.

4.2 The LGKE Scheme

As commonly adopted, we assume the base station is globally trusted and powerful in this study. We introduce the following notations.

Cluster head: A group head that acts as an information switch between the nodes in a group and the base station. To be consistent with the terms in the literature such as (Simplot-Ryl et al., 2005), we will use cluster head throughout this chapter. A non-cluster-head node communicates with the base station by sending messages to a cluster head first and the cluster head forwards them to the base station. Cluster heads are illustrated in Fig. 4.1. We will treat the number of cluster head as a smoothing parameter and find out the optimal value in the later section.

Agent: A sensor node which acts as an information switch between a node and the nodes in its neighboring group. A node communicates with the nodes in its neighboring group via an agent. As illustrated in Fig. 4.2, nodes n_x and n_y are the agents of nodes n_i and n_j respectively.

Neighboring groups: If two groups are neighbors in terms of geographic location, we define them as neighboring groups. As shown in Fig. 4.1, each group has 8 neighboring groups. The neighbors of G22 are G11, G12, G13, G21, G23, G31, G32, and G33.

Non-neighboring groups: If two groups are not adjacent in terms of geographic location, we call them non-neighboring groups. As shown in Fig. 4.3, groups G_u and G_w, G_p and G_q are non-neighboring groups.

Scalability Percentage: The ratio of the new sensor nodes to the original network size.

Preloaded key: The keys are stored in each node before deployment and are used within the same group.

Path key: The keys are randomly generated and are used to communicate with the nodes in neighboring groups.

Inter non-neighboring group key: The keys are used to communicate with the nodes in

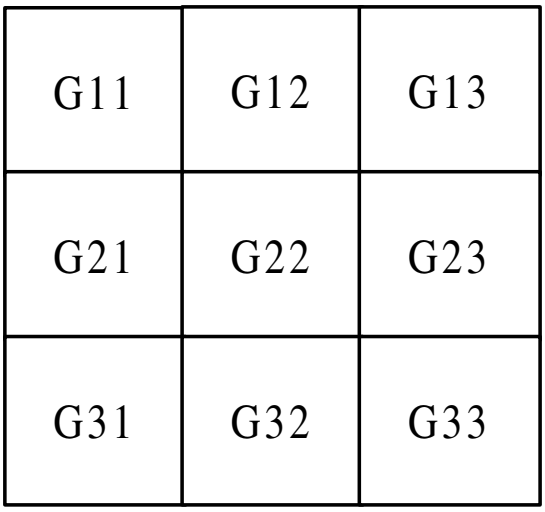

Figure 4.1: An illustration of neighboring groups. The neighboring groups of G22 are its geographic neighbors: G11, G12, G13, G21, G23, G31, G32, and G33. While in Fig. 4.3, groups G_u and G_w, G_p and G_q are non-neighboring groups.

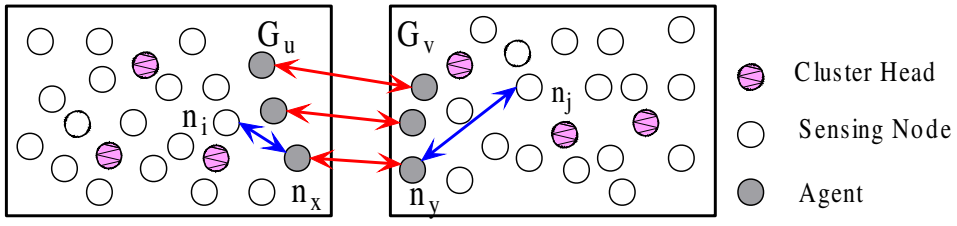

Figure 4.2: The path key establishment between neighboring groups through 3 agents. n_i in G_u and n_j in G_v will randomly choose agents n_x and n_y as the intermediate nodes; then the communication path key $K_{i,j}$ between n_i and n_j can be established through n_x and n_y.

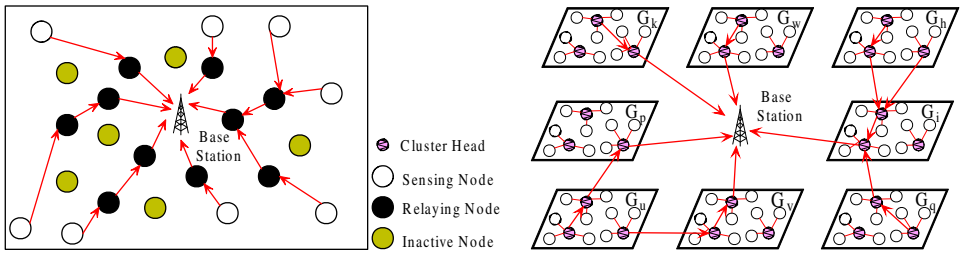

Figure 4.3: The architecture of the LGKE scheme. The left figure is a traditional flat sensor network without layers. The right one is the architecture of a two-layer sensor network. In the two-layer network, a logical top layer is composed of the base station and the cluster heads while a logical lower layer consists of other sensor nodes which are divided into groups.

non-neighboring groups.

4.2.1 The Scheme Architecture

We define the WSN topology for the LGKE scheme as shown in Fig. 4.3, which differs from traditional flat networks. In a flat network, all nodes are identical and their architecture depends merely on the geographic deployment. Although simple for small networks, this network architecture suffers from scalability. For example, adding a new node needs to inform the whole network, which involves in large communication energy consumption and is infeasible for the WSN because of its intrinsic resource limitation.

To overcome this scalability problem, the LGKE scheme adopts a two-layer architecture: a logic top layer that is composed of the base station and the cluster heads and a logic lower layer that consists of the non-cluster-head nodes. This architecture coincides with the scale free network model (Barabási and Albert, 1999) that there is a high probability that a node links to a vertex that has already a large number of connections. It means that for each communication unit, or group, there will be one or two nodes which act as switches aggregating and forwarding information with other groups.

Also, in practice, the sensor nodes in a WSN are usually deployed in groups with hierarchial architectures (Du et al., 2004a; Karlof and Wagner, 2003). Therefore, the two-layer

group-based LGKE architecture is reasonable and practical for the large scale network. As shown in Fig. 4.3, each group contains three nodes acting as cluster heads communicating with the base station and the other cluster heads. If two groups are neighbors, the secured inter-group communication is via path keys while the secured intra-group communication is via pair-wise key establishment since unique pair-wise keys have perfect resilience against node captures. If two groups are not neighbors, the sensors will communicate via the cluster heads and the base station. The details of the layered topology are described as follows.

4.2.2 The Top Layer

In the top layer, we adopt and extend the EBS technique to ensure the secure communication between the base station and the cluster heads (Eltoweissy et al., 2004). The EBS group keys management method is based on a combinatorial formulation and is defined as a collection Γ of subsets of the set of cluster heads. The elements of each subset $A \in \Gamma$ are the cluster heads that have a key. For simplicity, we denote A to both the key and the corresponding subset. We also assume that, in addition to the administrative keys corresponding to the subsets in Γ, the base station has a session key known to all cluster heads, and a personal key known only to each cluster head (and the base station). The session key is clearly needed for multi-casting encrypted data messages to all cluster heads. Personal keys are used for cluster head authentication and for uni-casting initialization information when an individual cluster head joins a multi-cast group.

We use EBS(n, k, m) to denote an EBS of dimension (n, k, m), where n is the number of the cluster heads of the top layer system, k is the number of keys stored by each cluster head, and m is the number of keys for the global key set not stored in one subset. n, k and m are positive integers, and $1 < k$, $m < n$. We denote A_i as the key known by each of the cluster heads and the corresponding subset consisting of the cluster heads knowing the key A_i. Γ is the collection of subsets A_i. The i^{th} cluster head only knows k keys, and there are

m keys known by other cluster heads except the i^{th} cluster head. An example is given to demonstrate how to build the EBS and to assign the keys as follows.

- Construct an EBS system

We illustrate the construction of an EBS system as follows. For a total 10 cluster heads system, the optimal value of k is $\lfloor \log_2 10 \rfloor$ (Eltoweissy et al., 2004), which is 3; m is selected as 2 to ensure $\binom{k+m}{k} \geq n$ or $\binom{5}{3} \geq 10$. As shown in Fig. 4.4, in each column, we have 3 (corresponding to k) 1s and 2 (corresponding to m) 0s. We can see that subset A_1 contains cluster heads $\{5, 6, 7, 8, 9, 10\}$, which means in row 1 the items at columns $\{5, 6, 7, 8, 9, 10\}$ are 1s, and the cluster heads $\{5, 6, 7, 8, 9, 10\}$ know the key A_1. Consequently, A_2 contains cluster heads $\{2, 3, 4, 8, 9, 10\}$, A_3 contains cluster heads $\{1, 3, 4, 6, 7, 10\}$, A_4 contains cluster heads $\{1, 2, 4, 5, 7, 9\}$, A_5 contains cluster heads $\{1, 2, 3, 5, 6, 8\}$. And the collection set $\Gamma = A_1, A_2, A_3, A_4, A_5$. Therefore, we have:

$$\begin{aligned}
\{1 \cdots 10\} - \{1\} &= A_1 \cup A_2 \\
\{1 \cdots 10\} - \{2\} &= A_1 \cup A_3 \\
\{1 \cdots 10\} - \{3\} &= A_1 \cup A_4 \\
\{1 \cdots 10\} - \{4\} &= A_1 \cup A_5 \\
\{1 \cdots 10\} - \{5\} &= A_2 \cup A_3 \\
\{1 \cdots 10\} - \{6\} &= A_2 \cup A_4 \\
\{1 \cdots 10\} - \{7\} &= A_2 \cup A_5 \\
\{1 \cdots 10\} - \{8\} &= A_3 \cup A_4 \\
\{1 \cdots 10\} - \{9\} &= A_3 \cup A_5 \\
\{1 \cdots 10\} - \{10\} &= A_4 \cup A_5
\end{aligned} \tag{4.1}$$

where $\{1 \cdots 10\}$ means this set contains all the nodes from 1 to 10.

row_1	0	0	0	0	1	1	1	1	1	1
row_2	0	1	1	1	0	0	0	1	1	1
row_3	1	0	1	1	0	1	1	0	0	1
row_4	1	1	0	1	1	0	1	0	1	0
row_5	1	1	1	0	1	1	0	1	0	0
Col	1	2	3	4	5	6	7	8	9	10

Figure 4.4: An EBS construction matrix for $k = 3$ and $m = 2$. There are three ones and two zeros in each column.

