SMART WIRELESS SENSOR NETWORKS

Edited by **Mr. Hoang Duc Chinh and Dr. Yen Kheng Tan** (Editor-in-Chief)

INTECHWEB.ORG

Smart Wireless Sensor Networks
Edited by Mr. Hoang Duc Chinh and Dr. Yen Kheng Tan (Editor-in-Chief)

Published by InTech
Janeza Trdine 9, 51000 Rijeka, Croatia

Publishing Process Manager Jelena Marusic
Technical Editor Goran Bajac
Cover Designer Martina Sirotic
Image Copyright Gemenacom, 2010. Used under license from Shutterstock.com

First published December, 2010
Printed in India

A free online edition of this book is available at www.intechopen.com
Additional hard copies can be obtained from orders@intechweb.org

Smart Wireless Sensor Networks, Edited by Mr. Hoang Duc Chinh and Dr. Yen Kheng Tan (Editor-in-Chief)
p. cm.
ISBN 978-953-307-261-6

Contents

Preface

For the past decade, there has been rapid development and advancement in the communication and sensor technologies that results in the growth of a new, attractive and challenging research area – the wireless sensor network (WSN). A WSN, which typically consists of a large number of wireless sensor nodes formed in a network fashion, is deployed in environmental fields to serve various sensing and actuating applications. With the integration of sensing devices on the sensor nodes, the nodes have the abilities to perceive many types of physical parameters such as, light, humidity, vibration, etc. about the ambient conditions. In addition, the capability of wireless communication, small size and low power consumption enable sensor nodes to be deployed in different types of environment including terrestrial, underground and underwater. These properties facilitate the sensor nodes to operate in both stationary and mobile networks deployed for numerous applications, which include environmental remote sensing, medical healthcare monitoring, military surveillance, etc. For each of these application areas, the design and operation of the WSNs are different from conventional networks such as the internet. The network design must take into account of the specific applications. The nature of deployed environment must be considered. The limited of sensor nodes' resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. As such, a smart wireless sensor network, able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of network's operation for a maximized lifetime, has been illustrated.

In this smart wireless sensor network (WSN) book, various aspects of designing a smart WSN have been investigated and discussed. The main topics include: advances in smart wireless ad hoc and sensor networks, algorithms and protocols for smart WSN management and performance and quality of service (QoS) of smart WSNs. Several key issues, challenges and state-of-the-art methods for designing and developing smart WSNs will be addressed throughout the 23 chapters of this book. Chapter 1 presents communication protocol stacks for WSNs which include physical layer, medium access control layer and network layer. State-of-the-art solutions applied in different layers to guarantee the communication reliability are discussed and evaluated. Novel communication protocols and simulation tools are proposed to enhance the performance and reliability of smart sensor systems. Chapter 2 discusses the factors that may influence the desired operation of WSNs. The impact of sensor nodes characteristics and network deployment on WSNs' performance are investigated. WSNs' information functions including the parameters and method of evaluating data importance are also presented. Chapter 3 and 4 focuses on design methodologies for WSNs. Chapter

3 provides a survey of cross-layer protocol design frameworks and define some major criteria to evaluate these frameworks. Meanwhile, chapter 4 proposes a novel model which applies the concept of intelligent multi-agent system on designing distributed sensor networks.

Chapter 5 to 11 present various protocols and algorithms proposed for WSNs with the expectation of improving communication efficiency, saving energy and maximizing network lifetime. Chapter 5 deals with a broadcast storm problem, an efficient broadcast protocol is proposed in order to achieve maximum lifetime of the WSNs. Chapter 6 focuses on developing multi-hop routing protocol for WSNs which consists of unavailable nodes due to failure. The protocol is designed and implemented in real sensor nodes. Experiments are conducted to evaluating the performance of the networks. Chapter 7 introduces a relational model that represents the dependences between nodes of the network and defines the actions of these nodes in different situations. Based on this model, communication activities of the network are managed in order to route the message from nodes to the base station efficiently. Chapter 8 presents a framework for an effective support of mobility in WSNs. The approach is using the mobile IPv6 protocol, the Neighbor Discovery for finding sink nodes and subsequent node registration, and the soft hand-off mechanisms for maintaining connectivity of moving nodes. In chapter 9, game theoretic model is applied to form cluster-based WSNs. A cooperative game theoretic clustering algorithm is proposed for balancing energy consumption of sensor nodes and increasing network lifetime. The system-wide optimization is obtained from the conditions of cooperation, each sensor node tradeoff individual cost with the network-wide cost. Chapter 10 shows another energy-efficient cluster formation method. The optimized clustering structure is achieved by preventing unequal size of clusters, finding the optimal number of nodes in a cluster, and re-electing cluster head for balancing local cluster. Chapter 11 deals with the problem of maximizing the covered area of 3-dimensional WSNs. A distributed algorithm is developed and executed at sensor nodes to establish a connected topology while maximize the covered sensing area of the network.

Chapter 12, 13, and 14 introduce novel techniques and mechanisms used for managing the Quality of Service (QoS) of WSNs. Chapter 12 provides the understand of QoS mechanisms, presents research on an instance of QoS and shows the improvement achieved by applying this instance. Chapter 13 presents a new method which can be used to guarantee various level of communication reliability in WSNs. A flexible loss recover mechanism is proposed and the tradeoff between end-to-end delays and memory requirements for different levels of communication reliability is evaluated. Chapter 14 focuses on improving the transmission energy consumption of WSNs while the QoS of communication is guaranteed. Chapter 15 and 16 discuss the time synchronization techniques for WSNs. Chapter 15 provides an overview of time synchronization in WSNs. Fundamental techniques, influenced factors, uncertainties and errors, as well as evaluating metrics of time synchronization are identified. Different time synchronization methods are presented and evaluated. Chapter 16 focuses on time synchronization for underwater WSNs. The typical attributes of this type of WSNs are addressed; the effect of underwater environment on the performance of a specific time synchronization algorithm is studied and demonstrated through simulation.

Chapter 17 to 23 present the security problems in WSNs. Chapter 17 gives an introduction of security threats in WSNs, classify security management method into different categories, discuss and suggest future research issues on security of WSNs. Chapter 18 proposes a compromise-resilient pair-wise rekeying protocol in a three-tier WSN. Performance analysis of this method shows that it is significantly improve the security level in order to prevent the stealth of secret information of the network during node capture attack. Chapter 19 focuses on detecting node capture attacks in WSNs in order to avoid the harm created by attackers to WSNs. Chapter 20 introduces a security architecture that provides confidentiality, integrity and authentication with trust management for WSNs. A cross-layer wireless sensor network trust model based on cloud model is also developed and proved to be able to decrease trust risk of nodes and enhance successful cooperation ratio of WSN's system. Chapter 21 highlights the security problems at the physical layer and hardware platform. Security challenges and potential physical attacks in WSNs are listed; the trusted platform and security architecture for sensor nodes are also presented. Chapters 22 and 23 describe technologies and architectures of WSNs. A special type of WSNs, wireless multimedia sensor networks (WMSNs), is highlighted and studied. This chapter also discusses and compares different hardware platforms and architectures for WMSNs.

In summary, with a variety of design and development aspects being considered and discussed, the concept introductions and research discussions of this smart wireless sensor network (WSN) book are expected to benefit both the industry developers working in sensor network systems, as well as the researchers and graduate students conducting research on WSNs. The editor would like to take this opportunity to thank all the authors for their kind contributions and to all those people who have directly or indirectly helped to make this work possible. Special thanks are also presented to Yen Kheng Tan, chief editor of Smart and Sustainable WSN book series, and Mrs Jelena Marusic, process manager, whom are responsible for the coordination of this entire project.

Mr. Hoang Duc Chinh and Dr. Yen Kheng Tan (Editor-in-Chief)

Part 1

Overview and Design Methodology

1

Advanced Communication Solutions for Reliable Wireless Sensor Systems

Jari Nieminen, Shekar Nethi, Mikael Björkbom,
Aamir Mahmood, Lasse Eriksson and Riku Jäntti
Aalto University, School of Science and Technology
Finland

1. Introduction

State-of-the-art Wireless Sensor Network (WSN) technology enables design and implementation of novel, intriguing applications that can be used to address numerous industrial, environmental, societal and economical challenges and thus, the importance and potential of WSNs are constantly growing. Wireless sensor nodes constituting a WSN consist of a sensor interface, microcontroller, memory and battery units together with a radio module. Hence, wireless sensor nodes are able to carry out distributed sensing and data processing, and to share the collected data using radio communications. In the beginning the development of wireless sensors was driven by military applications but the introduction of civilian wireless sensor systems has greatly diversified application domain which has further boosted research efforts in the field of wireless sensor networks. Present state of the evolution of wireless sensor nodes allows utilization of smart sensors to enhance the performance and robustness of WSNs.

From the communication engineering point of view the large number of possible applications, see e.g. (Römer & Mattern, 2004), introduces unforeseen problems for which classical communication solutions are not suitable while smart sensors give us tools for finding answers to these new-found questions. Furthermore, a large number of communication protocols have been designed for specific applications but the lack of generic solutions brings up problems with respect to large scale economic success. Since versatility of WSN applications is unimaginable and the amount of possible operation scenarios is unlimited, designed protocols should be suitable for various purposes of use. Consequently, scalability and flexibility of technical solutions are extremely important to enable economic feasibility of energy-constrained wireless sensor networks.

The chapter discusses new communication protocols and state-of-the-art design methodologies as well as good practices that together enable reliable operation of various wireless sensor networks. We especially focus on reliability issues since many WSN applications are located in troublesome environments. For example, in the context of industrial WSNs reliability has been denoted as one of the fundamental design goals

(Gungor & Hancke, 2009). In this chapter we only consider so called media layers, i.e. physical, data link and network layers, and exclude upper layers. Naturally, research efforts in the field of WSNs include various other aspects as well and we direct an interested reader to see (Yick et al., 2008) and (Akyildiz et al., 2002) for comprehensive surveys.

The main contributions of this chapter include a review of current technologies used in wireless sensor networks and of the state-of-the-art solutions. We also discuss and propose novel communication protocols to enhance the performance and reliability of smart sensor systems. In each of the sections we present a comprehensive literature review and give the main references for an interested reader to further pursue on the topics. In the end of each section current state-of-the-art solutions will be introduced along with measurement and/or simulation results.

The chapter is outlined as follows. First, we review several existing physical layer methods that can be used to improve the reliability of WSNs and discuss utilization of antenna diversity in this context. After this, we cover possible media access mechanisms to guarantee data transmissions by considering both, single- and multi-channel systems. Next, solutions for enhancing reliability on the network layer are studied. Finally, we will investigate some practical WSN applications, mainly focusing on wireless automation and control, with a full-scale simulator to validate and justify the proposed designs.

2. Physical Layer and Diversity for Reliability

The main task of physical layer algorithms is to enable reliable delivery of bit streams over physical medium by carrying out transmission, reception and signal modulation. Other objectives include cooperation with the Media Access Control (MAC) layer to ensure error-free communications and providing channel information for MAC layer to make operational decisions. Due to the inherent characteristics of WSNs, physical layer solutions have strict limitations in terms of energy consumption and processing power compared to traditional wireless systems. Hence, the sensors' hardware abilities have to be taken into account while designing physical layer solutions.

In the context of wireless sensors, several options for transmission medium exist. Optical communications, such as laser and infrared, can be exploited if a line-of-sight connection between a transmitter and receiver is available. On the other hand, in underwater WSN applications acoustic communications are used due to the signals attenuation properties of water (Akyildiz et al., 2005). Nevertheless, undoubtedly most of the current WSN applications use radio frequencies and exploit global, unlicensed frequency bands, for example the Industrial, Scientific and Medical (ISM) band, for communications. Therefore, we focus exclusively on these particular frequency bands in this chapter.

This section consists of two main parts. In the first part we present and discuss existing physical layer methods, such as signal multiplexing, modulation and error coding, by focusing especially on reliability issues. In the second part we consider exploitation of antenna diversity in advanced sensor systems and present measurement results which imply that antenna diversity should be exploited to improve reliability in WSNs.

2.1 Bandwidth, Multiplexing and Modulation

In general, physical layer techniques in WSNs can be divided into three different classes based on bandwidth requirements: narrow band, spread spectrum and ultra-wideband (Yick et al., 2008). As the name indicates, narrow band systems utilize only a small portion of spectrum which approximately corresponds to the used symbol rate. Although bandwidth efficiency is the strength of narrow band systems, i.e. achieved data rate over bandwidth is high, narrow band systems are very vulnerable to interference, jamming and fading. As a consequence, narrow band systems cannot provide robust and reliable communications. Moreover, Orthogonal Frequency Division Multiplexing (OFDM) is a digital modulation scheme which divides the data into several streams and then transmits each stream on an individual subchannel. In OFDM, subchannels are closely-spaced while still ideally orthogonal. Each of the subchannel s can be treated separately (e.g. modulation) and hence, data rate of each subchannel is equal to narrow band systems using the same band. Although OFDM is widely used in wireless communications, complexity and processing power requirements of OFDM are unacceptably high for current sensor nodes.

In spread spectrum technologies the bandwidth of the original signal is expanded over a wider frequency band using a spreading function. In fact, the spreading function defines the used bandwidth and thus, the final bandwidth is independent of the bandwidth of the original signal. Spread spectrum systems are characterized by low transmission powers and robustness to narrow-band interference. In addition, impairments caused by multipath fading of signals can be cancelled effectively compared to simple narrow band systems. Spread spectrum signals appear as noise-like signals at unwanted receivers and therefore, the technology offers resistance against jamming and eavesdropping as well. Furthermore, since the data signal is spread over a wider frequency band for transmission and transformed back to the original format at the receiver using the same spreading function, spread spectrum approaches offer *spreading gain* which is defined by the transmitted bandwidth divided by the information bandwidth. By multiplying the received signal with the particular spreading code the desired signal can be raised over the noise floor which helps detection and thus, enables multiple users to access the same band simultaneously.

Ultra-wideband (UWB) systems utilize even wider frequency bands than spread spectrum technologies. UWB systems spread data signals over frequency bands of gigahertz and as a result, UWB devices use low transmission powers such that UWB signals are buried under other signals without interfering existing systems. In general, UWB technology is suitable for short-range data transmissions. However, development of UWB technology in the field of WSNs has been slow and large-scale deployment of UWB technology in WSNs is still to be seen, even though the IEEE 802.15.4a standard includes an UWB option (IEEE 802.15.4a, 2007). To conclude, spread spectrum technology has several advantages compared to other approaches in the context of reliable communications in WSNs and thus, it is natural that spread spectrum is the most popular physical layer method used in existing WSNs.

Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) are the main methods in the class of spread spectrum technologies. In the basic form of DSSS the signal is multiplied by a fixed code to spread the original data signal over a wider band. Several wireless communication systems exploit DSSS such as IEEE 802.11b (IEEE

802.11, 2007) and IEEE 802.15.4 (IEEE 802.15.4, 2006). On the other hand, FHSS devices hop on different frequency channels based on a predetermined pseudorandom code during the operations. Advanced version of the basic FHSS is used in Bluetooth, which is based on (IEEE 802.15.1, 2005), where hopping patterns are adjusted depending on the experienced channel conditions such that better quality channels are exploited more often.

In digital communication systems digital bit streams are transmitted over analog channels. For this, bits have to be transformed from digital representation form to analog symbols. This digital-to-analog conversion is carried out by digital modulation which can be done in several ways, such as using phase (PSK), frequency (FSK) or amplitude shift keying (ASK). Moreover, if at least two different phases and amplitudes are used, we have quadrature amplitude modulation (QAM). In general, the more digital bits an analog symbol represents the higher the data rate, however, in the meantime reliability is compromised since the probability of symbols' misinterpretation increases. Hence, while choosing the used modulation scheme a trade-off between data rate, reliability and transmission range has to be made. For example, in the 2.4 GHz band IEEE 802.15.4 utilizes Orthogonal-QPSK and spreading is enforced by using 4 bits to select 1 out of 16 different 32-bit code words.

2.2 Coding for Error Control

Due to the rigorous energy consumption constraints minimization of transmission powers is extremely important in WSNs. Reduction of the transmission power decreases the Packet Delivery Ratio (PDR) due to the nature of the radio environment such that fewer packets can be received. However, lower signal to noise ratios can be compensated by error control coding and thus, reliability of packet transmissions can be improved. On the other hand, efficient error coding allows longer hop distances with the same transmission power while sufficient PDR is maintained.

In wireless communication systems error correction schemes can be divided into three categories based on operation principles: Automatic Repeat and Request (ARQ), Forward Error Correction (FEC) and Hybrid ARQ (HARQ). If a packet transmission fails for some reason and the packet cannot be decoded properly at the receiver, the straightforward solution is to retransmit the entire packet again. This kind of approach is called Automatic Repeat and Request (ARQ). The purpose of Forward Error Correction (FEC) approach is to enhance error resiliency by including redundant information to packets such that decoding is possible even though some bits are misinterpreted. By combining both of these approaches we get Hybrid ARQ (HARQ) schemes which aim to improve reliability by adding redundant bits in an incremental fashion depending on the number of experienced packet losses. HARQ –based schemes can be further sorted into two categories, Type I and Type II, depending on the information included in retransmitted packets. In Type I HARQ –schemes receivers do not store packets whereas in Type II HARQ –schemes packets are stored which enables soft combining of multiple packets.

Several FEC algorithms have been developed during the evolution of communication systems. For example, convolutional codes are utilized in countless applications to provide trustworthy delivery of packets by adding redundancy to bit streams. Each m bit stream is converted to n symbols such that the input stream is convoluted with the impulse response

of the encoder. Several research articles consider the applicability of convolutional codes for WSNs, see e.g. (Sankarasubramaniam, 2003), and the general conclusion is that the power consumption of such codes is too large for WSNs. Furthermore, by exploiting rateless codes, such as Raptor codes (Shokrollahi, 2006), near optimal performance can be achieved. Nevertheless, rateless codes are in general unsuitable for WSNs since extremely large payloads are required for efficient operations and usually payloads in various WSN applications are relatively small.

The most prominent class of FEC codes in WSN applications encompasses of BCH codes. BCH codes are linear, cyclic block codes which use especially selected generator polynomials for encoding. Decoding of BCH codes can be done in an efficient manner which makes such codes feasible for sensor systems. Codes in this class can be designed to match the requirements of various applications. This kind of flexibility enables effective utilization of error codes. For example, the Reed-Solomon codes, which are extensively exploited in communication networks, belong to this category of error coding. To summarize, although several FEC codes have been designed to optimize the performance with respect to certain radio environments, packet sizes and reliability constraints, in the end BCH codes seem to be the most suitable for WSNs (Vuran & Akyildiz, 2009). However, even though decoding can be done in low complexity, the encoding process is typically computationally intensive and requires special purpose digital signal processors. Hence, most sensor systems are not using any kind of FEC currently. Instead, only Cyclic Redundancy Check (CRC) is used for error detection, where a check sum is calculated from the raw data using a predetermined code.

2.3 Antenna Diversity

Co-existence of high power wideband wireless local area networks (WLANs) and low power wireless sensor networks on unlicensed ISM bands is challenging. Several studies have investigated the coexistence problem of IEEE 802.11 family radios (WLAN) and IEEE 802.15.4 (WSN) radios, see e.g. (Polepalli et al., 2009). The general conclusion is that coexistence on the same band is possible if there is enough spatial separation between the systems or channel utilization of the WLAN is below a certain threshold. In case of IEEE 802.11b/g transmitters, three IEEE 802.15.4 channels are "sub-orthogonal" to the WLAN channels. That is, they only experience adjacent channel interference which is at least 30 dB lower than the interference on the signal band. For IEEE 802.11n, the situation gets worse and there could be only a single IEEE 802.15.4 channel which experiences solely adjacent channel interference. Hence, in the worst case, there could be only one channel available for IEEE 802.15.4 sensor network operation which should be utilized as efficiently as possible. Because of the propagation environment antenna diversity could be utilized to mitigate the effects of fading and guarantee reliable packet delivery.

Potential of spatial diversity has not been fully exploited yet in wireless sensor networks and only some efforts have been done in this direction. In (Shin et al., 2007) experimental results to evaluate channel dynamics and delay spread of 2.4 GHz systems in an indoor multipath environment are presented whereas in (Shuaib et al., 2006), a dual band double-T printed monopole is developed and tested for 2.4 GHz and 5.2 GHz operating frequencies. Therefore, to assess the physical properties of a real radio environment and investigate the

use of antenna diversity in WSNs, measurements using real nodes were carried out in an industrial warehouse.

The measurement setup consists of a sensor node equipped with a CC2431 (802.15.4 PHY) radio module connected to an Anritsu 50 Ω 2.41-2.45 GHz portable antenna. Four receivers (compact ceramic antennas) are arranged in an array and placed at distance of 0.0625 m from each other, which is half the wavelength at 2.4 GHz. Channel 26 is used since it experiences minimal interference from other wireless devices and fading is the main cause of packet drops. Fig. 1 shows the percentage of packet drops experienced by different receivers. The data is collected at 3 different industrial environments and since the antennas are at least half a wavelength apart from each other, each receiver sees independent fading. Therefore, the percentage of successful packet reception varies for each receiver and in each location different antenna gives the best performance. Thus, we conclude that use of antenna diversity significantly improves the reliability of WSNs if the antenna which experiences the least packet drops is chosen. Antenna diversity can be utilized if the sensor nodes are large enough so that at least two antennas can be fitted or an external antenna attached to the node and can be easily implemented on any commercial radio simply by applying a RF switch.

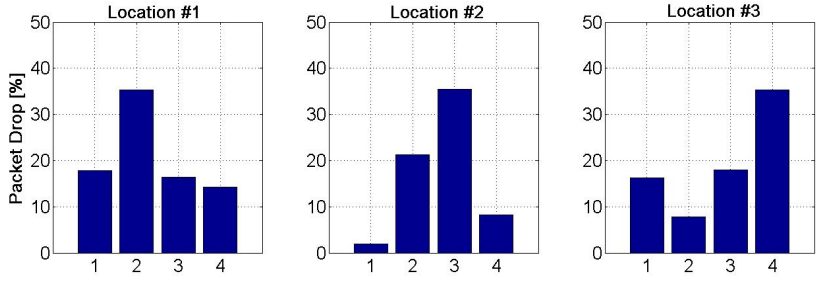


Fig. 1. Measurement data from a field test at an industrial warehouse. Indexes on x-axis represent individual antennas.

2.4 Summary

In this section we discussed several physical layer solutions which impact on reliability in wireless communication systems. First of all, the chosen bandwidth should be large enough such that narrow band interference does not deteriorate the performance significantly. Moreover, spread spectrum techniques enable low transmission powers and simultaneous multi-user spectrum access on the same frequency band. We also showed measurement results from industrial environments which imply that antenna diversity should be exploited in WSNs to guarantee sufficient packet delivery ratios regardless of the receiver's location.

3. MAC Protocols for Guaranteed Access

The main objective of the Medium Access Control (MAC) layer is to enable collision-free transmissions in an efficient manner. During the development of WSNs, research efforts in

the field of access mechanisms for single-channel wireless sensor networks have been extensive. However, the performance of WSNs could be improved by exploiting multiple frequency channels simultaneously to ensure robustness, minimize delay and/or enhance throughput.. Naturally, special characteristics of WSNs have to be taken into account while designing suitable MAC protocols such as limited transmissions powers, available energy and hardware abilities. Various WSN applications have distinct requirements for a MAC protocol. For example, real-time applications have strict delay constraints while in some applications it is important to maximize network lifetime. Nevertheless, for all applications it is extremely important to ensure reliable packet delivery which can be enhanced on the MAC layer by providing collision-free transmissions. With these issues in mind it is justifiable to have a generic MAC solution that can be tuned depending on the requirements of a particular application to enable economic success of WSNs instead of designing a new protocol for each emerging application.

In principle, orthogonal data transmissions can be achieved using various traditional methods. First of all, Frequency Division Multiple Access (FDMA) technique distributes data transmissions on different frequency bands which are orthogonally spaced, i.e. bands do not overlap. Moreover, the main purpose of Time Division Multiple Access (TDMA) schemes is to avoid collisions by ensuring that each user has its own time slot when to transmit data. Combination of FDMA and TDMA is used for example in GSM systems to provide orthogonal multi-user access. In case of spread spectrum systems Code Division Multiple Access (CDMA) can be exploited. In CDMA each user has its own orthogonal spreading function to provide efficient packet reception at the receiver. Third generation mobile phone systems exploit CDMA to enable spectrum access for multiple users simultaneously.

In order to assure proper and effective use of both single- and multi-channel communications, channel ranking is required to find out the most suitable channels for transmissions. In this section we first consider single-channel MAC protocols designed especially for WSNs. Secondly, the most common multi-channel MAC approaches for ad hoc networks will be reviewed. In the end we present our novel multi-channel MAC design along with a new channel ranking algorithm. We show theoretical and simulation results to justify our approaches.

3.1 Single-Channel MAC Solutions

Since present WSN implementations are able to utilize only one carrier frequency at a time, most of research work has concentrated on single-channel systems. In consequence, innumerable single-channel MAC protocols have been proposed for WSNs exclusively. We direct an interested reader to see (Bachir et al., 2010) for a comprehensive literature review on the topic. Usually single-channel MAC protocols are divided into the following classes based on the operation characteristics. Scheduled MAC protocols utilize TDMA on a single frequency whereas contention-based MAC algorithms do not reserve resources in advance. In addition, hybrid MAC schemes aim to exploit the benefits of both approaches to optimize the performance.