We can easily verify that cluster head 1 knows the keys A_3, A_4, A_5 ($k = 3$ keys), and does not know keys A_1, A_2 ($m = 2$ keys). Based on the property of EBS, we can use this mechanism to rebuild the secure communication when one cluster head, e.g. 1 is captured. Cluster head 1 only knows keys A_3, A_4, and A_5. So, these keys need to be changed and two messages containing new values can be encrypted by keys A_1 and A_2 respectively and be sent out. Each message contains four subparts: 1) a new session key, 2) replacement key A'_3 encrypted by the former A_3 key, 3) replacement key A'_4 encrypted by the former A_4 key, and 4) replacement key A'_5 encrypted by the former A_5 key. It can be easily verified that the two messages can ensure the remaining cluster heads to communicate securely, and cluster head 1 cannot decipher the messages since it does not know A_2.

When an arbitrary cluster head departures or is captured, the EBS can be constructed in a similar way through EBS(n, k, m). The base station can send m messages encrypted by A_1, A_2,$\cdots$, A_m respectively with $\bigcup_{i=1}^{m} A_i = \{1, \cdots, n\} - \{t\}$, and the i^{th} message contains the new session key and new personal keys encrypted by their former ones to ensure that the information can be deciphered by the corresponding cluster heads.

- Deployment Methods

There are many methods and models to deploy sensor networks (Liu et al., 2005; Zhou

et al., 2005). For example, an airplane can scatter the nodes over the battlefield. Once the sensor nodes are deployed, we assume they are static. And after being deployed, each node will send a message to the base station containing its position; the base station will calculate the distance between the node and itself. As shown in Fig. 4.3, the base station determines the cluster heads in each group and communicates with them via the EBS scheme.

4.2.3 The Low Layer

- Key Establishment

Preloaded key establishment: A unique pairwise key is preloaded for every intra-group sensor pair. A standard group size is selected to be $\gamma = 100$ as the same number is taken in (Du et al., 2003; Zhou et al., 2005; Chan and Perrig, 2005; Liu and Ning, 2003a), such that each sensor stores 99 keys. If the key size is 64 bits, each sensor requires 792 bytes. For sensor nodes as Mica2 Mote sensors designed by Berkeley that have 4KB SRAM (Pottie and Kaiser, 2000), the memory usage is less than 20%. Also, we can halve the memory requirement using the method in (Chan and Perrig, 2005) such that less than 10% memory usage for keys storage is enough to ensure that any pair of sensors within the intra-group share a unique preloaded key.

Path key establishment: As shown in Fig. 4.2, the number of agent t is arbitrarily selected as 3, which means group G_u has 3 nodes who can directly communicate with group G_v via keys encryption. And other nodes in group G_u have to communicate with the nodes in group G_v via the three agents in G_u. The number of agents per group is related to the group size and preloaded keys and its calculation will be discussed in the next section.

More generally, each sensor node holds m path keys, each group can has $t = \lceil \frac{m\gamma}{8} \rceil$ agents in its neighboring group. For example, if node n_i in G_u wants to establish the key agreement with n_j in G_v, it will follow the following steps:

1. n_i, n_j will randomly choose agents n_x and n_y as the communication intermediary;

2. n_i randomly generates a key $K_{i,j}$ and sends to n_x encrypted with the pairwise key $K_{i,x}$ shared with n_x;

3. n_x decrypts the received packet containing $K_{i,j}$ by using the pairwise key $K_{i,x}$, and re-encrypts the packet with the key $K_{x,y}$ shared with n_y, and sends it to n_y;

4. n_y decrypts the packet and re-encrypts it with the key $K_{y,j}$, and sends it to n_j;

5. n_j deciphers the packet using $K_{y,j}$, and stores the path key $K_{i,j}$ corresponding to n_y.

Inter-non-neighboring-group key establishment: If two groups are not neighbors as shown in the example in Fig. 4.5. For example, G_u and G_w are non-neighboring groups; G_p and G_q are non-neighboring groups; and G_u and G_v, G_u and G_p are neighboring groups. If a sensor node n_i in G_u wants to communicate with n_t in G_w, a possible solution is to build a path via the agents in groups G_u, G_p, G_k and finally reach to G_w. However, this method involves in too many communication hops between non-cluster head sensor nodes and therefore consumes too much energy. It is unwise in a energy limited sensor network. Here we present an alternative scheme in which the sensor node n_i can reach to n_t via the cluster heads and base station, as shown in Fig. 4.5. This multi-hop between cluster heads and the base station ensures their communication will consume much less energy than multi-hop communication between non-cluster heads sensor nodes.

Also, since the communication between the sensor nodes and the cluster heads, say, n_i and c_x, n_t and c_y, is within the same group, it involves in the local communication. Hence, this scheme decreases the communication energy wasted in multi-hop communications. Moreover, the communication between the cluster heads and the base station is secure because the message is encrypted with the pairwise key through EBS.

The key agreement setup is similar with the steps in neighboring-group key agreement setup and is shown in Fig. 4.5.

1. n_i and n_t will randomly choose cluster heads c_x and c_y as the communication intermediary;

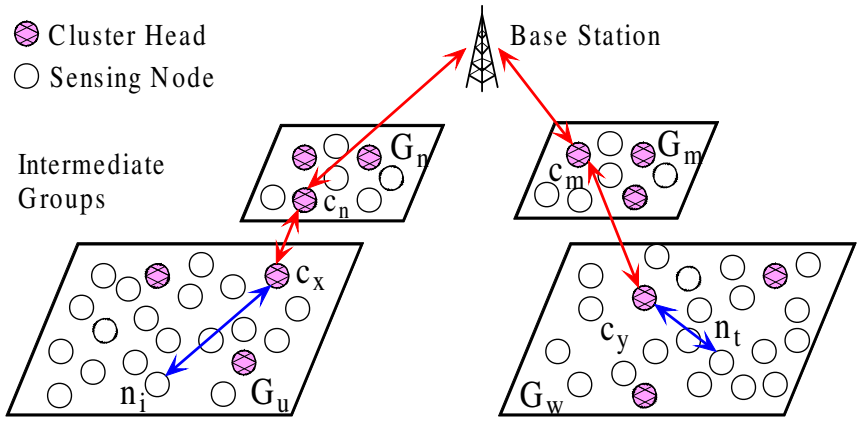

Figure 4.5: Inter-non-neighboring-group key establishment between non-neighboring groups. n_i in G_u and n_t in G_w will randomly choose cluster heads c_x and c_y as the intermediate nodes; then the communication key $K_{i,t}$ between n_i and n_j can be established through c_x, c_n, the base station, c_m and c_y.

2. n_i randomly generates a key $K_{i,t}$ and sends to c_x encrypted with the pairwise key $K_{i,x}$ shared with c_x;

3. c_x decrypts the received packet containing $K_{i,t}$ by using the pairwise key $K_{i,x}$, and re-encrypts the packet with the key $K_{x,n}$ shared with some cluster head c_n in some intermediate group G_n which is nearer to the base station than Group G_u. We can choose more than one group as the intermediate groups according to the locations and groups and base station.

4. c_n decrypts the received packet containing $K_{i,t}$ by using the personal key $K_{x,n}$, and re-encrypts the packet with the key $K_{n,b}$ shared with the base station, and sends it to the base station;

5. The base station decrypts the packet and re-encrypts it with the key $K_{b,m}$, and sends it to the cluster head c_m in the intermediate group G_m;

6. The cluster head c_m decrypts the packet and re-encrypts it with the personal key $K_{m,y}$, and sends it to c_y;

7. The cluster head c_y decrypts the packet and re-encrypts it with the key $K_{y,t}$, and sends it to n_t;

8. n_t deciphers the packet using the pairwise key $K_{y,j}$, and stores the path key $K_{i,t}$ corresponding to the cluster head c_y.

For a large scale senor network, it is not unusual to communicate between two non-neighboring groups. For example, if a query requested in one group may need to know the sensing information of its non-neighboring groups, a communication path will be built to meet this query. Traditional optimal path search methods like Bellman-Ford algorithm, involve many communications within the whole network and hence consume much energy, which increases the possibility of being impersonated. While the communication via the base station decreases this possibility since the base station is safe and resource rich as assumed, and the selection of the optimal path to the base station is confined in a subnet containing the base station only.

4.2.4 Scalability Analysis

The LGKE architecture is more scalable than the flat distributed architecture and the EBS key establishment mechanism ensures an easy way to add/delete a group.

- Add a Group

To add a group to a WSN, we need to add 3 cluster heads since each group contains 3 cluster heads. We propose two algorithms to add a new cluster head to the EBS system. The first algorithm is to let the base station randomly generates a new key, and lets all the existing cluster heads (including the new one) know the new key. We can add a new bottom row to the matrix, as shown in the left figure of Fig. 4.6, and place one 1 in each of the existing columns of that row. We need to add a new column and place $k+1$ 1's, but with one

Figure 4.6: An illustration of adding a cluster head. The main difference is that the first algorithm (the left) appends an all-1 row to the matrix while the second one (the right) appends an all-0 row. The new added column must be different from the existed columns but have same numbers of ones and zeros.

0 in the last row. Placing one 1 in each of the existing columns means letting existing users know the new key. This method corresponds to an $EBS(n+1, k+1, m)$ system, and $k+1$ keys are known to each cluster head, and m packets are needed for re-keying operations. The other two cluster heads can be added similarly.

The second algorithm is to add a new bottom row to the matrix and to place one 0 in each of the existing columns. Then add a new column that it also has k 1's, but with one 1 in the last row, as shown in the right figure of Fig. 4.6. Placing a 0 in each of these columns corresponds to not giving the new key to any of the existing users. Placing one 0 in each of these columns corresponds to no need for each existing cluster head to store the new generated key. That is, the method is to extend the existing EBS system to $EBS(n+1, k, m+1)$. k keys are known to each cluster head and $m+1$ packets are needed for re-keying operations.

There is a tradeoff in the memory usage and the communication overhead between the above two algorithms on the top layer when adding or deleting groups. The first algorithm (add ones in the bottom row) requires more memory space for the cluster heads, since when a new cluster head joins in, a new key is needed to store in all the cluster heads, but it does not increase the re-keying operations. Therefore, compared to the second algorithm, the fist one needs more memory space but less communication overhead. An appropriate algorithm to balance the memory usage and the communication overhead should be considered for the

base station. Here we choose the algorithm according to the scalability percentage. For example, for a network with $10,000$ sensor nodes and being divided into 100 groups with 100 nodes per group, we need large scalability percentage 100%, which means to add 100 more groups. The first algorithm needs another 300 keys while the second one needs another 300 re-keying operations. Considering the energy consumption in re-keying operations, we prefer the first method to extend the network.

- Delete a Group

When a group is evicted, the base station can re-key and notify the system by m multicast packets as previously indicated. Since one group contains 3 cluster heads, we assume the 3 cluster heads are evicted one by one when a group is evicted. It is doable because the eviction of sensor nodes in a group should be slower than the communication between the base station and the sensor nodes, and there must be some interval between the evictions of two cluster heads. However, by deleting a group, an EBS(n, k, m) may reduce the number of keys stored in each node or the number of packets needed for re-keying.

In fact, after some evictions and additions, the evolved EBS system could be far from the optimal and the base station may re-allocate the keys for the top layer, if necessary. For example, for the above network, in the top layer of an EBS system, the total number of cluster heads N_e is $3 \times 100 = 300$, k can be chosen as $\lceil \log_2 300 \rceil = 9$, and m is chosen to ensure $\binom{k+m}{m} > N_e$, say, $m = 4$, such that the top layer can be denoted as $EBS(300, 9, 4)$. When adding 100 groups and deleting 50 groups, the EBS system contains 450 cluster heads and 309 keys and needs 4 re-keying operations, which is far beyond the optimum. The base station can re-allocate the keys distribution to $EBS(450, 10, 4)$ to spare more memory for the cluster heads.

4.2.5 Properties of LGKE

The LGKE scheme has the following properties in terms of QoS evaluation:

Resilience against impersonation: Since all the packets between the sender and the receiver are encrypted and deciphered by the shared pairwise key, these packets can not be deciphered or impersonated by the attackers, such as Sybil attack(Karlof and Wagner, 2003).

Less communication overhead: All the communications are within the local distance. Therefore, the energy consumption due to the communication overhead is less than the consumptions in other schemes, such as GKE and PIKE, which are all involved in network-wide communications and thus consume more energy.

Resistance against group capture: LGKE can resist the group capture. Once a group is captured, the base station will exclude the cluster heads from the top layer system through EBS.

Dynamic scalability: If x groups want to join in the sensor network, the base station runs the ADD algorithm $x \times 3$ times, since each group contains 3 cluster heads. This property is a distinct advantage over other schemes. For instance, for a $10,000$ sensor nodes network, in a flat EBS scheme $EBS(10000, 14, 6)$ in (Eltoweissy et al., 2004), adding 100 nodes makes not much difference, and the scalability percentage is only $\frac{100}{10000} = 1\%$. But in LGKE, since the network is divided into groups, the N_e is $3 \times 100 = 300$, k can be chosen as $\lceil \log_2 300 \rceil = 9$, and m is chosen to ensure $\binom{k+m}{m} > N_e$, e.g. $m = 4$, the system is hence denoted as $EBS(300, 9, 4)$, and the scalability percentage is $\frac{100*3}{300} = 100\%$. While the overhead of running the ADD algorithm can be ignored since algorithm is executed in the base station, which is powerful and energy-unlimited. While other schemes, such as *Random pairwise key scheme* (Chan et al., 2003), *location-based random pairwise key schemes* Du et al. (2003); Liu and Ning (2003a), GKE (Zhou et al., 2005, 2006)and PIKE (Chan and Perrig, 2005), assume the network is static after being deployed which either ignore the scalability issue or have low scalability percentage.

Denial of Service(DoS) Resistance: The intermediate nodes may come across the DoS attack during the establishment of the path key in the low layers. Since the intermediate

agents are randomly chosen, and the messages routed from n_i to n_j may have more than one braided path, which provides probabilistic protection against DoS attack.

Resistance against cluster heads capture: The EBS scheme is able to reconstruct the top layer when the cluster heads are compromised. Even when more than one cluster head are captured, the EBS scheme can reconstruct the top layer.

4.3 QoS Evaluation

We evaluate the QoS performance of the LGKE scheme based on two metrics: the survivability and the resource consumption. The measure of survivability is defined to determine the resilience, resistance and robustness against the attackers, and the resource consumption focuses on the communication overhead and memory usage.

4.3.1 Survivability

Various survivability definitions have been proposed in different disciplines, such as the definition from Ellison et al. (1997), which emphasizes the time-varying behavior of the system after a failure or an attack. Knight Knight et al. (2000) introduces a general definition of survivability for critical information systems: *A survivability specification is a four-tuple, (E; R; P; M) where: E is a statement of the assumed operating environment for the system, R is a set of specifications each of which is a complete statement of a tolerable form of service that the system must provide, P is a probability mass function across the set of specifications, and M is a finite state machine.*

For sensor networks which have many characteristics that make them more vulnerable to attacks than conventional computing equipments, we define two QoS criteria to represent desirable characteristics of the LGKE based on the definition of Knight et al. (2000); Li and Yang (2006): resilience and robustness. Resilience is defined as one of the following:

(i) the probability that at least a link is compromised when an adversary captures a node, (ii) number of nodes whose security credential is compromised when an adversary captures a node, or (iii) number of sensor nodes required to be captured to compromise the whole network. Robustness considers the probability that two (or more) sensor nodes store the same key or keying material that can be used to establish pair-wise keys.

4.3.2 Quantitative QoS Metrics

In order to compare the QoS performance of the LGKE scheme with other schemes, we use the following system settings in the quantitative analysis. The size of the sensor network N ranges from 10,000 to 50,000, with 10,000 being the default value. The group size γ is set to be 100 as other group-based schemes Chan et al. (2003); Du et al. (2003); Zhou et al. (2005). Consequently, the number of groups g is equal to N/γ, varying from 100 to 500.

We evaluate the LGKE scheme in terms of its resilience and robustness against node capture and attacks based on the definition above and formulate into two metrics: (1) When x nodes are captured, what is the probability that at least one secure connection is compromised? This QoS analysis shows the network's resilience with x nodes are captured. (2) When x nodes are captured, what is the probability of the actual two nodes share one pairwise key? This QoS analysis shows the networks's robustness against the x nodes' being captured. In our analysis, we assume that the attackers have no priori knowledge of the keys carried by each sensor and therefore we model the attacker as compromising random nodes.

- Resilience

Let n_i and n_j be two un-compromised nodes. Let $L_{i,j}$ be the connection and $K_{i,j}$ be the shared key between them. Let $\Upsilon(K_{i,j})$ be the event that $K_{i,j}$ is a preloaded key, let $\Pi(K_{i,j})$ be the event that $K_{i,j}$ is a path key, and let $\Phi(K_{i,j})$ be the event that $K_{i,j}$ is an inter-non-neighboring-group key. Let $\bar{L_{i,j}}$ be the event that connection $L_{i,j}$ is compromised, and $C(x)$

be the event that x sensors have been compromised. According to the Bayes theorem, the probability that $\bar{L_{i,j}}$ has occurred given that x sensors have been compromised equals to the probability of n_i and n_j are within the same group, plus the probability of n_i and n_j are within neighboring groups, and plus the probability of n_i and n_j are within non-neighboring groups, or alternatively,

$$\begin{aligned} Pr[\bar{L_{i,j}}|C(x)] = & Pr[\bar{L_{i,j}}|C(x) \wedge \Upsilon(K_{i,j})] \times Pr[\Upsilon(K_{i,j})] \\ & + Pr[\bar{L_{i,j}}|C(x) \wedge \Pi(K_{i,j})] \times Pr[\Pi(K_{i,j})] \\ & + Pr[\bar{L_{i,j}}|C(x) \wedge \Phi(K_{i,j})] \times Pr[\Phi(K_{i,j})] \end{aligned} \tag{4.2}$$

Since in LGKE preloaded pairwise keys are unique, the communication secured by a preloaded key can not be compromised unless one of its endpoints is compromised(Zhou et al., 2005, 2006; Chan and Perrig, 2005). Therefore, LGKE achieves perfect resilience against node captures by $Pr[\bar{L_{i,j}}|C(x) \wedge \Upsilon(K_{i,j})] \times Pr[\Upsilon(K_{i,j})] = 0$.