Scheduled algorithms divide time into multiple time slots such that only a single transmission can take place in a collision domain. The strength of this kind of approach is that in case of stable channel conditions, fixed network topology and periodic packet arrivals, transmissions can be scheduled in an optimized manner and no overhead is induced due to resource negotiations. Ideally scheduled systems do not suffer from collisions and can guarantee fixed delays, however, such systems require precise time synchronization which complicates system design. In general scheduled MAC protocols perform well under high traffic loads while suffering from network topology changes, irregular generation of packets and inaccurate timing.

Traditional contention-based MAC schemes used in wireless systems are ALOHA and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The basic operation of ALOHA is simple. If a node generates a packet it will try to transmit immediately. In case of a collision the packet is delayed and retransmitted later on. To improve the throughput time can be divided into multiple time slots such that packets can be sent only in the beginning of a time slot. On the other hand, CSMA/CA systems first sense the channel to see whether it is idle or not and then exchange resource request and response messages before the actual data transmission. This kind of message exchange mainly eliminates the *hidden node problem*, which means that several nodes that cannot hear each other transmit simultaneously leading to packet collisions at the receiver, experienced by ALOHA. Although CSMA/CA is widely used in different wireless systems, such as in IEEE 802.11 networks, its performance degrades under high traffic loads.

A hybrid MAC solution is used in IEEE 802.15.4 networks which consists of beacon periods, Contention Access Periods (CAPs) and Contention Free Periods (CFPs). The beacon period is used to distribute general information about the network, frame structure and so forth. During CAP nodes that do not have enough resources can compete for transmission opportunities using CSMA/CA and CFP is reserved for periodic messaging. The frame structure also allows inactive periods if there is nothing to be sent. While a node is idle it can turn its radio off and sleep to minimize energy consumption.

3.2 Multi-Channel MAC Approaches

Due to the challenging nature of radio channels and coexistence of various systems on unlicensed frequency bands, multi-channel communications can be utilized to enhance reliability of wireless networks. Since only a few multi-channel MACs have been designed especially for WSNs, we discuss the main approaches proposed for ad hoc networks in this subsection. In general, existing multi-channel MACs can be divided into four main classes, namely split phase, common hopping, parallel rendezvous and dedicated control channel.

Dedicated control channel schemes (Wu et al., 2000) tune one receiver on the chosen common control channel to avoid the *multi-channel hidden node problem*, which occurs if the channel usage of neighbor nodes is not known and nodes choose to transmit on a busy channel, and use a transceiver to carry out data transmission on different channels. In split phase based random access approaches the operation is divided into two parts. First, during the contention period nodes reserve resources on the chosen common control channel and afterwards, data transmissions will take place during the data period (So & Vaidya, 2004).

On the other hand, the basic idea behind common hopping approaches is to use periodic channel hopping on every channel in order to avoid availability and congestion problems of the common control channel (Tzamaloukas & Garcia-Luna-Aceves, 2000). Furthermore, the fundamental concept of parallel rendezvous approaches (So et al., 2007) is that all the nodes employ individual predetermined hopping patterns. If a node wants to transmit a packet, the node tunes onto the receiver's hopping pattern and the RTS/CTS message exchange and data transmission will be carried out on the receiver's current channel or alternatively by continuing the receiver's hopping pattern, depending on the protocol in question.

Since dedicated control channel schemes require one additional receiver, the approach is not suitable for simple, low-cost WSNs. Performance of different approaches was studied in (Mo et al., 2008) by performing theoretical analysis and simulations with respect to throughput and delay in a single collision domain. Results show that parallel rendezvous approaches outperform common hopping and split phase approaches in a single collision domain. However, parallel rendezvous approaches are unable to neither dynamically adjust to changes in radio environment since the hopping patterns are predetermined nor allow sleeping. The same applies to common hopping approaches as well. The difference in performance of common hopping and parallel rendezvous approaches is due to the fact that after a transmission the channel can be immediately reused in parallel rendezvous approaches while in common hopping approaches the channel cannot be reused until the hopping cycle reaches this particular channel again. The main problem with split phase based schemes is that a fixed part of the frame cycle is reserved for resource negotiations which causes throughput degradation and incurs additional delay. If a packet is generated during a data period, it has to wait at least until the beginning of next data period to be sent. Since delay is of significant importance in various wireless applications, we have designed a novel, delay efficient multi-channel MAC which will be presented next.

3.3 Generic Multi-Channel MAC Protocol

The proposed Generic Multi-channel MAC (G-McMAC) protocol is a hybrid CSMA/TDMA protocol for multi-channel systems which is scalable with respect to packet transmission delays and throughput. In G-McMAC, contention and data periods are merged to minimize delays. G-McMAC is presented in (Nethi et al., 2010) in detail along with a comprehensive set of simulation results and here we only summarize operations of the protocol and show some of the simulation results.

The operation of the protocol is divided into two segments: Beacon Period (BP) and Contention plus Data Period (CDP). Common Control Channel (CCC) can be used for data transmissions if the amount of available channels is otherwise small. If the CCC is used for data transmissions, delay constraints have to be relaxed since in that case secondary contentions can be performed rarely. G-McMAC uses the following messages: Beacons are sent periodically in order to keep time synchronization accuracies under control and routing information up to date, Resource Request (RsREQ) messages are used for making resource requests and Resource Acknowledgment (RsACK) messages are used for responding to the resource requests. Nodes have to sense the desired data channel before data transmissions to avoid the multi-channel hidden node problem. Fig. 2 shows the operation principles of G-McMAC for clarity.

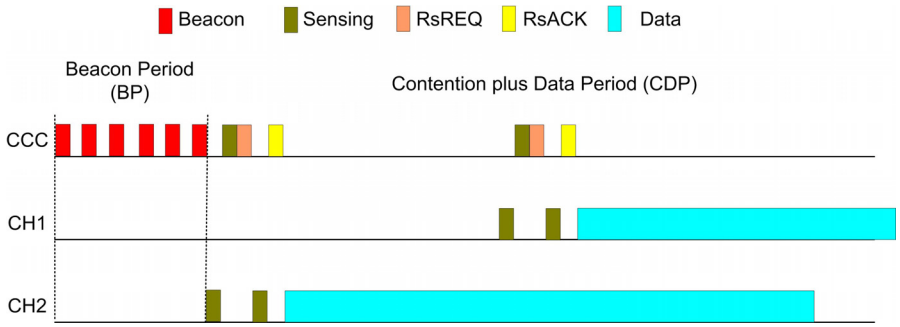


Fig. 2. Demonstration of G-McMAC functionalities.

We implemented G-McMAC on ns-2 (ns-2, 2010) and simulated a real-world industrial warehouse scenario. The scenario considers co-existence of three applications in an industrial environment: Crane Control System (Grey), Machine Health Monitoring System (Red) and Cooling system (Green), as indicated in Fig. 3. Typical communication constraints for Crane Control System (CCS) include a 500ms upper bound for delay and the Gateway (GW) should receive packets from all its sensors within this time limit. Failing to do so results in noticeable delay by the crane operator and the crane will shutdown. If the GW receives a response from all the sensors within 500ms after the polling is initialized, the attempt is considered as successful. In our scenario CCS is the primary network because of the strict delay requirements while Machine Health Monitoring System (MHS) and Cooling System (CS) have lower priority, i.e. they will compete for the rest of the resources. MHS monitors vibrations of the machine structure and in case of MHS, a successful attempt corresponds to MHS gateway node receiving current data sets from all the nodes on the Lathe machine in time. In addition, we also have sensors reporting the measured temperature values to the cooling system. The cooling unit controls temperature in the warehouse through air conditioning system. IEEE 802.15.4 radios are used for wireless communications. Fig. 3 illustrates the scenario.

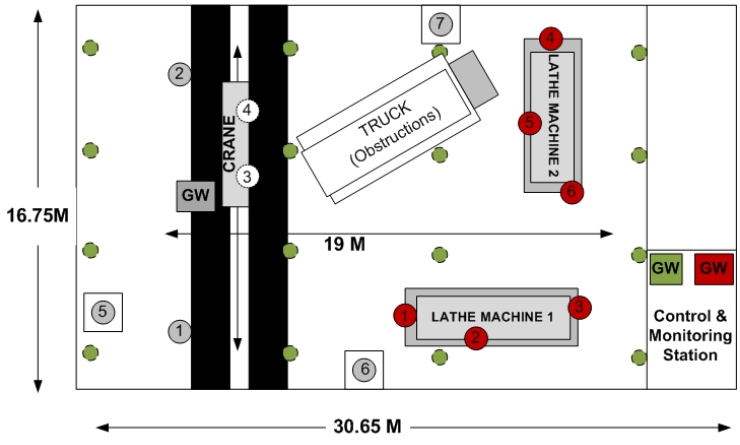


Fig. 3. Demonstration of the simulated industrial warehouse scenario.

The corresponding results for G-McMAC are presented in Fig. 4. CCS maintains high success rate for low channel resources and the performance improves as the number of available channels increases. On the contrary, since MHS is a low priority application, scarcity of channel resources leads to low performance. While the performance of MHS improves as the number of available channels grows, the performance of the cooling system deteriorates since MHS throttles the throughput of the cooling system.

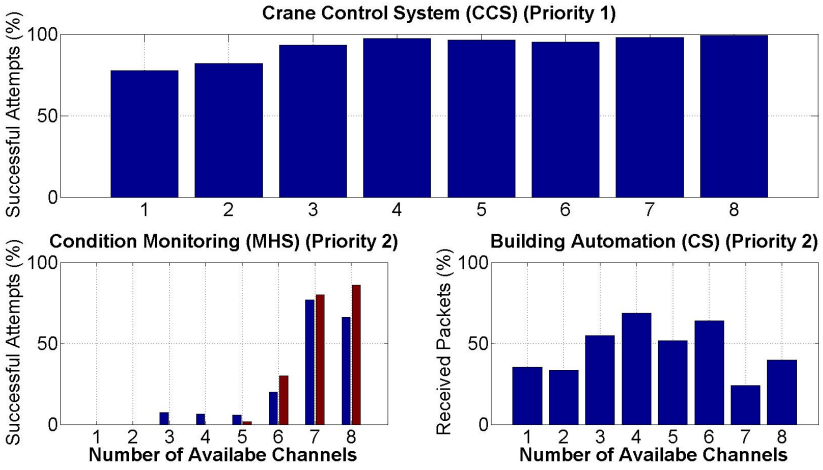
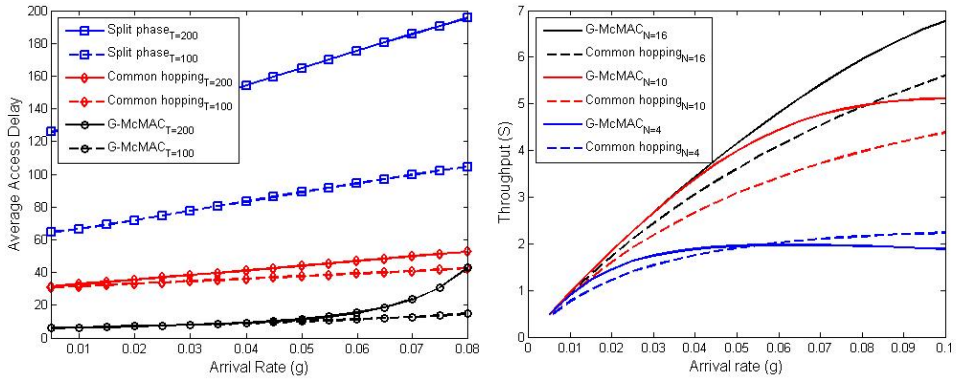


Fig. 4. Simulation Results using G-McMAC.

We have also compared the performance of different multi-channel protocols in case of Poisson arrivals in (Nieminen & Jäntti, 2010). In the paper we studied delay-throughput characteristics of various approaches and derived closed-form equations for different schemes by assuming fixed packet sizes. Time was dividided into small time slots for the analysis and we verified the correctness of theoretical results by simulations using Matlab. Some of the results are depicted in Fig. 5. We denote the number of available channels by N and T is the packet size (in time slots). The results in Fig. 5(a) undoubtedly prove that G-McMAC outperforms other approaches in terms of delay regardless of the number of available channels, packet arrival rate or packet size. In case of Poisson arrivals, the delay of parallel rendezvous approaches is equal to the delay of common hopping approaches. Since the delay of split phase approaches is very high in case of Poisson arrivals, we only compare the throughput of G-McMAC to common hopping approaches in Fig 5(b). As we can see, G-McMAC achieves the highest throughput in many cases. However, in some cases other approaches may offer higher throughput. The performance of the different approaches is discussed in the paper more in depth. Nevertheless, since access delay is the most important parameter for many WSN applications, we conclude that utilization of G-McMAC is feasible in multi-channel WSNs.



(a) Average Delay (b) Throughput

Fig. 5. Performance comparison of different multi-channel approaches.