Since each group has 8 neighboring groups to communicate using path key and each group has $t = \lceil (\frac{m\gamma}{8}) \rceil$ agents in every other neighboring group, where m is number of inter-group pairwise keys each node stores, and γ is the group size, 100. Let α be the probability that either n_i or n_j is the agent of its neighbor's group, then $\alpha = \frac{\binom{t}{1}}{\binom{\gamma}{1}} = \frac{t}{\gamma}$. $\Pi_1(K_{i,j})$ is the event of either n_i or n_j being the agent of the other group but not both; and $\Pi_2(K_{i,j})$ is the event of neither n_i nor n_j being agents. Therefore, we have:

$$\begin{aligned} Pr[\Pi_1(K_{i,j})] &= 2\alpha(1-\alpha) \\ Pr[\Pi_2(K_{i,j})] &= (1-\alpha)^2 \end{aligned} \tag{4.3}$$

$p_1 = \binom{N-3}{x} / \binom{N-2}{x}$ is the probability of the agent used to send the path key $K_{i,j}$ which is not compromised when n_i and n_j are not compromised but x nodes being compromised. Then $Pr[\bar{L_{i,j}}|C(x) \wedge \Pi_1(K_{i,j})]$ can be obtained as $1 - p_1$. Similarly, we can get $Pr[\bar{L_{i,j}}|C(x) \wedge$

$\Pi_2(K_{i,j})]$, and the two formulations are as follows:

$$\begin{aligned} Pr[\bar{L_{i,j}}|C(x) \wedge \Pi_1(K_{i,j})] &= \frac{x}{N-2} \\ Pr[\bar{L_{i,j}}|C(x) \wedge \Pi_2(K_{i,j})] &= 1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}} \end{aligned} \tag{4.4}$$

So, the resilience for the inter-neighboring-group communication can be obtained as:

$$\begin{aligned} Pr[\bar{L_{i,j}}|C(x) \wedge \Pi(K_{i,j})] &= Pr[\bar{L_{i,j}}|C(x) \wedge \Pi_1(K_{i,j})] \times Pi_1(K_{i,j}) \\ &\quad + Pr[\bar{L_{i,j}}|C(x) \wedge \Pi_2(K_{i,j})] \times Pi_2(K_{i,j}) \\ &= 2\alpha(1-\alpha)\frac{x}{N-2} + (1-\alpha)^2(1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}}) \end{aligned} \tag{4.5}$$

where $Pr[\Pi(K_{i,j})]$ is the ratio of the number of path keys to the total number of keys among all pairs of neighboring sensors.

For the inter-non-neighboring-group communication, $\Phi_1(K_{i,j})$ is the event of either n_i or n_j being the cluster head but not both; and $\Phi_2(K_{i,j})$ is the event of neither n_i nor n_j being cluster heads. Therefore, we have:

$$\begin{aligned} Pr[\Phi_1(K_{i,j})] &= 2\binom{\gamma-3}{1}\binom{3}{1}/\binom{2\gamma}{2} \\ Pr[\Phi_2(K_{i,j})] &= \binom{\gamma-3}{1}^2/\binom{2\gamma}{2} \end{aligned} \tag{4.6}$$

$p_2 = \binom{N-3}{x}/\binom{N-2}{x}$ is the probability of the cluster head used to send the path key $K_{i,j}$ which is not compromised when n_i and n_j are not compromised but x nodes being compromised. Then $Pr[\bar{L_{i,j}}|C(x) \wedge \Phi_1(K_{i,j})]$ can be obtained as $1 - p_2$. Similarly, we can get $Pr[\bar{L_{i,j}}|C(x) \wedge$

$\Phi_2(K_{i,j})]$, and the two formulations are as follows:

$$Pr[\bar{L_{i,j}}|C(x) \wedge \Phi_1(K_{i,j})] = 1 - \binom{N-3}{x} / \binom{N-2}{x}$$
$$Pr[\bar{L_{i,j}}|C(x) \wedge \Phi_2(K_{i,j})] = 1 - \binom{N-4}{x} / \binom{N-2}{x} \quad (4.7)$$

where $Pr[\Phi(K_{i,j})]$ is the ratio of the number of inter-non-neighboring-group keys to the total number of keys among all pairs of non-neighboring sensors. Consequently, the final expression of resilience is:

$$\begin{aligned} Pr[\bar{L_{i,j}}|C(x)] = ((1-\alpha)^2(1 - \frac{\binom{N-3}{x}}{\binom{N-2}{x}}) \\ + 2\alpha(1-\alpha)(1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}})) \times Pr[\Pi(K_{i,j})] \\ + (\binom{\gamma-3}{1}^2 / \binom{2\gamma}{2}(1 - \frac{\binom{N-3}{x}}{\binom{N-2}{x}}) \\ + 2\binom{\gamma-3}{1}\binom{3}{1} / \binom{2\gamma}{2}(1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}})) \times Pr[\Phi(K_{i,j})] \end{aligned} \quad (4.8)$$

Fig. 4.7 shows the numeric resilience analysis, compared with the results of GKE and PIKE. When 100 to 900 of 10,000 nodes are compromised, the fraction of communication is compromised. It can be noticed that in Du's scheme (Du et al., 2003), when around 350 of 10,000 sensors are compromised, the resilience decreases dramatically. In contrast, the algorithms (LGKE, GKE and PIKE) show graceful degradation of resilience (below 6%), therefore attackers are unable to compromise a large fraction of other communication links by compromising a small number of sensors. It also can be seen that LGKE scheme is better than GKE and PIKE with regard to resilience because LGKE scheme considers not only the preloaded keys and path keys as PIKE and GKE, but also the inter-non-neighboring-group keys.

- Robustness

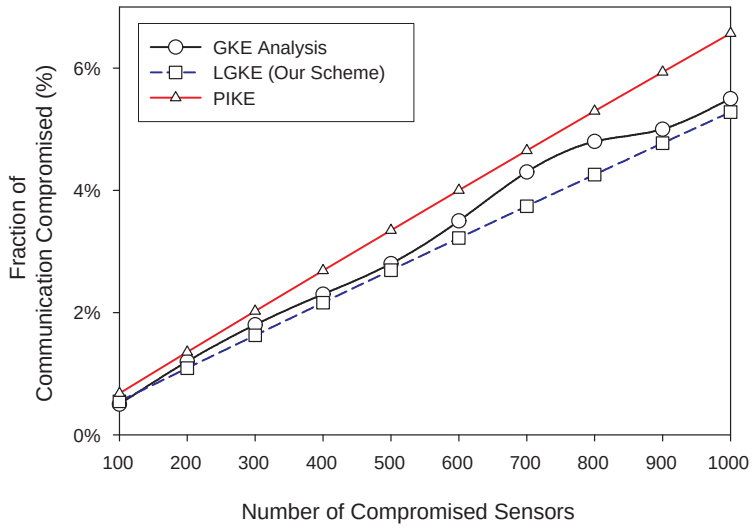

Figure 4.7: Comparison of the resilience of GKE, PIKE, and LGKE. LGKE has the best QoS performance with the number of captured sensors increasing from 100 to 1000. The network has 10,000 sensors, and each group contains 100 sensor nodes.

The quantitative formulation of robustness is the probability that at least one pair of nodes stay connected when x nodes are captured, which shows the robustness of the sensor network to provide information service when attacks occur. Let $L_{i,j}$ be the connection and $K_{i,j}$ be the shared key between arbitrarily selected n_i and n_j. Let S be the events set that the connection $L_{i,j}$ of the nodes n_i and n_j is compromised given that x nodes are captured. Specifically, let S_Υ be the event that $L_{i,j}$ is compromised when $K_{i,j}$ is a preloaded key, let S_Π be the event that $L_{i,j}$ is compromised when $K_{i,j}$ is a path key, and let S_Φ be the event that $L_{i,j}$ is compromised when $K_{i,j}$ is a inter-non-neighboring-group key. Let $C(x)$ be the event that x nodes have been compromised in the network. We have the probability of at least one connection exists as follows:

$$\begin{aligned} & Pr[\text{at least one connection exists}|C(x)] \\ & = 1 - Pr[\text{all connections are compromised}|C(x)] \\ & = 1 - Pr[S_\Upsilon \cap S_\Pi \cap S_\Phi|C(x)] \end{aligned} \tag{4.9}$$

Since events S_Υ, S_Π, S_Φ are mutually exclusive, therefore

$$Pr[S_\Upsilon \cap S_\Pi \cap S_\Phi | C(x)] = Pr[S_\Upsilon | C(x)] \times Pr[S_\Pi | C(x)] \times Pr[S_\Phi | C(x)] \tag{4.10}$$

So the expression of robustness can be derived as

$$\begin{aligned} &Pr[\text{at least one connection exists} | C(x)] \\ &\qquad = 1 - Pr[S_\Upsilon | C(x)] \times Pr[S_\Pi | C(x)] \times Pr[S_\Phi | C(x)] \end{aligned} \tag{4.11}$$

As mentioned above, since preloaded pairwise keys are unique, the communication secured by a preloaded key can not be compromised unless one of its endpoints is compromised. Therefore, we can obtain

$$Pr[S_\Upsilon | C(x)] = \frac{2\binom{N-1}{x-1} + \binom{N-2}{x-2}}{\binom{N}{x}} \tag{4.12}$$

$Pr[S_\Pi | C(x)]$ can be calculated using Eqs. 4.3 and 4.4, and $Pr[S_\Phi | C(x)]$ can be calculated using Eqs. 4.6 and 4.7. We have

$$\begin{aligned} Pr[S_\Pi | C(x)] &= 2\alpha(1-\alpha)(1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}})) \times Pr[\Pi(K_{i,j})] \\ &\quad + ((1-\alpha)^2(1 - \frac{\binom{N-3}{x}}{\binom{N-2}{x}}) \\ Pr[S_\Phi | C(x)] &= 2\binom{\gamma-3}{1}\binom{3}{1}/\binom{2\gamma}{2}(1 - \frac{\binom{N-4}{x}}{\binom{N-2}{x}})) \times Pr[\Phi(K_{i,j})] \\ &\quad + (\binom{\gamma-3}{1}^2/\binom{2\gamma}{2}(1 - \frac{\binom{N-3}{x}}{\binom{N-2}{x}}) \end{aligned} \tag{4.13}$$

The robustness approaches to 1, which means LGKE can always assure a secured communication path with $x \leq N-2$ nodes being captured.

4.3.3 Memory Usage

The LGKE scheme needs low memory requirements. For a sensor network of N sensors, with group size γ, LGKE requires each sensor to be preloaded with $\gamma - 1$ pairwise keys shared with sensors within the same group and $t = \lceil \frac{m\gamma}{8} \rceil$ path keys shared with sensors that are in the different groups, including $k = \lceil \log_2 3g \rceil$ inter-non-neighboring-group keys especially for the 3 cluster heads. t is the same as the number of agents per group. We also use the method in (Chan and Perrig, 2005) to halve the memory requirement. Therefore, the total memory overhead per sensor is $(\gamma - 1)/2 + \lceil \frac{m\gamma}{8} \rceil$ keys for non-cluster-head nodes, and $(\gamma - 1)/2 + \lceil \frac{m\gamma}{16} \rceil + \lceil \frac{1}{2} \log_2 3g \rceil$ for the cluster heads.

4.3.4 Communication Overhead

The communication overhead is defined as the average number of hops that a message has to be transmitted in order to establish a key between any pair of neighboring sensors (Zhou et al., 2005, 2006; Chan and Perrig, 2005).

Under the LGKE scheme, the selection of cluster heads is based on the geographical characteristics. The top layer communication is between base station and the cluster heads. The communication between neighboring groups is based on agents and the communication between non-neighboring groups is based on cluster heads and base station.

- Path Key Communication Overhead

As mentioned in Section 4.2.3, n_i in G_u wants to establish the key agreement with n_j in G_v, it will involve in messages from n_i to n_x, through n_y to n_j. Let $h(n_p, n_q)$ be the hop distance between n_p and n_q. The total hops from n_i to n_j can be expressed as: $H(n_i, n_j) = h(n_i, n_x) + h(n_x, n_y) + h(n_y, n_j)$. If H_{LGKE} is the expected number of hops for path key establishment in LGKE, the linearity of expectation leads to $H_{LGKE} = 2\bar{h}_{LGKE} + \bar{h'}_{LGKE}$,

where $\bar{h}_{LGKE}$ is the expected hop distance between any two nodes within a group, and $\bar{h'}_{LGKE}$ is the expected hop distance between any two nodes from neighboring groups.

If two nodes are separated by physical distance $\bar{\lambda}$, and the transmission radius is r for a node and infinity for the base, we will need at least $\bar{\lambda}/r$ hops for sensor nodes, which can be used as a lower bound for the average hop distance.

The expected physical distance between n_i and n_j is the expected distance between two randomly picked points in a square of area $a \times a$, which is known to be $0.52a$ (Ghosh, 1951). Therefore, in LGKE, for n_i to n_x, and n_y to n_j, the expected distance is $\bar{\lambda}_{LGKE} = 0.52a$.

Let $\bar{\lambda'}_{LGKE}$ be the expected distance between n_x and n_y. Since two neighboring groups may have two adjacent methods: along an edge or at a corner, we can define that $\bar{\lambda'}_{\bigoplus}$ be the expected distance between two random points picked randomly from neighboring squares that are vertically (or horizontally) disposed (along an edge), and $\bar{\lambda'}_{\bigotimes}$ be the expected distance between two random points picked from neighbors that are diagonally disposed (at a corner). According to the Bayes theory, we have

$$\bar{\lambda'}_{LGKE} = Pr[\bigoplus]\bar{\lambda'}_{\bigoplus} + Pr[\bigotimes]\bar{\lambda'}_{\bigotimes} \tag{4.14}$$

where $Pr[\bigotimes]$ is the probability that neighboring squares are horizontally, and $Pr[\bigoplus])$ is the probability that neighboring squares are diagonally disposed.

Since the expected distance between two random points in an $a \times 2a$ rectangle is $0.804a$ (Ghosh, 1951), and these two points are from the same square or the different square with the probability 0.5, we can use the linearity of expectation to obtain $0.804a = 0.5\bar{\lambda}_{LGKE} + 0.5\bar{\lambda'}_{\bigoplus}$ such that $\bar{\lambda'}_{\bigoplus} = 1.088a$.

For $\bar{\lambda'}_{\bigotimes}$, considering two random points in a $2a \times 2a$ square, which consists of four $a \times a$ squares. The expected distance between two random points in a $2a \times 2a$ square is the double of 0.52, or $1.04a$. The probability that these two points originate from the same $a \times a$ square is 0.25. The probability is 0.5 that they come from two horizontally or vertically adjacent

$a \times a$ squares and is 0.25 that they come from two diagonally disposed $a \times a$ squares. We can obtain $1.04a = 0.25\bar{\lambda}_{LGKE} + 0.5\bar{\lambda'}_{\bigoplus} + 0.25\bar{\lambda'}_{\bigotimes}$ and $\bar{\lambda'}_{\bigotimes} = 1.464a$.

For $Pr[\bigotimes]$ and $Pr[\bigoplus]$, n_i needs to communicate with a neighbor n_j from a vertically adjacent square only when n_i is within a distance r from the top edge of its cell. Similarly, n_i needs to communicate with a neighbor from a diagonally disposed square only when n_i is within a distance r from the corner, that is, inside the quarter circle area. Using the area ratio, we have, $Pr[\bigotimes] = \frac{\pi r^2}{\pi r^2 + 4ar^2}$ and $Pr[\bigoplus] = \frac{4ar^2}{\pi r^2 + 4ar^2}$. By substituting to Eq. 4.14, we can get $\bar{\lambda'}_{LGKE} = \frac{4.35a^2 + 1.46\pi ra}{4a + \pi r}$. Finally, the path key communication overhead can be expressed as:

$$\begin{aligned} H_{LGKE} &= 2\bar{h}_{LGKE} + \bar{h'}_{GKE} \\ &= \frac{1.04\sqrt{A/g}}{r} + \frac{4.35A/g + 1.46\pi r\sqrt{A/g}}{(4\sqrt{A/g} + \pi r)r} \end{aligned} \tag{4.15}$$

- Inter-non-neighboring-group Key Communication Overhead

As shown in Fig. 4.5, n_i in G_u wants to establish the key agreement with n_t in G_w, it will involve in messages from n_i to n_t, through c_x, c_n, the base station, c_m and c_y. Similar to the path key communication overhead analysis the total hops from n_i to n_t can be expressed as: $H(n_i, n_t) = h(n_i, c_x) + h(c_x, c_n) + h(c_n, \cdots) + h(\cdots, BS) + h(BS, \cdots) + h(\cdots, c_m) + h(c_m, c_y) + h(c_y, n_j)$, where "$\cdots$" means we do not know the exact number of hops from the original group to the base station, or from the base station to the source group. A shortest path selection algorithm can be used to determine the hops. If H'_{LGKE} is the expected number of hops for inter-non-neighboring-group key establishment in LGKE, the linearity of expectation leads to $H'_{LGKE} = 2\bar{h}_{LGKE} + \beta * \bar{h''}_{LGKE}$, where $\bar{h''}_{LGKE}$ is the expected hop distance between the cluster heads, or from the cluster heads to the base station, β is the expected hops between the intermediate cluster heads and the base station, or specifically, between c_n, c_m and the base station, while $\bar{h}_{LGKE}$ is $\frac{0.52\sqrt{A/g}}{r}$ as calculated in Section 4.3.4.