3.4 Channel Ranking

A sensor network can experience interference in temporal and spatial domains on all the available channels which causes performance degradation. The solutions posed for such situations must efficiently incorporate interference avoidance schemes which are suitable for resource constrained wireless sensor networks (Stabellini & Zander, 2010). For interference avoidance, a single-/multi-channel sensor network must be able to identify the channel(s) offering relatively higher temporal and spatial gaps. This task requires designing the interference characterizing estimator algorithms that can evaluate the impact of temporal occupancy and signal level of a channel and combine the two estimates in a smart way to find an accurate relative channel ranking.

Channel ranking can be performed in an active or passive manner. The active approach, link level interference characterizing model PDR-SINR (Sha et.al., 2009), correlates the PDR with SINR by using the active measurement packets. It is an accurate approach in capturing any link dynamics in the presence of interference, however, it incurs high convergence time and overhead. Moreover, this model is not available during network initialization. A passive scheme to identify the spectrum access opportunities is spectrum sensing. Spectrum sensing allows exploiting the degrees-of-freedom in spatial separation and temporal gap of available channels and achieving orthogonality against the interference (Geirhofer, 2008).

In (Mahmood & Jäntti, 2010), we propose a channel ranking scheme based on spectrum sensing in the presence of WLAN interference. It estimates the interference estimators, activity factor and strength from a receiver centric perspective. Since during network initialization the link qualities are not known, the impact of interference estimators on a sensor location cannot be identified. Therefore, a design of generic consensus is required to weight and combine the two interference estimators according to their impact. Assuming $p(c_i)$ and $P(c_i,s_i)$ the channel occupancy and the signal strength of interference respectively on a channel c_i as perceived by a sensor at location s_i, the interference vector can be written as a function of the interference estimators as

$$\psi(c_i, s_i) = f\left(w_\rho \rho(c_i), w_P P(c_i, s_i)\right) \quad (1)$$

where w_p and w_P are the desired weights of the temporal occupancy and strength level. In order to find the channel ranking based on the influence the two estimators have on the suitability of the channels, a decision theoretic approach (Saaty, 1980) can be used which allows defining the impact of the interference estimators on the fitness of a channel to establish channel ranks. We found that two distinct decision rules for weighting the interference estimates can be derived by using theoretical PDR-SINR performance model. The rules are independent from the PDR-SINR model and a transition boundary governs the transition between the rules depending on the spread of the strength level estimator of the interfered channels. The rules are applicable without loss of generality to any modulation type employed by the sensors which makes the proposed method unique.

The decision rules on weighting the interference estimators are set according to the strength level estimator of the channels. Provided the strength level at a location for candidate channels is less than 1.4 dB, the two interference estimators must be weighted equally to minimize the ranking error. These channels are called as Type-I channels. Otherwise, strength estimator must be weighted 6-7 times more than the activity estimator. We call these channels as type-II channels. The ranking error determined for these channel types with respect to different scaling factors of interference estimators is shown in Fig. 6(a). The trend line shows the average ranking error for each channel type and the vertical bars along each the line indicates the confidence interval of ranking error. The possibility to find a single best channel by assigning these scales is shown in Fig. 6(b) where check mark (√) indicates the best channel is found independent of the weight preference to any estimator otherwise it is crossed (x). The results are based on a real-world measurement campaign performed in the university campus area at Aalto University.

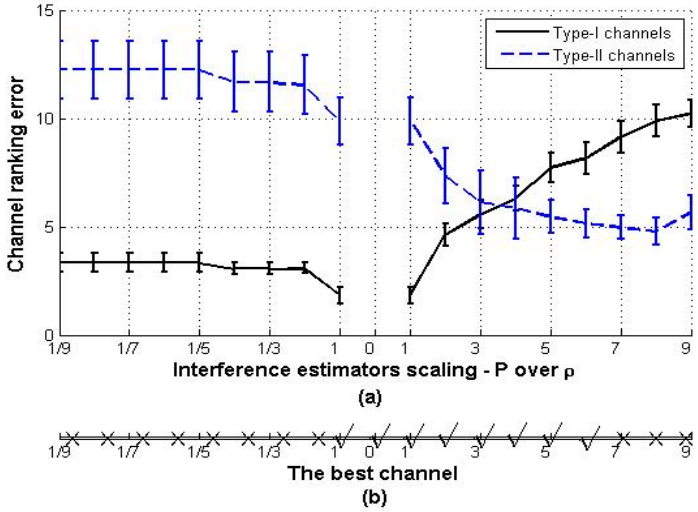


Fig. 6. Channel ranking error for two channel types with respect to the preference scale of interference estimators.

3.5 Summary

In the beginning of this section general aspects of MAC layer design were discussed. Then, we reviewed the most common single- and multi-channel approaches and concluded that for guaranteed medium access, multi-channel communications are required. After this, our proposed multi-channel MAC protocol designed especially for WSNs, named Generic Multi-channel MAC (G-McMAC), was introduced along with theoretical and simulation results to demonstrate the performance. Finally, we considered the importance of channel ranking in WSNs. In this case a novel algorithm was presented along with measurement results.

4. Network Layer and Reliable Routing

Routing in WSNs has specific requirements which means that routing protocols have to take into account such factors as limited bandwidth, variable capacity of radio links and energy-efficiency. Therefore, it is not a trivial task to find a path from one node to a possibly distant destination node if the network topology is dynamic, individual nodes are unreliable and only the nearest neighbors can be reached directly. Since wireless sensor nodes can communicate only with their nearest neighbors because of power limitations, a connection between two nodes often uses several intermediate nodes as relays (multi-hop connection). In general, the main objective of WSN routing protocols is to enable reliable communications between nodes while minimizing power consumption in order to prolong network lifetime. Supporting real-time communications with given delay bounds is also extremely important since some applications need to rapidly respond to sensor inputs. Added to this, practical algorithms should provide robustness against link failures, e.g. by performing multi-path routing, and track changes in the network topology in case of mobile nodes to ensure connectivity.

Routing solutions for other types of networks (e.g. wireline, MANET) cannot be employed directly since they have limitations regarding the WSNs. Nevertheless, due to the importance of routing, the topic has been widely studied and countless protocols have been proposed (Al-Karaki & Kamal, 2004). In this section we will overview some of the proposed solutions for WSNs by focusing on the main routing classes: hierarchical, multipath and flat routing. We also introduce a novel routing protocol which is designed particularly for WSNs. The main benefits of the proposed protocol are that it can be easily implemented on ZigBee and it outperforms the currently used protocol which is shown by simulations.

4.1 Classification of Routing Protocols in WSNs

Hierarchical routing is based on the creation of clusters and the assignment of different tasks to cluster heads and other nodes. Hierarchical approach allows more complicated data processing operations to be carried out by cluster heads. Due to data aggregation and fusion in the cluster heads, the number of transmitted messages in the WSN can be significantly reduced and hence, the energy efficiency increased. As a representative of hierarchical routing methods in WSNs, we consider the Ripple-Zone (RZ) routing scheme (Hu et al., 2005) where sensors are assigned to different ripples based on their distances in number of hops from the actuator. In each ripple, some sensors are chosen as masters based on the Topology Discovery Algorithm (TDA) previously proposed by the authors. Each master

collects data from the sensors in its zone and then transmits data to a master in the next ripple that is closer to the actuator. In the paper, authors show that the protocol is energy efficient, reliable and scalable. Moreover, it can adapt to changing network topology by employing the local link failure repair method. However, the cases where several actuators are interested in the same sensed data and coordination issues among actuators were not taken into account. The performance of the scheme in terms of latency, which is a crucial issue in real-time WSN applications, was neglected in the study as well.

An example of the flat routing approach is the Delay-Energy Aware Routing Protocol (DEAP) (Durresi et al., 2005) which is designed for heterogeneous sensor and actuator networks. The major components of DEAP are loose geographic routing protocol based on Forwarding sets, which in each hop distributes the load among a group of neighbor nodes and the Random Wakeup Scheme (RAW) that controls the wake up cycle of sensors based on experienced packet delay. DEAP combines routing and sensor wake-up schemes and finds a trade-off between transmission delays and energy consumption. It is also capable of adapting to changes of network topology and takes advantage of actor nodes by using their resources when possible. Furthermore, Scalable Source Routing (SSR) proposed in (Fuhrmann, 2005) is a fully self-organizing protocol for efficient routing in large random networks. In the paper, the authors also point out disadvantages of routing schemes based on source routing bridges and shortest path routing (link state or distance vector) and come to the conclusion that these techniques must be avoided to obtain the desired efficiency.

As the name indicates, multipath routing protocols use multiple paths instead of a single path in order to enhance network performance and reliability. Successful delivery of data is ensured by exploiting optional paths if primary paths fail. By transmitting the same packet over several different paths, the probability of successful packet delivery can be increased at the cost of increased energy consumption and traffic overhead (Al-Karaki & Kamal, 2004). Another advantage of multipath routing is load-balancing, where traffic between a source and destination is split across multiple (partially or completely) disjoint paths. Load balancing spreads energy utilization across nodes in a network and this way prolongs its lifetime. Multipath routing is a promising approach for WSNs since high node densities allow utilization of multiple paths with similar costs. Most of the up-to-date multipath routing schemes are either targeted to find a number of disjoint routes or energy efficient routes (Ganesan et al. 2001; Li & Cuthbert, 2004; Popa et al., 2006). In these schemes, load is either distributed or sent on the best (e.g., most energy-efficient, best in QoS, etc.) path. In the first case, i.e. distributing load over multiple paths, the destination node has to cope with synchronization of arrival packets. Choosing the best path could avoid synchronization issues but the process easily drains out batteries of the participating nodes because the source node continuously uses the particular path until the link breaks. Because of these issues we propose a novel routing algorithm which gives the source and intermediate nodes freedom to choose from multiple local paths to the destination based on a cost function.

4.2 Localized Multiple Next-hop Routing (LMNR) protocol

The design of the ZigBee routing scheme is based on the Ad-hoc On-demand Distance Vector (AODV) (Perkins & Royer, 2001). AODV is an on-demand routing algorithm, meaning that the routes are established only when there is information to be sent and

maintained as long as they are needed for communication. Route freshness is ensured by using sequence numbers. AODV is loop-free, self-starting, and scalable. In AODV, if a source node does not have information about a destination node in its routing table, it initiates the Route Discovery procedure. The procedure starts by broadcasting a Route Request (RREQ) packet to the neighbor nodes. The RREQ automatically sets up a reverse path to the source from all intermediate nodes lying on the path from the source to the destination. The destination node sends a Route Reply (RREP) after receiving the first RREQ. Each intermediate node forwards the RREP to its preceder until the RREP arrives at the source node. Meanwhile, each node (including the source node) having received the RREP establishes a route entry in its route table.

In Localized Multiple Next-hop Routing (LMNR) (Nethi et al., 2007c) we classify all the paths between a source-destination pair into two types: I) node disjoint paths and II) local paths. Instead of sending packets parallel using solely disjoint paths, the used paths can be selected locally. The novelty is that the source and intermediate nodes are given freedom to choose from multiple local paths based on a cost function. This will reduce delay and routing overhead which improves the network performance. HELLO messages of AODV are used to update the cost of each individual node. Since LMNR uses existing information in AODV and does not require any change in routing packets, the protocol is able to co-exist with AODV and easy to implement on ZigBee based systems. Our algorithm also adapts to topology changes by monitoring the activity of the neighbors. If the next hop on the path is unreachable, an unsolicited RREP with a new sequence number is propagated through the upstream of the break. Moreover, if the source node still requires a route to the destination, it can restart the discovery procedure. Since AODV restricts intermediate nodes to have a single route to the destination, link stability becomes a problem. Consequently, the delivery performance is degraded and reliability is compromised. We modify the route discovery process to incorporate multiple routes such that when a node receives another copy of RREQ from the same source, it will check the routing table as follows

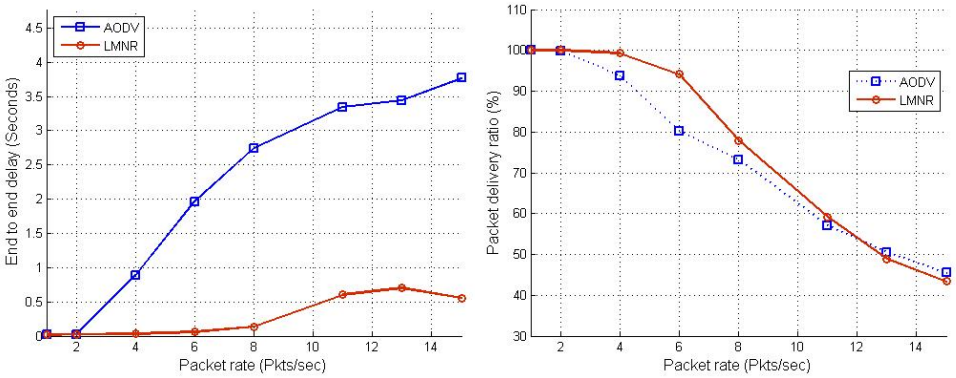
1. If the new RREQ has a smaller hop count (i.e., shorter distance to the source node), it updates the route entry as original AODV does.
2. If it equals to the one(s) in route table, the node simply adds a new route (multipath to source).