For $\bar{h''}_{LGKE}$, we can consider the communication between the cluster heads and the base station as within a bigger group with the square size $\tilde{a} \times \tilde{a}$, therefore, $\bar{h''}_{LGKE}$ can be simplified to $0.52\tilde{a}/r$. The simulation result shows that the expected number of hops β is 6 when the number of sensor nodes equals to $50,000$ and the corresponding cluster heads equals to $1,500$ $(3 * 50,000/100)$ as shown in Fig. 4.8. Also, after carefully deployment of the base station (Pan et al., 2005a), the coefficient β can be further smaller. Finally, the inter-non-neighboring-group key communication overhead can be expressed as:

$$H'_{LGKE} = 2\bar{h}_{LGKE} + \beta * \bar{h''}_{GKE} = \frac{1.04\sqrt{A/g}}{r} + \frac{3.12\tilde{a}}{r} \tag{4.16}$$

Fig. 4.8 shows the average number of hops to establish a path key in PIKE, GKE and the lower layer of LGKE. The average hops to establish an inter-non-neighboring-group key for the top layer in LGKE is also given. The network size varies from 10,000 to 50,000

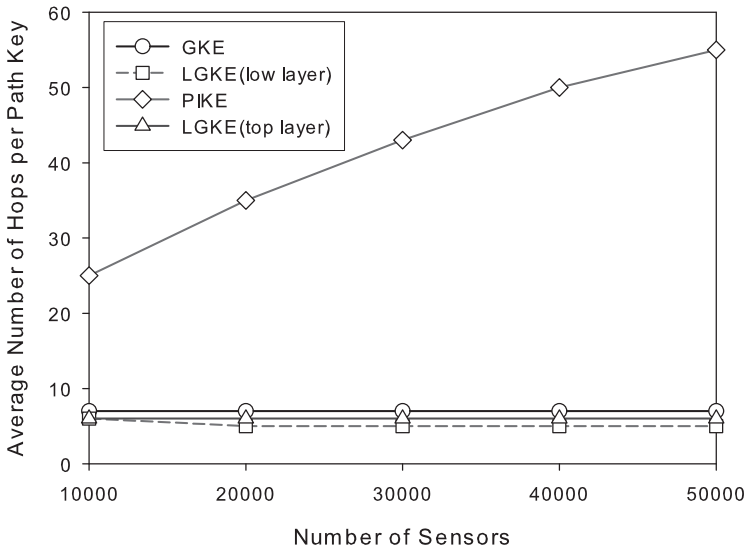

Figure 4.8: Average number of hops v.s. network size using GKE, PIKE and LGKE. The *GKE*, *PIKE* and *LGKE (low layer)* plots show the average number of hops to establish a path key. The *LGKE(top layer)* plot shows the average number hops to establish an inter-non-neighboring-group key. The network size varies from 10,000 to 50,000 with fixed group size 100.

with a fixed group size 100. It shows that LGKE outperforms GKE and PIKE with smaller number of hops and hence lower communication overhead. It benefits from the novel layered architecture of LGKE. We may notice that, in Du's schemes (Du et al., 2003), communication between two key-space-sharing sensors only involve in local connectivity, therefore the average number of hops required on a route connecting the two sensors is pretty small. When $Pr(local) = 0.3$, the expected value of hops is about 2.2, which is smaller than that of PIKE, GKE and LGKE.

Fig. 4.9 shows the average number of hops per path key with the cluster heads per group changing from 1 to 10 in LGKE. It can be seen that the number of hops increases with the increase of number of cluster heads. This is because more cluster heads lead higher probability for the packets to be transmitted between the cluster heads. Considering both survivability and the resource consumption, we choose the number of cluster head be three instead of one or two even though one or two cluster heads result in the same average of hops.

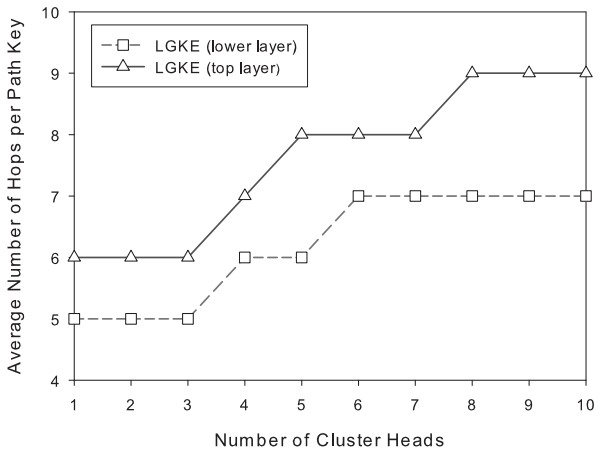

Figure 4.9: Average number of hops v.s. number of cluster heads per group. The average number of hops to establish a path key and an inter-non-neighboring-group key are given for the number of cluster heads varying from 1 to 10 in a network with 50,000 nodes with group size 100.

4.4 Summary

Establishing pairwise keys for sensors is one of the most challenging security issues for wireless sensor networks. It provides the QoS for confidential communications for some critical applications. This chapter presents LGKE, a novel layered group-based key establishment QoS scheme for wireless sensor networks. LGKE has a number of advantages over existing schemes in terms of QoS. First, it supports dynamic scalability for large scale networks through its layered architecture and EBS techniques. Second, its hierarchical architecture conforms to real world settings, making it suitable for a wide range of applications. Third, LGKE is resilient against node capture attacks, due to the uniqueness of pairwise key establishment mechanism. Finally, LGKE involves only local communication to establish pairwise keys which reduces communication overhead as well. The QoS analysis and the quantitative evaluation show the superiority of LGKE with regard to scalability, resilience, robustness and communication overhead.

Chapter 5

Conclusion and Future Directions

As described in Chapter 1, wireless sensor networks are resource constrained, and vulnerable to attacks. Some researches attempt to solve the energy efficient problems in WSN. Other researches pay more attentions to address the security problems. In this dissertation, we have focused on in finding a QoS provision for WSN considering both the energy efficient issues and the security issues. In this chapter, we first summarize the finding for QoS provision in terms of energy efficient routing, energy efficient coverage and key establishment schemes, and then briefly present the direction for the future research.

5.1 Summary of Findings

In Chapter 2, we presented a energy-aware QoS routing protocol for sensor networks. We built a model for a two-layered sensor network by using a deployment method. We then derived a general linear programming formulation for the network lifetime maximization problem. Finally we presented our heuristics algorithms to achieve optimal network lifetime by finding the QoS paths. It is the first approach in literature to develop the hybrid energy-efficient routing protocol while minimizing the communication energy consumption using the nodes deployment knowledge.

In Chapter 3, we presented the energy-aware coverage and connectivity scheme for WSN. In order to save energy and obtain better QoS performance, we developed a linear prediction model to activate and deactivate sensor nodes in advance, we adopted dynamic coverage configuration protocol to maintain coverage, and we used SPAN to build the communication backbone. We also provided QoS evaluation metrics in term of energy consumption, mean delay time until detection, and responsive time, and probability detection were introduced to measure our algorithms. Our algorithm achieves better QoS performance in terms of the mean delay time until detection and responsive time. Our algorithm validated that sensor networks are more efficient to detect the enemy in term of lower delay time and quicker responsive time when sensor networks are used to detect huge amount of intruders.

In Chapter 4, we proposed LGKE to ensure the secured communication between the sensor nodes in a dynamic large scalable wireless sensor network. We used group-based model to deploy the sensors, and preload unique pairwise keys for robust communication. We extended the Exclusion Basic System (EBS) technique (Eltoweissy et al., 2004) to achieve dynamic scalability. The QoS evaluation results show that LGKE is more resilient against node capture attacks, more robust against communication compromised, and involves in less communication overheads.

5.2 Future Directions

In Chapter 2, we assumed the sensor nodes are static and derived our heuristic algorithms based on a centralized assumption. While it is impractical for the base station to make and send the decisions for the huge amount of sensor nodes because it involves in heavy communication overhead. Normally sensor nodes make decisions based on the information from their neighbors. Therefore, a decentralized algorithm for every sensor to select the QoS path is a possible research direction. Another interesting issue for QoS routing protocols

is the consideration of node mobility. It is because the frequent update of the position of the command node and the sensor nodes and the propagation of that information through the network may excessively drain the energy of nodes. Other possible future research for routing protocols includes the integration of sensor networks with IP-based networks (e.g. Internet).

In Chapter 3, we used the linear prediction model and CCP to schedule sensor nodes activated and deactivated to ensure energy saving and dynamic coverage. We assumed the sensing area of the sensor nodes are circle and their radius are the same. However, in some applications, the sensing radius are different. Furthermore, even the sensing area of sensor nodes are irregular, for example, video sensors have cone sensing area. The customized coverage schemes should be investigated for these applications.

In Chapter 4, we presented the Key Establishment scheme based on one channel. The key establishment schemes for multiple channels available for sensor networks should be investigated also. Another area is to explore stronger threat models. Finally, it is interesting to consider the optimal strategy against some attack models.

Bibliography

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithmes, and applications*. Prentice Hall, Inc., Upper Saddle River, NJ.

Akkaya, K. and Younis, M. (2005a). An energy-aware qos routing protocol for wireless sensor networks. *Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003)*, pages 710–715.

Akkaya, K. and Younis, M. (2005b). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349.

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Network*, 38:393–422.

Alam, S. M. N. and Hass, Z. J. (2006a). Coverage and connectivity in three-dimensional networks. *Proc. IEEE Mobicom06*.

Alam, S. M. N. and Hass, Z. J. (2006b). Topology control and network lifetime in three-dimensional wireless sensor networks. *CORR cs.NI/0609047*.

Allred, J., Hasan, A. B., Pisano, W., Panichsakul, S., Gray, P., Huang, J., Han, R., Lawrence, D., and Mohseni, K. (2007). Sensorflock: An airborne wireless sensor network of micro-air vehicles. *Sensys'07*.

Barabási, A. and Albert, R. (1999). Emergence of scaling in random networks. *Science*.

Bhardwaj, M. and Chandrakasan, A. P. (2002). Bounding the lifetime of sensor networks via optimal role assignments. *IEEE InfoCOM*, 3:1587–1596.