By this mechanism, alternate (and equal hop count) paths at each intermediate nodes for one source-destination communication pair will be found. Furthermore, dynamic adjustment should be considered so that the intermediate nodes either shall not drain out all their energy or alleviate and balance the routing load. For this purpose we modify the AODV neighbor table, and introduce a new metric *Node Cost* (*NC*), which is put into the neighbor table. Actually the node cost function can be chosen from the following metrics (or a combination of them): outgoing queue buffer occupation ratio, congestion measurement which is proportional to the MAC layer contention (backoff) window size, measure of routing table size and freshness of route entries and/or packet leaving rate at the network layer outgoing queue. For more detailed information about the operations see (Nethi et al., 2007c).

With the knowledge of the routes each intermediate node can now avoid using (next-hop) nodes which have higher cost function, without increasing the number of hops to the destination. However, it is possible that for a given intermediate node all of its next-hop nodes may have very high cost. To cope with this problem, a back-propagation mechanism is introduced. The back-propagation logic can be described as follows. If a node sees that all its next hop nodes' costs are greater than the given threshold, the node will back propagate this update to its preceder so that the preceder is able to give up using this path. Once the RREQ-RREP procedure is completed, the source-destination pair and intermediate nodes involved will select a single path amongst all the available (local) paths.

4.3 Simulation Results

We implemented LMNR on ns-2 (ns-2, 2010) and carried out simulations to see how much gain LMNR achieves compared to AODV in practice. In the simulation scenario 50 nodes, which use IEEE 802.11 radios for communications, were randomly positioned on a grid. 10 source-destination pairs are randomly selected and each source generates Constant Bit Rate (CBR) traffic flows with the given packet rate (packets/second). The used NC metric was based on the size of routing tables and freshness of routes. Simulation setup is explained in (Nethi et al., 2007c) in detail and some of the results are depicted in Fig. 7. Fig. 7(a) compares the performance of the protocols with respect to end to end delay and as we can see, our scheme outperforms AODV clearly as traffic loads increase. The reason behind this is that LMNR can always find an optimal path due to the dynamic local next-hop selection mechanism. On the contrary, in AODV only one route is established which means that a new route-finding procedure is initiated in case of congestion. This can be also verified by Fig. 7(b), which shows the packet delivery ratios of the two routing protocols. LMNR is better than AODV at medium traffic loads whereas the performance is similar with low and high traffic volumes. This is because of the fact that LNMR tries to find a better next-hop path instead of initiating a Route Error (RERR) as AODV does. As traffic load increases, the entire network becomes saturated and hence, the performance of both protocols decreases.



(a) End to end delay (b) Packet delivery ratio

Fig. 7. Performance comparison between LMNR and AODV.

The simulation results show that LMNR outperforms AODV in terms of end to end delay. Furthermore, the results also indicate that the link failure resilience of LMNR is higher compared to the conventional AODV routing protocol since less packet drops are experienced with moderate traffic loads. LMNR requires only minor modifications on AODV and thus, the proposed protocol can be used, for example, in legacy ZigBee systems.

4.4 Summary

In this section, we focused on network layer operations and considered the main problems related to routing in WSNs. We categorized routing approaches into three cateories: hierarchical, multipath and flat routing. Pros and cons of each approach were analyzed and an example algorithm was given for each class. We drew a conclusion that the use of multipath routing is feasible in WSNs because of high node densities due to which there exists many paths with similar cost . Multipath routing enables transmission of multiple packet copies over multiple paths and load-balancing. Finally, we presented a novel routing algorithm which can be easily implemented on ZigBee, called Localized Multiple Next-hop Routing (LMNR), and demonstrated the achievable benefits by simulations.

5. Performance of Various Applications with Communication Co-Simulation

In addition to the theoretical results, co-simulation of the communication and application is important and necessary for several reasons. Simulations are a feasible way to test and evaluate wireless applications, such as sensor networks, distributed data processing algorithms, and wireless control systems. With simulations, the critical properties and behaviour of the network, and the impact on the application can be analyzed. Problems occurring in the network and the reaction and resulting performance of the algorithms to these issues can be studied. These issues, in particular the protocol specific ones, are hard to be approached analytically. Especially the study of wireless networked control systems (WiNCSs) benefit from co-simulation, where the real-time requirement of control is affected by the unreliability of wireless communication.

Simulation of wireless applications with a specific network protocol is thus needed. Therefore, the network and control co-simulator PiccSIM (Nethi et al., 2007a) has been developed. PiccSIM is aimed at communication and control co-simulation, especially for the study of WiNCSs. In PiccSIM, specific network protocols and control algorithms can be studied. The strength of PiccSIM is to enable one to quickly test several control algorithms in realistic WiNCS scenarios. In the following sections PiccSIM is described in more detail and some simulation cases are presented that show the benefits of co-simulation for WiNCSs design. The simulation cases involve multiple networked control loops, which cannot be studied without co-simulation.

5.1 PiccSIM

PiccSIM integrates two simulators to achieve an accurate and versatile simulation system at both the communication and control level for WiNCSs. PiccSIM stands for *Platform for integrated communications and control design, simulation, implementation and modeling*. It has the unique feature of delivering a whole chain of tools for network and control modeling and

design, integrated into one package with communication and control co-simulation capabilities. The PiccSIM simulator is an integration of Matlab/Simulink where the dynamic system is simulated, including the control system, and ns-2, where the network simulation is done. The PiccSIM Toolchain is a graphical user interface for network and control design, realized in Matlab. It is a front-end for the PiccSIM simulator and delivers the user access to all the PiccSIM modeling, simulation and implementation tools (Kohtamäki et al., 2009).

There are already some suitable simulators for WiNCSs, such as TrueTime (Cervin et al, 2003) and Modelica – ns-2 (Al-Hammouri et al., 2007). Modelica/ns-2 is a very similar platform to PiccSIM. As in PiccSIM, the network simulation is done in ns-2, but the plant dynamics and the control simulation are done in Modelica. The simulation is controlled by ns-2 and the traffic is defined beforehand, so event-driven communication is not possible, contrary to PiccSIM where Simulink controls the communication based on the outcome of the dynamic simulation model. Perhaps the most well-known Simulink network blockset is TrueTime, which is actively developed at the Lund University, Sweden. It supports many network types (Wired: Ethernet, CAN, TDMA, FDMA, Round Robin, and switched Ethernet, and wireless networks: 802.11b WLAN and IEEE 802.15.4) and it is widely used to simulate wireless NCSs (Andersson et al., 2005). Besides the dynamic system simulation offered by Simulink, network node simulation includes simulation of real-time kernels. The user can write Matlab m-file functions that are scheduled and executed on a simulated CPU.

Two wireless node operating system simulators, TOSSIM (Levis et al., 2003) and COOJA (Österling et al., 2006), are worth mentioning. Both are sensor node operating system simulators, which simulate the code execution on the wireless nodes. They have simple range-based network propagation models to allow simulation of many nodes communicating with each other. They do not specifically support control system simulation, but complete wireless applications can be simulated with these tools, including input/output for sensing and actuation.

5.2 PiccSIM Architecture

The PiccSIM simulator consists basically of two computers on a local area network (LAN): the Simulink computer for system simulation, including plant dynamics, signal processing and control algorithms, and the ns-2 computer for network simulation. For further details see (Nethi et al., 2007a), where the integration of ns-2 and Simulink is reported, and (Kohtamäki et al., 2009) for the description of the PiccSIM Toolchain. The network is simulated in PiccSIM by the ns-2 computer. Packets sent over the simulated network are routed through the ns-2 computer, which simulates the network in ns-2 according to any TCL script specification generated automatically by a network configuration tool based on the user-defined settings. Simulation time-synchronization is performed between the computers.

Since PiccSIM is an integration of two simulators, they are by definition separated. To close the gap between the simulators, a data exchange mechanism is implemented, which can pass information from one simulator to the other. This enables the simulation of cross-layer protocols that take advantage of information from the other application layers. An example where the data exchange mechanism can be used is with mobile scenarios. Ns-2 supports

node mobility, but natively only with predetermined or random movement. There exist, however, many applications, such as search-and-rescue, exploration, tracking and control, or collaborating robots, where the control system or application determines the node movement in run-time. In these cases the controlled node positions must be updated from the dynamic simulation to the network simulator. The updated node positions are then used in the network simulation, and they affect, for instance, the received signal strength at the nodes. Moving nodes will eventually cause changes in the network topology, which requires re-routing.

5.3 Simulation cases

With PiccSIM, simulation of systems involving many interacting wireless protocols and algorithms, for example multiple control loops, can be studied. The intricate interaction between the network, such as routing and traffic pattern, and the control system, including mobility, can only be assessed by simulation. The application generated traffic and network performance affect the outage lengths, packet drops, and delays, which affect the whole application in some particular way. The capabilities of the PiccSIM simulator are demonstrated here in three different scenarios to show how the application performance can be assessed with co-simulation.

The first case is a building automation application where the temperature and ventilation of an office is controlled using wireless measurements. This case focuses on the throughput, packet drops, and structure of the network. The second case is a robot squad, which moves in various formations. This case is more demanding for the wireless network, as the formation changes alter the topology of the network and re-routing must be done continuously to maintain the communication between the robots. These example cases have previously been presented in (Nethi et al., 2007b), and (Pohjola et al, 2009). It is notable that the performance of these control systems cannot be determined analytically beforehand.

An office with wireless control of the heating, ventilation and air conditioning is simulated. The layout of the office is shown in Fig. 8 with a total of 39 rooms. The temperature and CO_2 of the office rooms, which depend on the occupancy of the room, are modeled using first principles (Nethi et al., 2007b). The network is a wireless IEEE 802.15.4 network using the AODV routing protocol. Wireless sensors in each room measure the temperature and CO_2 concentration and additionally presence event messages are sent to the central command when people enter or exit a room. The central control system coordinates the heating and ventilation of the individual rooms based on the wirelessly communicated measurements. The local heating/cooling and ventilation commands are transmitted back to the rooms. The wireless network deals with both time and event-triggered messaging. Because of the quantity of nodes, multiple hops, radio environment, and random access MAC, there are packet drops, which impair the control result.

The temperature variation in each room depends on the movement of people in and out of the room and the compensation done by the control system. The case is simulated and compared to the control performance with perfect communication. Generally, the fewer measurements are dropped by the network the better the control result is. Fig. 8 shows the increase of the maximum deviation from the desired temperature when using the wireless

network for delivering the measurements. The results with one access-point are not satisfactory, so another access-point is added near room number 19. The access-points are connected with a high-speed backbone network. With two access points the communication quality is so good that no difference in the control performance from the case with a wired system is discernible. Thus, by designing the network to be reliable enough, the control application works equally well to perfect communication.

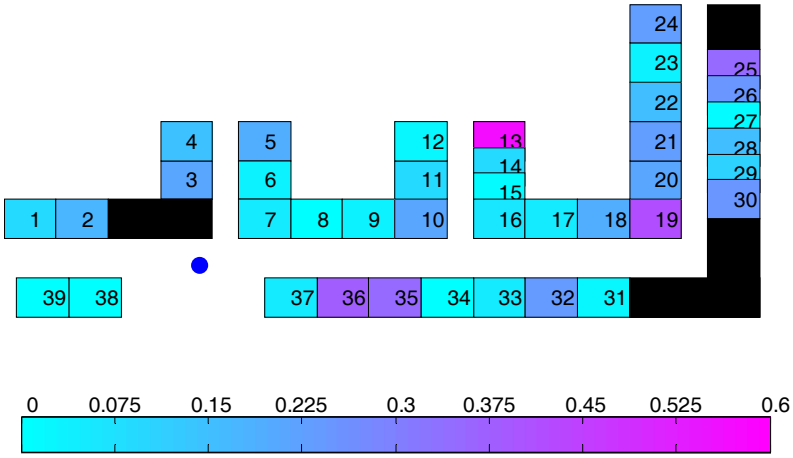


Fig. 8. Increase in maximum temperature error for wireless temperature control with one access point (blue dot) compared to perfect communication.

The second scenario considers a target tracking and control case with grid of nodes forming a static sensor network and a mobile wireless robot. The sensor network serves as an infrastructure network for transmitting measurement and control signals from/to the mobile node and providing a localization service. The objective for a centralized controller located at an edge of the infrastructure grid, is to control the mobile node according to a predefined track. On the control side a Kalman filter is used for filtering the mobile node position and predicting the position if the information is not available, due to packet drops. A PID controller is then used to control the mobile node. The control signal is routed to the mobile robot, which applies the acceleration command.

Nearby infrastructure nodes can measure their distance to the mobile node, for example by using ultrasound. The distances are transmitted to the controller. Using at least three distance measurements, the controller can determine the position of the mobile node by triangulation. By simulation it is noted that the requirement to receive three measurements from the same sampling interval is not always fulfilled. Hence the controller has to use data from older sampling instants for which more measurements have arrived, which causes trouble to the control application. A comparison between a singlepath routing protocol, specifically AODV and the LMNR multipath routing protocol is done in simulations. The simulation results listed in Table 1 show that the multipath routing protocol has better communication and control performance measures. The control performance is evaluated by

the integral of squared error (ISE) between the robot desired and actual position. This simulation shows that multipath is advantageous in some mobile scenarios, since at a link break it can quickly switch to a backup route (a counter-example is given next). Moreover, by combining these results (IEEE 802.15.4) with the results in Section 4.3 (IEEE 802.11 radios) we infer that LMNR performs well regardless of the used radio technology.

	Average delay [s]	Routing overhead [%]	Packet loss [%]	Control cost (ISE)
AODV	0.08	8.1	23	18
LMNR	0.001	0.5	10	8.6

Table 1. Network and control performance metrics from the target tracking case

The third scenario is similar to the previous case and considers a squad of mobile wireless robots moving in various formations. A possible application is a search and rescue or exploration scenario. A leader robot controls the positions of the other robots. The assumption is that the robots can localize themselves based on GPS, odometer or inertia measurements. The robots transmit their positions to the leader robot. The leader then calculates the control signals for the locomotion, taking into account collisions and the final formation, and transmits, at every sampling time, the control message to the other moving robots. The communication is done over an IEEE 802.15.4 radio with a maximum communication range of 15 m. The communication conditions are modeled in ns-2 with Ricean fading, which results in individual packet losses because of fading links. Furthermore, the links may break due to mobility as well.

In this scenario, the speeds of the control system dynamics and the network are of the same magnitude. This means that the network delays are significant for the control system performance. Both the network and the control system need to be simulated at the same time to get accurate results of the whole networked system. As the robots change formation, the communication links might break, and a new route must be established. The speed at which the path is re-established depends on the routing protocol. The network performance, and ultimately the control performance, depends on the formation of the robots and how the packets are routed through the network. The communication outages naturally degrade the control performance. More generally, instead of mobility, the outages can be caused by a changing environment, such as moving machinery in a factory.

Simulations of three formation changes of a squad of 25 robots are done (Pohjola et al., 2009). The differences between using the AODV and LMNR routing protocols are evaluated. The results are compared to the case without network, i.e., control with perfect communication, and with no mobility, i.e. no topology changes. Some network and control results are in Table 2. The control cost is significantly higher than for the case without a network, and slightly higher with a network but without mobility. Thus, the network has a considerable impact on the control system. According to the performance metrics, singlepath routing has, contrary to the previous case, an advantage over multipath. This advantage is because in the high mobility case, there are more link breaks when using multipath routing, which generate more routing overhead.

	Average delay [s]	Routing overhead [%]	Packet loss [%]	Control cost (ISE)
No network				0.1
No mobility	0.009	0.8	0.1	2.3
AODV	0.015	3.2	30	2.7
LMNR	0.09	11.2	20	3.3

Table 2. Network and control performance metrics from the robot squad case

5.4 Summary

The communication and control co-simulator PiccSIM was introduced. With PiccSIM, wireless applications can be simulated and studied. The application performance, which partly depends on the network design, can be measured. The presented simulation cases show the benefit of communication and control co-simulation of WiNCS. With simulation, the effect of the network on the application and the resulting performance can be assessed. The optimal network design depends on the application and is determined by the specific application operation and needs. This guides the protocol design to improve the essential network problems experienced by the application. More efficient design is obtained as the issues affecting the application the most can be identified and improved

6. Conclusions

Rapid development of small, low-cost sensors has opened the way for implementation of wireless sensor network technology in countless applications. Although research has been comprehensive in various important fields in the context of WSNs, such as energy efficiency and security, reliability of the underlying communication system has received less attention. Hence, in this chapter we considered robustness of existing protocols and discussed advanced communication solutions for reliable wireless sensor systems by considering physical, medium access and network layers. On the physical layer antenna diversity should be exploited to further enhance WSNs resiliency. Collision-free medium access enables reliable delivery of packets and by using efficient channel ranking algorithms and multi-channel communications the performance of the system can be improved, especially under interference. Furthermore, multipath routing provides several trails between transmitters and receivers with similar costs which can be utilized to ensure trustworthy communications in systems where links are relatively stable. Finally, we introduced the network and control co-simulator PiccSIM and studied the performance of some real-world applications by simulations.

7. References

Akyildiz, I. F.; Su, W. & Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, Vol. 38, No. 4, March 2002, pp. 393-422.

Akyildiz, I. F.; Pompili, D. & Melodia, T. (2005). Underwater Acoustic Sensor Networks: Research Challenges. *Computer Networks*, Vol. 3, February 2005, pp. 257-279.

Al-Hammouri, A.T.; Liberatore, V.; Al-Omari, H.; Al-Qudah, Z.; Branicky M.S. & Agrawal D. (2007). A Co-Simulation Platform for Actuator Networks. *ACM Conference on Embedded Networked Sensor Systems*, pp. 383-384, Sydney, Australia, November 2007.

Al-Karaki, J. N. & Kamal, A. E. (2004). Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, vol. 11, issue 6, pp. 6-28, December 2004.

Andersson, M.; Henriksson, D.; Cervin A. & Årzén K.-E. (2005). Simulation of Wireless Networked Control Systems, *44th IEEE Conference on Decision and Control and European Control Conference*, pp. 476-481, Seville, Spain, December 2005.

Bachir, A.; Dohler, M.; Watteyne, T. & Leung, K. K. (2010). MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, Vol. 12, No. 2, April 2010, pp. 222-248.

Cervin, A.; Henriksson, D.; Lincoln, B.; Eker J. & Årzén, K.-E. (2003). How Does Control Timing Affect Performance?. *IEEE Control Systems Magazine*, vol. 23, No. 3, June 2003, pp. 16- 30.

Durresi, A.; Paruchuri, V. & Barolli, L. (2005). Delay-Energy Aware Routing Protocol for Sensor and Actor Networks. *11th International Conference on Parallel and Distributed Systems*, pp. 292-298, Fukuoka, Japan, July 2005.

Fuhrmann, T. (2005). Scalable Routing for Networked Sensors and Actuators. *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 240-251, Santa Clara, CA, USA, September 2005.

Ganesan, D.; Govindan, R.; Shenker, S. & Estrin, D. (2001). Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. *ACM SIGMOBILE Mobile Computer Communications Review*, vol. 5, no. 4, pp. 11–25, October 2001.

Geirhofer S.; Tong L. & Sadler B. M. (2008). Cognitive Medium Access: Constraining Interference Based on Experimental Models. *IEEE Journal on Selected Areas in Communications*, Vol. 26, No. 1, pp. 95-105, January 2008.

Gungor, V. C. & Hancke, G. P. (2009). Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 10, October 2009, pp. 4258-4265.

Hu, F.; Cao, X.; Kumar, S. & Sankar, K. (2005). Trustworthiness in Wireless Sensor and Actuator Networks: Towards Low-Complexity Reliability and Security. *IEEE Global Telecommunications Conference*, pp. 1696-1700, St. Louis, MO, USA, November 2005.

IEEE 802.11. (2007). IEEE Standard for Information Technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pp.C1-1184, June 2007.

IEEE 802.15.1. (2005). IEEE Standard for Information Technology - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs), IEEE Std 802.15.1-2005, pp. 1-580, 2005.

IEEE 802.15.4. (2006). IEEE Standard for Information Technology - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 1-305, 2006.

IEEE 802.15.4a. (2007). IEEE Standard for Information Technology - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 1-305, 2006.

Kohtamäki, T.; Pohjola, M.; Brand, J. & Eriksson, L. M. (2009). PiccSIM Toolchain – Design, Simulation and Automatic Implementation of Wireless Networked Control Systems. *IEEE International Conference on Networking, Sensing and Control*, pp. 49-54, Okayama, Japan, March 2009.

Levis, P.; Lee, N.; Welsh, M. & Culler D. (2003). TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. *ACM Conference on Embedded Networked Sensor Systems*, pp. 126-137, Los Angeles, CA, USA, November 2003.

Li, X. & Cuthbert, L. (2004). On-demand Node-Disjoint Multipath Routing in Wireless Ad Hoc Networks. *29th Annual IEEE International Conference on Local Computer Networks*, pp. 419-420, Tampa, FL, USA, November 2004.

Mahmood A. & Jäntti R. (2010). A Decision Theoretic Approach for Channel Ranking in Crowded Unlicensed Bands. Submitted.

Mo, J.; So, H.-S.W. & Walrand, J. (2008). Comparison of Multichannel MAC Protocols. *IEEE Transactions on Mobile Computing* , Vol.7, No.1, January 2008, pp.50-65.

Nethi, S.; Pohjola, M.; Eriksson, L. & Jäntti, R. (2007a). Platform for Emulating Networked Control Systems in Laboratory Environments. *8th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1-8, Helsinki, Finland, June 2007.

Nethi S.; Pohjola, M.; Eriksson, L. & Jäntti, R. (2007b). Simulation Case Studies of Wireless Networked Control Systems. *10th ACM/IEEE International Symposium on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 100-104, Crete, Greece, October 2007.

Nethi, S.; Gao, C.; Jäntti, R. & Pohjola, M. (2007c). Localized Multiple Next-hop Routing Protocol. *7th International Conference on ITS Telecommunication*, pp. 1-5, Sophia Antipolis, France, June 2007.

Nethi, S.; Nieminen, J. & Jäntti, R. (2010). Exploitation of Multi-Channel Communications in Industrial Wireless Sensor Applications: Avoiding Interference and Enabling Coexistence. Submitted.

Nieminen, J. & Jäntti, R. (2010). Delay-Throughput Analysis of Multi-Channel Media Access Control Protocols in Ad Hoc Networks. Submitted.

ns-2. (2010). The Network Simulator, online: http://www.isi.edu/nsnam/ns/ and http://nsnam.isi.edu/nsnam/index.php/Main_Page. Referenced 13.8.2010.

Perkins, C. E. & Royer, E. M. (2001). Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Internet Draft, RFC 3561, July 2003.

Pohjola, M.; Nethi, S. & Jäntti, R. (2009). Wireless Control of a Multihop Mobile Robot Squad. *IEEE Wireless Communications*, *Special Issue on Wireless Communications in Networked Robotics*, Vol. 16, No. 1, February 2009, pp. 14-20.

Polepalli, B.; Xie, W.; Thangaraja, D.; Goyalm, M.; Hosseini, H. & Bashir, Y. (2009). Impact of IEEE 802.11n Operation on IEEE 802.15.4 Performance. *International Conference on Advanced Information Networking and Applications Workshops*, pp. 328-333, Bradford, United Kingdom, May 2009.

Popa, L.; Raiciu, C.; Stoica, I. & Rosenblum, D. S. (2006). Reducing Congestion Effects in Wireless Networks by Multipath Routing, *14th IEEE International Conference on Network Protocols*, pp. 96-105, Santa Barbara, California, USA, November 2006.

Römer, K. & Mattern, F. (2004). The Design Space of Wireless Sensor Networks. *IEEE Wireless Communications,* Vol. 11, No. 6, December 2004, pp. 54-61.

Saaty T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting Resource Allocation*, McGraw Hill International Book Co., ISBN 0070543712, New York.

Sankarasubramaniam Y.; Akyildiz I. F. & McLaughlin S. W. (2003). Energy Efficiency Based Packet Size Optimization in Wireless Sensor Networks. *IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 1-8, Anchorage, AK, USA, May 2003.

Sha, M.; Xing, G.; Zhou, G.; Liu, S. & Wang X. (2009). C-MAC: Model-driven Concurrent Medium Access Control for Wireless Sensor Networks. *28th IEEE Conference on Computer Communications*, pp. 1845-1853, Rio de Janeiro, Brazil, April 2009

Shin, S. Y.; Choi, S.; Park, H. S. & Kwon, W. H. (2007). Packet Error Rate Analysis of IEEE 802.15.4 under Saturated IEEE 802.11b Network Interference. *IEICE Transactions on Communications*, Vol. E90-B, No. 10, October 2007, pp. 2961-2963

Shokrollahi A. (2006). Raptor Codes. *IEEE/ACM Transactions on Networking, Special Issue on Networking and Information Theory*, vol. 52, no. 6, June 2006, pp. 2551-2567.

Shuaib, K.; Boulmal, M.; Sallabi, F. & Lakas, A. (2006). Co-existence of ZigBee and WLAN - a Performance Study. *Wireless Telecommunications Symposium*, pp. 1-6, Pomana, CA, USA, April 2006.

So, H.-S. W.; Walrand, J. & Mo, J. (2007). McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol. *IEEE Wireless Communications and Networking Conference*, pp. 224-339, Hong Kong, Hong Kong, March 2007.

So, J. & Vaidya, N. (2004). Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. *5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 222-233, Tokyo, Japan, May 2004.