Bhardwaj, M. and Chandrakasan, A. P. (2003). Bounding the lifetime of sensor networks via optimal role assignments. *IEEE International Symposium on comuters and Communication*.

Biagioni, E. and Bridges, K. (2002). The application of remote sensor technology to assist the recovery of rare and endangered species. *Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, (3).

Biagioni, E. and Sasaki, G. (2003). Wireless sensor placement for reliable and efficient data collection. *Proceedings of the Hawaiii International Conference on Systems Sciences*.

Box, G. E. P. and Jenkins, G. M. (1991). Time series analysis. *Prentice Hall*.

Camtepe, S. and Yener, B. (2005). Key distribution mechanisms for wireless sensor networks: a survey. *Technical Report TR-05-07, Rensselaer Polytechnic Institute, Computer Science Department, Troy, NY*.

Cao, Q., Abdelzaher, T., He, T., and Stankovic, J. (2005). Towards optimal sleep scheduling in sensor networks for rare event detection. *IPSN05*.

Cardei, M., Thai, M. T., Li, Y., and Wu, W. (2005). Energy-efficient target coverage in wireless sensor networks. *IEEE INFOCOM*.

Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. (2001). Habitat monitoring: Application driver for wireless communications technology. *Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*.

Chakrabarti, S. and Mishra, A. (2001). Qos issues in ad hoc wireless networks. *IEEE Communications Magazine*, pages 142–148.

Chan, H. and Perrig, A. (2005). Pike: Peer intermediaries for key establishment in sensor networks. *IEEE INFOCOM, Miami, FL*, pages 524–535.

Chan, H., Perrig, A., and Song, D. (2003). Random key pre-distribution schemes for sensor networks. *IEEE Symposium on Security and Privacy, Berkeley, CA*, pages 197–213.

Chang, J.-H. and Tassiulas, L. (1999). Routing for maximum system lifetime in wireless ad hoc networks. *The 37th Annu. Allerton Conf. Communication, Control, and Computing*.

Chang, J.-H. and Tassiulas, L. (2000). Energy conserving routing in wireless ad hoc networks. *Proc. IEEE INFOCOM*, pages 22–31.

Chen, B., Jamieson, K., Balakrishnan, H., and Morris, R. (2001). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *MOBICOM2001*, page 85.

Cheng, R., Kalashnikov, D., and Prabhakar, S. (2003). Evaluating probabilistic queries over imprecise data. *proceedings of SIGMOD03*.

Cheng, R., Singh, S., and Prabhakar, S. (2005). U-dbms: A database system for managing coustantly-evolving data. *VLDB05*.

Cortes, J. and Bullo, F. (2005). Integrated coverage and connectivity configuration for energy conservation in sensor networks. *SIAM Journal on Control and Optimization*, (5):1543 – 1574.

Demetrios, Z. (2001). A glance at quality of services in mobile ad-hoc networks. *http://www.cs.ucr.edu/ csyiazti/cs260.html*.

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., and Hong, W. (2004). Model-driven data acquisition in sensor networks. *VLDB04*.

Diffie, W. and Hellman, M. (1976). Multiuser cryptographic techniques. *AFIPS Conference, New York, NY, USA*, pages 109–112.

Du, W., Deng, J., Han, Y. S., and Varshney, P. (2003). A pairwise key predistribution scheme for wireless sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington DC, USA*, pages 42–51.

Du, W., Deng, J., Han, Y. S., and Varshney, P. (2004a). A key management scheme for wireless sensor networks using deployment knowledge. *IEEE INFOCOM, Los Alamitos, CA*, pages 586–597.

Du, W., Deng, J., Han, Y. S., and Varshney, P. K. (2004b). A key management scheme for wireless sensor networks using deployment knowledge. *IEEE INFOCOM, Los Alamitos, CA*, pages 586–597.

Ellison, B., Fisher, D., Linger, A., Lipson, R. C., Longstaff, H. F., and Mead, T. (1997). Survivable network systems: An emerging discipline. *CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA*.

Eltoweissy, M., Heydari, H., Morales, L., and Sudborough, H. (2004). Combinationrial optimizations of group key management. *Journal of Networks and Systems Management*, pages 33–49.

Erdös, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Universitatis Debreceniensis*, 6:290–297.

Eschenaer, L. and Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. *ACM Conference on Computer and Communications security (CCS), Washington DC, USA*, pages 41–47.

Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. (1999). Next centuary challenges: Scalable coordination in sensor networks. *Proc. ACM/IEEE MobiCom*.

Ghosh, B. (1951). Random distance within a rectangle and between two rectangles. *Bull. Calcutta Math. Soc.*, pages 17–24.

Gui, C. and Mohapatra, P. (2004). Power conservation and quality of surveillance in target tracking sensor networks. *MobiCom04*.

Gui, C. and Mohapatra, P. (2005). Virtual patrol: a new power conservation design for surveillance using sensor networks. *IPSN05*.

Gungor, V., Vuran, M., and Akan, O. (2007). On the cross-layer interactions between congestion and contention in wireless sensor and actor networks. *Ad hoc Networks*, 5:897–909.

Haque, I. T., Assi, C., and Atwood, J. W. (2005). Randomized energy aware routing algorithms in mobile ad hoc networks. *MSWim'05, Montreal, Quebec, Canada*.

Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2000a). Energy-efficient communication protocol for wireless microsensor networks. *The Hawaiian Int. Conf. Systems Science, Hawaii, USA*.

Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2000b). Energy-eficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, USA*.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E., and Pister, K. S. J. (2000). System architecture directions for networked sensors. *Architectural Support for Programming Languages and Operating Systems*, pages 93–104.

Hou, Y. H., Shi, Y., Pan, J., and Midkiff, S. F. (2005a). Maximizing lifetime of wireless sensor networks through optimal single-session flow routing. *IEEE Trans. on Mobile Computing*, 4(5):2579–2590.

Hou, Y. H., Shi, Y., and Sherali, H. D. (2005b). On node lifetime problem for energy constrained wireless sensor networks. *Mobile Networks and Applications, DOI: 10:1007/s11036-005-4444-6*.

Hou, Y. T., Shi, Y., Sherali, H. D., and Midkiff, S. F. (2005c). On energy provisioning and relay node placement for wireless sensor network. *IEEE Trans. on Wireless Communications*, 4(5):2579–2590.

Hsin, C.-F. and Liu, M. (2004). Network coverage using low duty-cycle sensors: Random and coordinated sleep algorithms. *IPSN04*.

Huang, D., Mehta, M., Medhi, D., and Harn, L. (2004). Location-aware key management scheme for wireless sensor networks. *2nd ACM workshop on Security of Ad Hoc and Sensor Networks,Washington, DC, USA*, pages 29–42.

Hull, B., Jamieson, K., and Balakrishnan, H. (2004). Mitigating congestion in wireless sensor networks. *Sensys04*.

Intanagonwiwat, C., Govindan, R., and Estrin, D. (2003). Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, (1):2–16.

Iyer, R. and Kleinrock, L. (2003). Qos control for sensor networks. *ICC 2003*.

Kahn, J. M., Katz, R. H., and Pister, K. S. J. (1999). Next century challenges: Mobile networking for smart dust. *Proceedingsof the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 483–492.

Kalpakis, K., Dasgupta, K., and Namjoshi, P. (2002). Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor network. *Computer Network*, 42(6):697–716.

Kang, J., Zhang, Y., and Nath, B. (2007a). An optimal resource control scheme under fidelity and energy constraints in sensor networks. *ACM Wireless Networks (WINET) Journal*.

Kang, J., Zhang, Y., and Nath, B. (2007b). Tara: Topology-aware resource adaptation to alleviate congestion in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18.

Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Networks*, pages 293–315.

Kim, S., Fonseca, R., Dutta, P., Tavakoli, A., Culler, D., Levis, P., Shenker, S., and Stoica, I. (2007). Flush: A reliable bulk transport protocol for multihop wireless networks. *Sensys07*.

Knight, J., Sullivan, K. J., Elder, M. C., and Wang, C. (2000). Survivability architectures: Issues and approaches. *DARPA Information Survivability Conference and Exposition (DISCEX 2000),Hilton Head, SC, USA*, pages 157–171.

Krishnan, R. and Starobinski, D. (2006). Efficient clustering algorithms for self-organizing wireless sensor network. *Ad Hoc Networks*, 4:36–59.

Kumar, S., Lai, T. H., and Balogh, J. (2004). On k-coverage in a mostly sleeping sensor network. *Mobicom04*.

Lazos, L., Poovendran, R., and Ritcey, J. A. (2007). Probabilistic detection of mobile targets in heterogeneous sensor networks. *IPSN'07*.

Levis, P., Lee, N., Welsh, M., and Culler, D. (2003). Tossim: Accurate and scalable simulation of entire tinyos applications. *ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*.

Li, L. and Halpern, J. Y. (2001). Minimum-energy mobile wireless networks revisited. *IEEE International Conference on Communications (ICC) 2001*, 1.

Li, M., Ganesan, D., and Shenoy, P. (2006). Presto: Feedback driven data management in sensor networks. *USENIX, the 3rd Symposium on Networked Systems Design and Implementation (NSDI)*.

Li, N. and Hou, J. C. (2006). A scalable, power-efficient broadcast algorithm for wireless networks. *ACM Wireless Networks*, 12:495–509.

Li, Q., Aslam, J., and Rus, D. (2001a). Hierarchical power-aware routing in sensor networks. *Proceedings of the DIMACS Workshop on Pervasive Networking*.