Stabellini L. & Zander J. (2010). Energy Efficient Detection of Intermittent Interference in Wireless Sensor Networks. *International Journal on Sensor Networks*, Vol. 8 No. 1, 2010, pp. 27-40.

Tzamaloukas, A. & Garcia-Luna-Aceves, J.J. (2000). Channel-hopping Multiple Access. *IEEE International Conference on Communications*, pp. 415-419, New Orleans, LA, USA, June 2000.

Vuran, M. C. & Akyildiz I. F. (2009). Error Control in Wireless Sensor Networks: A Cross Layer Analysis. *IEEE/ACM Transactions on Networking*, Vol. 17, No. 4, August 2009, pp. 1186-1199.

Wu, S.-L.; Lin, Y.; Tseng, Y.-C. & Sheu, J.-P. (2000). A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Mobile Ad Hoc Networks. *International Symposium on Parallel Architectures, Algorithms and Networks,* pp. 232-237, North Dallas, TX, USA, December 2000.

Yick, J.; Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, Vol. 52, No. 12, April 2008, pp. 2292-2330.

Österlind, F.; Dunkels, A.; Eriksson, J.; Finne, N. & Voigt, T. (2006). Cross-level Sensor Network Simulation with COOJA. *31st IEEE International Conference on Local Computers*, pp. 641-648, Tampa, Florida, USA, November 2006.

2

Factors that may influence the performance of wireless sensor networks

Majdi Mansouri*, Ahmad Sardouk, Leila Merghem-Boulahia,
Dominique Gaiti and Hichem Snoussi*
*ICD/LM2S, ICD/ERA, UMR 6279, Troyes University of Technology
France

Rana Rahim-Amoud
LaSTRe, Université Libanaise
Lebanon

Cédric Richard
Laboratoire FIZEAU UMR CNRS 6525 Université de Nice Sophia-Antipolis
France

1. Introduction

The Wireless Sensor Networks (WSNs) are penetrating more and more our daily life. They are used in a large type of applications as supervision, tracking and control in military, environmental, medical and several other domains. Therefore, new approaches and protocols are proposed every day in order to optimise the performance of the WSNs and to increase their reliability and quality of service. These new protocols take into consideration the challenges of the WSN and they are built up some key factors (parameters and concepts) to achieve their goals.

The aim of this chapter is to study the factors that may influence the desired performance of the WSNs. These factors are inspired from the sensor nodes characteristics, the physical deployment of the WSNs and the WSNs' information functions. Firstly the sensor nodes are characterized particularly by their limited power and memory capacities. The power is used to be a key parameter for any approach supposing that sensor nodes' batteries are unchangeable and not rechargeable. The power would influence the reliability of the network, if the residual battery of an important node, as a cluster head, is limited. Respecting the residual battery of the node leads to a more efficient routing path, cluster head designation, aggregation point selection, etc. The limited memory is also a very important parameter as it defines the size of the operating system and the processing code. It also defines the amount of information that a node is able to store. For example, this parameter has to be managed in a mobile sink Chatzigiannakis et al. (2008) Cheng et al. (2009) approach, to define the duration that a node may tolerate before communicating its information to the sink and to optimize the mobile sink trajectory.

Secondly, the physical deployment of the WSN has to be studied to satisfy the application requirements. The network deployment identifies the density of the network. A random deployment may lead to different density levels in the network. Thus, the redundancy level will not be the same in the entire network, also a sleeping decision of a node in a dense zone will not have the same influence as in a sparse zone. The density could also be correlated with the sensing coverage of the nodes and the global covered area. The lower is the sensing coverage, the higher is its precision level. For example, in a bordure supervision application, the sensing coverage and the density should be combined to minimize the probability of having vulnerable zones. The random deployment leads also to a different nodes position. Thus, a node connecting two parts of the network has to be always activated to minimize, for example, the end-to-end communication delay or to insure a higher connectivity of the network. The position of a node within the WSN may optimise the definition of its role (aggregator, normal, cluster head) and main operation (routing, perception).

Thirdly, the radio communication defines several parameters as the transmission power, the signal to noise ratio and the radio coverage. The radio communication is known to be the main source of power consumption in WSN. Thus, higher is the transmission, shorter is the sensor node lifetime. However, the variable transmission power could be a good solution in a cluster based approach, where the members limit their transmission power to reach their cluster head and this latter will use a higher one to reach its neighbor's cluster heads. The signal to noise ratio could be also investigated to select an aggregator node in a zone with the higher ratio to avoid the estimation error. This ratio could be also used to avoid the radio communication interference in dense zones of the network. The last parameter is the radio coverage, which insures that the supervised area is completely covered and the deployment of new nodes will not lead to the creation of isolated networks. It is also a parameter that may define the necessity of deploying new sensor nodes.

Thirdly, the information is certainly the goal of the WSN deployment. Therefore several methods exist to estimate the relevance of the gathered information, to estimate future information and to eliminate the redundancy. Thus, in this chapter, we discuss some of the parameters and the models that are used to determine the importance of the information and to estimate it in order to optimize the end-to-end delay.

The remaining of this chapter will discuss, in section 2, the sensor nodes characteristics and their possible influence on the WSN performance. Then, in section 3, we discuss the impact of the network deployment on the accuracy of the gathered data and on the optimal WSN lifetime. Next, in section 4, an analytical study is giving about the sensor nodes' information characteristics, in terms of relevance and prediction. Finally, the conclusion is given, in section 5.

2. Sensor nodes' characteristics

The hardware capacities of the sensor nodes define the limits of any application or optimization proposal in WSN. Indeed, the algorithms that they are not limited by the CPU, memory, radio communication or power constraints could lead to a high performance in terms of real-time communications and successful data delivery and precision. However, the main challenge in WSNs is the limited hardware capacities. Thus, in this section, we discuss the general types of the sensor nodes and we present their actual technologies advancement.

2.1 Sensor nodes' types: a classification by application nature

The WSNs are penetrating our daily life in several kinds of applications such as military, environmental, health, habitat, industrial, etc. Indeed, multitude types of sensor nodes equipment with various capacities and goals are proposed to achieve the requirement of theses daily applications. These types could be classified up on the nature of the applications Yick et al. (2008). In the remains of this section, some sensor nodes' types will be discussed based on their application requirements.

For large scale environmental applications as in forest, desert or normal natural conditions, Terrestrial Sensor Nodes (TSNs) Yick et al. (2008) Akyildiz et al. (2002) could be deployed. The TSNs are supposed to be inexpensive and deployed in hundreds to thousands in an area of interest. They could be deployed randomly as threw by plane or placed in pre-planned positions Stavros & Leandros (2006) Pompili et al. (2006) by humans or robots. These sensor nodes are self organized; they built up autonomously the network connection and communicate in a multi-hop manner. The TSNs have to communicate efficiently their environmental measures back to a base station. However, their limited batteries could be a big challenge. To the best of our knowledge, the communication is the main consumer of the power in sensor nodes, while the TSNs' batteries could be unchangeable due to, e.g., the hazardous zone of the deployed nodes. Therefore, several approaches have been proposed to reduce the power consumption of the sensor nodes: (1) optimizing the data communication routes to be shorter and energy-aware Ok et al. (2009) Chang & Tassiulas (2004), (2) defining optimal duty cycles within an energy-aware Mac layer Ye et al. (2002) Polastre et al. (2004), (3) reducing the number of communication sessions and the amount of communicated data by applying efficient data aggregation methods Sardouk et al. (2011) or even (4) attaching secondary batteries or solar charger to the sensor nodes.

In special industrial applications, such as underground mine or petroleum fields, more physically powerful sensor nodes are needed. Thus, the UnderGround Sensor Nodes (UGSN) Ian F. & Erich P. (2006) Li & Liu (2007) are supposed to be more expensive than the TSNs as they had to ensure reliable communication through soil, rocks, water and other mineral contents. The UGSNs' deployment is application tailored and it could not be generalized. Also, the network maintenance and post deployment are expensive and quite difficult due to the nature of the monitored mine or cave. In addition, there is a high probability of communication problems as signal loss and high level of interference and attenuation caused by the nature of the environment.

Similarly to TSNs, the UGSNs have strict power constraints as their battery could be unchangeable or unchargeable. Thus a power aware network deployment, communication, and data aggregation had to be studied.

The UndeWater Sensor Nodes (UWSN) are designed to be deployed in underwater applications Heidemann et al. (2006). Indeed, due to the underwater conditions, these sensor nodes are supposed to be more expensive than the TSNs and somehow less expensive than the UGSNs. However, the underwater WSNs applications are not supposed to be as dense networks. The typical challenge of the (UWSN) is the acoustic communication problem as the high propagation delay, the limited bandwidth and the signal fading. Moreover, the acoustic conditions increase the sensor nodes failure, which leads to serious network partitioning and data loss.

Here also, the UWSNs are similar to TSNs and UGSNs in terms of power constraints and impossibility of battery charging or replacing.

Another type of senor nodes that could be distinguished is for the multimedia applications Yick et al. (2008) Akyildiz et al. (2007). Thus, we call them as MSNs (Multimedia Sensor Nodes). They could be similar in physical forms to any one of the above mentioned types (TSN, UGSN and UWSN). However, the MSNs have, in addition, a built-in or attached cameras and they may require more powerful processing and storage units as they are supposed to communicate captured images, videos and/or sounds to a base station. Due to their nature, the radio entity of the MSNs should have some special specifications to ensure a minimal quality of service (QoS) level. The required QoS could be also influenced by the sensor node processor that may need to execute some image processing or compression before sending the results to a base station. However, the MSNs' deployment is generally pre-planned to ensure the aimed coverage level.

The TSNs, UGSNs, UWSNs and the MSNs could be fixed or mobile nodes. Indeed, the mobility could be an important issue as it may permit a better event or interest centric deployment. It offers a deeper and wider exploration of the area of interest. In terms of energy, the mobile sensor nodes are certainly more consumer, in order to supply the movement engine. However, they could be more efficiently chargeable throw sun panels as they could move to a better sun exposure.

The mobile and fixed TSNs, UGSNs, UWSNs and the MSNs could be used in numerous civil, military and industrial applications. In the above discussion, two main challenges could be pointed out. The first one is the limited power of the sensor nodes and the second challenge is the required reliable communication in various condition (underground, underwater, with QoS, etc.) The multimedia WSNs define also the importance of the processing and storage capacities. However, the optimization, in terms of power and communication, passes generally through algorithms as softwares for the application layer, or protocols for the transport, network or Mac layers. Indeed, more powerful are these algorithms, more the power and the communication are optimized. Thus, the processing unit capacity could also be a key factor in any optimization proposal for the WSN.

In the next section, a brief discussion of today technologies advancement in terms of processors speeds, memory storage and power consumption is presented.

2.2 Technologies advancements

In our days, TinyOs Hill et al. (2000) and sunSPOT Sun (2008) seem to be the most important technologies of wireless sensor nodes. The first one is a simple, lightweight event-based operating system written in nesC Gay et al. (2003) that is widely spread (it is used on Crossbow motes, Moteiv motes and similar devices).
The second, sunSPOT, is a product of Sun Microsystems, Inc. encompassing both hardware and software Sun (2008). The project started in 2003 on the experience of the company with the technologies related to java ME, and the first released occurred in April 2007. The recent release of platform Platon & Sei (2008) entails that the hardware provides among the most powerful sensor nodes, with similar size and scale factors of motes. The software part is independent from the hardware and consists of the Sun Squawk Java virtual machine Sun (2008).

Squawk is a closed-source JVM that encompasses necessary operating system functionalities, so that it can run directly on hardware Shaylor et al. (2003).
The remains of this section presents the hardware capacity of these technologies and a comparison with other technologies.

Hardware

A sensor node is made up of five basic entities: sensors, processor, memory, radio, and power entity. They may also Akyildiz et al. (2002)have application dependent additional components such as location finding system, a power generator and a mobilize.
Sensors are electronic devices that are capable to detect environmental conditions such as temperature, sound, chemicals, or the presence of certain objects. They send detected values to the processor which runs the sensor operating system and manages the procedures required to carry out the assigned sensing task. This processor retrieves the application code from the memory unit which stores also the operating system and the sensed values.

The radio permits to the wireless sensor nodes to communicate with other nodes, to receive commands and updates from the sink and to send sensed data to the sink.

The key element in a sensor node is the power entity which is generally composed of a couple of standard AA batteries. The size of these batteries usually determines the size of the sensor. Further, studies Baronti et al. (2007) are currently under way to replace/integrate battery sources with some power scavenging methods such as solar cells. In fact, there are some limits about the actual effectiveness of such methods. For example, solar cells do not produce much energy indoor or when covered by tree foliage.