Li, Q., Aslam, J., and Rus, D. (2001b). Online power-aware routing in wireless ad-hoc networks. *Proceedings of MOBICOM, Rome, Italy*, pages 97–107.

Li, X. and Yang, D. (2006). A quantitative survivability evaluation model for wireless sensor network. *IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL*.

Li, X., Yang, D., and Sawhney, R. (2009). Layered group key establishment for wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing, Accepted*.

Lindsey, S., C.Raghavendra, and Sivalingam, K. (2001). Data gathering in sensor networks using energy delay metric. *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, (San Francisco, CA)*.

Lindsey, S. and Raghavendra, C. S. (2002). Pegasis: Power-efficient gathering in sensor information systems. *Proc. of IEEE Aerospace Conference*, 3:1125–1130.

Liu, D. and Ning, P. (2003a). Establishing pairwise keys in distributed sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security,Washington D.C., USA*, pages 52–61.

Liu, D. and Ning, P. (2003b). Location-based pairwise key establishments of static sensor networks. *ACM Workshop in Security in Ad Hoc and Sensor Networks, Fairfax, Virginia*, pages 72–82.

Liu, D., Ning, P., and Du, W. (2005). Group-based key pre-distribution in wireless sensor networks. *Proceedings of 2005 ACM Workshop on Wireless Security (WiSe 2005), New York, NY*, pages 11–20.

Lu, C., Xing, G., Chipara, O., Fok, C.-L., and Bhattacharya, S. (2005). A spatiotemporal query service for mobile users in sensor networks. *International Conference on Distributed Computing Systems (ICDCS)*.

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA*.

Malan, D., Welsh, M., and Smith, M. (2004). A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. *First IEEE International Conference on Sensor and Ad Hoc Communications and Networks(SECON04),Santa Clara, California*, pages 71–80.

Manjeshwar, A. and Agarwal, D. P. (2001). Teen: a routing protocol for enhanced eciency in wireless sensor networks. *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*.

Manjeshwar, A. and Agarwal, D. P. (2002). Apteen: A hybrid protocol for ecient routing and comprehensive information retrieval in wireless sensor networks. *Parallel and Distributed Processing Symposium, Proceedings International, IPDPS*.

Meguerdichian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. (2001a). Coverage problems in wireless ad-hoc sensor networks. *Proc. IEEE Infocom01*.

Meguerdichian, S., Koushanfar, F., Qu, G., and Potkonjak, M. (2001b). Exposure in wireless ad-hoc sensor networks. *Mobile Computing and Networking*, pages 139–150.

Paek, J. and Govindan, R. (2007). Rate-controlled reliable transport for wireless sensor networks. *Sensys07*.

Pan, J., Cai, L., Hou, Y., Shi, Y., and Shen, S. (2005a). Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Trans. on Mobile Computing*, (5):458–473.

Pan, J., Cai, L., Hou, Y. H., Shi, Y., and Shen, S. X. (2005b). Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Trans. on Mobile Computing*, 4(5):458–473.

Perrig, A., Canetti, R., Tygar, D., and Song, D. (2002). The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5(2):2–13.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J. (2001). Spins: Security protocols for sensor networks. *Proc, of ACM Mobicom'01, Rome, Italy*, pages 189–199.

Pottie, G. J. and Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58.

Ren, S., Li, Q., Wang, H., Chen, X., and Zhang, X. (2005). Analyzing object tracking quality under probabilistic coverage in sensor networks. *ACM MC2R*, (1).

Ren, S., Li, Q., Wang, H., Chen, X., and Zhang, X. (2007). Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks. *IEEE Trans. on Parallel and Distributed Systems*, (3):334–350.

Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, (2):120–126.

Rodoplu, V. and Meng, T. H. (1999). Minimum energy mobile wireless networks. *IEEE Journal Selected Areas in Communications*, 17(8).

Rytter, A. (1993). Vibration based inspection of civil engineering structures. *Ph. D. dissertation, Dept. of building technology and structural engineering, Aalborg University, Denmark.*

Sadagopan, N. and Krishnamachari, B. (2004). Maximizing data extraction in energy-limited sensor networks. *IEEE InfoCOM*, pages 1717–1727.

Sankar, A. and Liu, Z. (2004). Maximum lifetime routing in wireless sensor networks. *Proc. IEEE INFOCOM*, pages 1089–1097.

Schwiebert, L., Gupta, S. K. S., and Weinmann, J. (2001). Research challenges in wireless networks of biomedical sensors. *Mobile Computing and Networking*, pages 151–165.

Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Stankovic, W. K. J., Young, D., and Porter, J. (2003). Luster: Wireless sensor network for environmental research. *Sensys'07.*

Siek, J., Lee, L.-Q., and Lumsdaine, A. (2000-2001). Boost graph library. http://www.boost.org/libs/graph/doc/index.html.

Simplot-Ryl, D., Stojmenović, I., and Wu, J. (2005). *Engery-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks.* John Wiley & Sons Inc., New York, NY.

Singh, S., Woo, M., and Raghavendra, C. (1998). Power-aware routing in mobile ad hoc networks. *Proc. ACE/IEEE MOBICOM98, Dallas, TX*, pages 181–190.

Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. J. (2002). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, (5).

Steere, D. C., Baptista, A., McNamee, D., Pu, C., and Walpole, J. (2000). Research challenges in environmental observation and forecasting systems. *Proceedings of the sixth annual international conference on Mobile computing and networking*, pages 292–299.

Tian, D. and Georganas, N. (2003). A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing Journal*.

Toh, C. K. (2001). Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communication Magazine*, pages 138–147.

Wan, C.-Y., Eisenman, S. B., and Campbell, A. T. (2003). Coda: Congestion detection and avoidance in sensor networks. *Sensys03*.

Wang, H., Elson, J., Girod, L., Estrin, D., and Yao, K. (2003). Target classification and localization in habitat monitoring. *Proceedings of the IEEE ICASSP 2003, Hong Kong*.

Wang, L. and Kulkarni, S. S. (2005). pcover: Partial coverage for long-lived surveillance sensor networks. *Technical Report, Michigan State University, MSU-CSE-05-30*.

Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., and Welsh, M. (2005a). Monitoring volcanic eruptions with a wireless sensor network. *In Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05)*.

Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., and Welsh, M. (2005b). Monitoring volcanic eruptions with a wireless sensor network. *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05)*.

Woo, A., Tony, T., and Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. *ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

Wui, K. and Harms, J. (2001). Qos support in mobile ad hoc networks. *Crossing Boundaries C an interdisciplinary Journal*, (1).

Xing, G., Wang, X., Zhang, Y., Lu, C., Pless, R., and Gill, C. (2005). Integrated coverage and

connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, (1):36–72.

Yan, T., He, T., and Stankovic, J. A. (2003). Differentiated surveillance service for sensor networks. *SenSys03*.

Ye, F., Zhong, G., Lu, S., and Zhang, L. (2003). Peas: A robust energy conserving protocol for long-lived sensor networks. *ICDCS03*.

Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. *INFOCOM02*.

Younis, M., Akkaya, K., and Kunjithapatham, A. (2002). Optimization of task allocation in a cluster-based sensor network. *IEEE InfoCom, 2002*.

Yu, Y. and Prasanna, K. (2003). Energy-balanced task allocation for collaborative processing in wireless sensor network. *ACM LCTES 2003*.

Zeinalipour, D., Aristeidou, S., and Kazeli, S. (1999). Ip quality of services (in greek). *http://www.cs.ucr.edu/ csyiazti/downloads/papers/ipqos/papers/ip-qos.pdf*.

Zhang, B. and Mouftah, H. T. (2006). Energy-aware on-demand routing protocols for wireless ad hoc networks. *ACM Wireless Networks*, 12:481–494.

Zhao, J. and Govindan, R. (2003). Understanding packet delivery performance in dense wireless sensor networks. *ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*.

Zhou, L., Ni, J., and Ravishankai, C. V. (2005). Efficient key establishment for group-based wireless sensor deployments. *Proceedings of 2005 ACM Workshop on Wireless Security (WiSe 2005), New York, NY*, pages 1–10.

Zhou, L., Ni, J., and Ravishankai, C. V. (2006). Supporting secure communication and data collection in mobile sensor network. *IEEE INFOCOM,Catalunya, Spain*, pages 1–12.

Zhu, S., Setia, S., and Jajodia, S. (2003). Leap: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Conference on Computer and Communications Security (CCS), Washington DC, USA*, pages 62–72.

Zussman, G. and Segall, A. (2003). Energy efficient routing in ad hoc disaster recovery networks. *Proc. IEEE INFOCOM*, 1:682–691.

Vita

Dengfeng Yang was born in a small village of Hubei Province, China, when his mother was working on a cotton field. A barefoot doctor delivered him with the help of his grandmother. After spending 18 years at his hometown, he attended University of Science and Technology of China, where he received Bachelor of Science in 2001, and Master of Engineering in 2004 from the department of Precision Machinery and Instrumentation. In the spring of 2005, Dengfeng was enrolled into The University of Massachusetts, Amherst. After one year at Amherst, in 2006, he transferred to The University of Tennessee, Knoxville in department of Industrial and Information Engineering as a doctoral student under the direction of incomparable Dr. Li, where he got a Ph.D. degree. He will work as a research associate at the department of Industrial Engineering and Management Sciences, Northwestern University. His permanent address is Room $6-503$, Longhe Rd. $\sharp 4$, Hefei, Anhui, China, P.R., 230031.