In table 1 we compare some important sensor nodes such as Micaz, sunSPOT, TelosB, Sentilla and IMote2. The first three rows represent their processing and storage capacities, where the remaining rows represent their power consumption in three different cases: idle, active, and sleep. As shown in this table TelosB and Sentilla consume the least but they are very limited in term of storage and processing compared to Micaz, sunSPOT and IMote2. On the second hand, sunSPOT and IMote2 are the most powerful in terms of processing and storage but in the same time they consumes a lot of power. We can mention also that IMote2 has the biggest story capacity, which is due to its utilization in multimedia WSN. Hence, the IMote2 nodes are supposed to store captured images, videos and/or sounds that may require high a relatively high large space.

	MICAZ(Crossbow)	SunSPOT	TelosB	Sentilla	IMote2
Processor (Mhz)	16	180	8	8	13-416
Ram (kb)	4	512	10	10	256
Flash (KB)	512	4096	1024	48	32000
Active draw (ma)	48	70	25	4	>44
Idle draw (ma)	8	24	2	1	>31
Sleep draw (μa)	15	32	6	1	387

Table 1. Sensor nodes features

Based on this comparison, Micaz appears to be a combination between the first two nodes and the sunSPOT and IMote2. It consumes less then the powerful nodes and it is more powerful in processing and memory storage than TelosB and Sentilla.

2.3 Radio entity: importance and power consumption

The radio communication entity of the sensor node is certainly, the main entity to build the wireless network. This entity is known to be the main power consumer of the node. However, this consumption is due to achieve an acceptable level of reliable communication. In this section, we will illustrate that by showing the importance of the high transmitting power.

Let suppose a network divided into clusters, hence the cluster members will communicate their data to the cluster head (CH). Thus, the total amount of required transmission power used by the i-th sensor within a cluster Cui et al. (2005) is proportional to:

$$\begin{aligned} P^i(t) &\propto d_i^{\lambda}(N_t^i - 1) \\ &\equiv \| \boldsymbol{s}^i - L_{CH_t} \|^{\lambda} (N_t^i - 1) \end{aligned} \tag{1}$$

where, d_i is the transmitting distance (meters) between the CH and the i-th sensor, L_{CH_t} is the location of the CH at the sampling instant t and λ is the path loss exponent.

The importance of the high transmitting power could be illustrated in figure 1.This figure presents the average distance estimation error versus transmitting power, in a tracking application using the variational filtering (VF) based on quantized proximity sensors Mansouri et al. (2009) (see section 4). In the X-axis, we change the transmitting power of the sensor node. Then, on the Y-axis, we observe the influence of the transmitting power on the separating (between target and sensor node) distance estimation. Thus, we can mention that in low transmission power, the distance estimation error (RMSE) is at its higher value (around 6 meters). However, by increasing the transmission power, the RMSE become lower. We can observe also that, after a certain value, the transmitting power could be optimized and there is no need to choose higher values. **Power consumption of the radio entity**

The amount of energy consumed in a communication could be computed Sohraby et al. (2007) by equation 2, where E_{TX} is the power consumed during the transmission and E_{RX} is the power consumed during the reception. Both of them are computed following the data length and transmission distance (radio range of the node) (l,d);

$$\begin{aligned} &E_{TX}(l,d) = lE_c + led^s \\ &E_{RX}(l,d) = lE_c \end{aligned} \quad \text{where e=}\{ \quad \begin{aligned} &e_1 \;\; s = 2, \;\; d < d_{cr} \\ &e_2 \;\; s = 4, \;\; d > d_{cr} \end{aligned} \tag{2}$$

Where E_c is the base energy required to run the transmitter or receiver circuitry. A typical value of E_c is 50nJ/bit for a 1-Mbps transceiver; d_{cr} is the crossover distance, and its typical value is 87m; e_1 (e_2 respectively) is the unit energy required for the transmitter amplifier when d $< d_{cr}$ (or d $> d_{cr}$ respectively). Typical values of e_1 and e_2 are 10pJ/bit.m^2 and 0.0013pJ/bit.m^4, respectively.

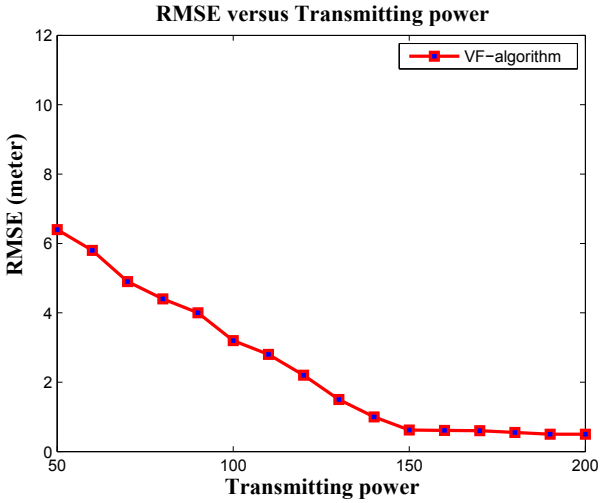


Fig. 1. RMSE vs Transmitting power varying in $\{50, ..., 200\}$.

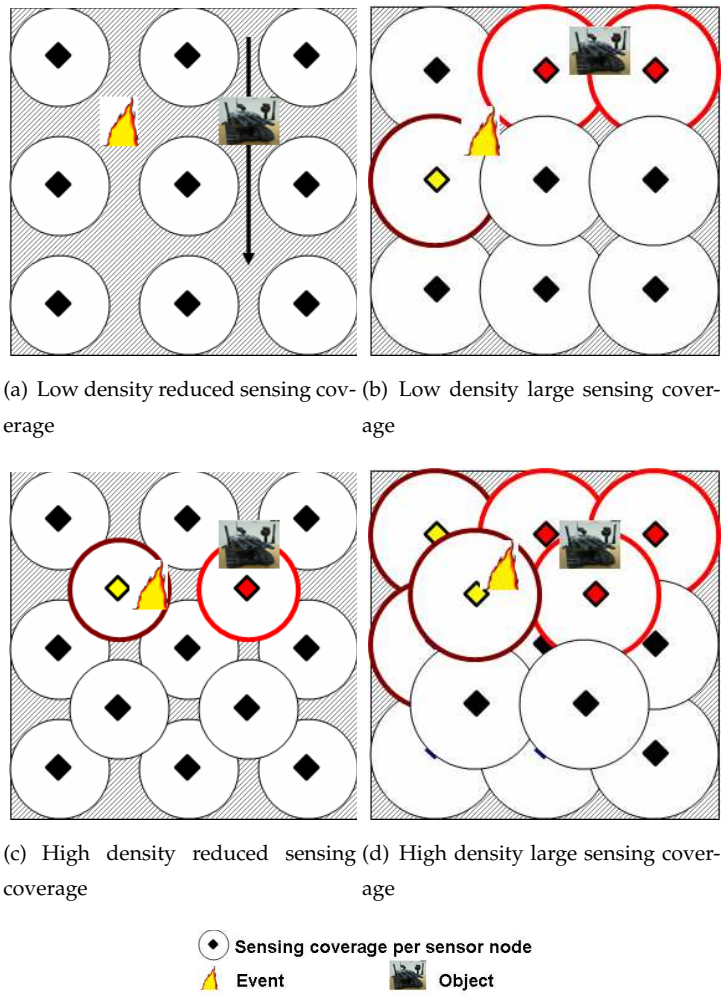
3. Impact of network deployment on data credibility

The data accuracy is one of the key factors for an efficient data aggregation in WSNs. Thus, the aggregated data need to represent, geographically, the maximum possible of the monitored area. Hence, several points related to the network deployment have to be adapted to have data representing the whole monitored area. Some of these points are as follows:

- Mainly the deployment model: In an accessible and small area, the sensor nodes could be placed one by one to insure a high representability of the monitored area. However, in large and inaccessible zones, the sensor nodes are supposed to be randomly deployed. Hence, the nodes could be grouped in some places while others are not covered;
- The network density: It represents the number of sensor nodes per square meter. This point could be easily managed in manual placed nodes. However, it seems to be difficult to manage in case of randomly deployed WSNs;
- The sensing coverage per sensor node: I.e. each node is supposed to represent a circle centered on it and with a radius r defined by the system developer.

The density and sensing coverage could impact together or separately the determination of the covered area, which could be illustrated in figure 2.

In figure 2a, the density is low and the sensing coverage is limited, hence the non covered surface (gray color) are important. By keeping the density low and enlarging the sensing coverage (figure 2b), the non covered area is reduced. Similarly, it is possible to increase the covered area, by keeping the sensing coverage limited and increasing the density (figure 2c).



(a) Low density reduced sensing coverage

(b) Low density large sensing coverage

(c) High density reduced sensing coverage

(d) High density large sensing coverage

Fig. 2. Impact of density and sensing coverage

Finally, by enlarging the sensing coverage and increasing the density (figure 2d), the non covered area could be reduced more and more.

The impact of the network deployment, in manual placed nodes, on data accuracy, is out of the scope of this section, as it is supposed to be influenced by the choices of the system developer. Thus, this section discusses the case of randomly distributed WSNs. The current analysis is based on the comparison between low density, and high density WSNs, with a variation of the sensing coverage per node. The comparison includes three types of random

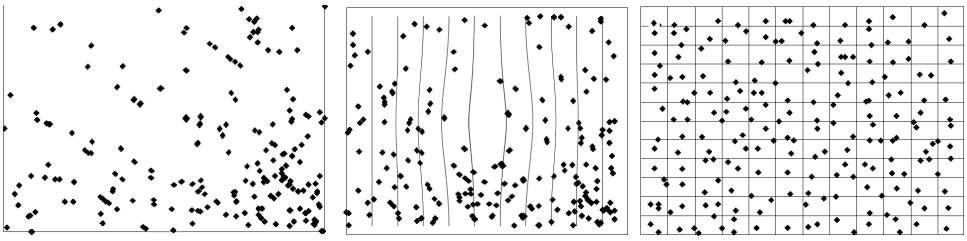
deployment, which are as follows:

- Uniform random distribution where all the nodes have equal probabilities to be placed in any position in the area;
- Column-based random distribution: It divides the network area into approximately equal columns. Then, it distributes the nodes randomly in each column. This type pf deployment is supposed to be closer to the reality than the first one;
- Grid-based random distribution: It divides the network area into approximately equal columns and rows. Then, it distributes the nodes randomly into the obtained cells. It is more complicated than the two others, however it is very probable in real applications.

For each one of these deployment methods, the current analysis discusses the distribution of the nodes and the percentage of the covered area regarding the whole monitored area.

3.1 Low density WSNs

Figure 3 illustrates the deployment of 200 sensor nodes in an area of 1000x1000 m^2. Figure 3a shows that, in a uniform random distribution, the density is high in the southeast quarter of the area, while it is very low in the northwest quarter. In the two other quarters, it is uniform. That means that the northwest quarter's data are not well represented while in the southeast quarter there is a redundancy in the data due to the correspondent density of nodes. Figure 3b, presents that in column-based random distribution the monitored area is better covered compared to the uniform random distribution. However, some zones are still better represented (south part) than others (north part). The Grid-based random distribution, Figure 3c, offers the best deployment where there is somehow an equitable representation of the monitored area.



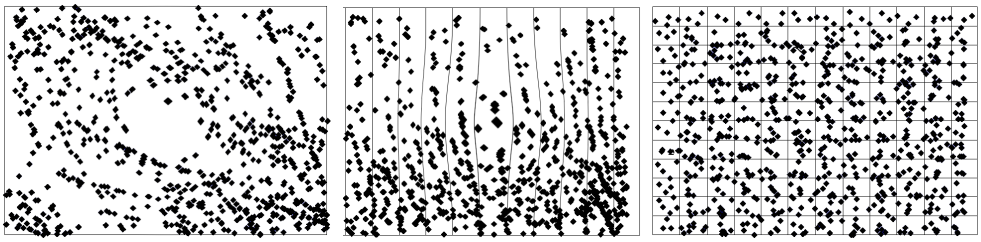
(a) Uniform random distribution (b) Column-based random distribution (c) Grid-based random distribution

Fig. 3. Impact of nodes deployment on data accuracy in low density WSN

3.2 High density WSNs

Figure 4 illustrates the deployment of 800 sensor nodes in the same area (1000x1000 m^2). Figure 4a shows that, in a uniform distribution, the density is high in the northwest and southeast sides of the area, while in the middle and borders it is low. In a column-based

distribution (figure 4b), it is much better except in the middle of the area, where it is not so representative. The grid-based method (figure 4c) distributes again equitably the sensor nodes over the monitored area.



(a) Uniform random distribution (b) Column-based random distribution (c) Grid-based random distribution

Fig. 4. Impact of nodes deployment on data accuracy in high density WSN

3.3 Network density

The network density is the number of nodes per square meter. It varies from one deployment to another and from one node to another within the same deployment depending on the node distribution.

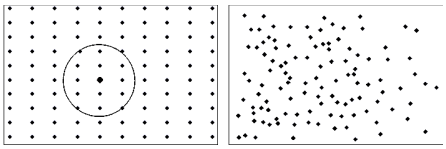
According to Akyildiz et al. (2002), this parameter does not have a fixed value to be used as a reference. The ideal value is application and environment dependent. In addition, this parameter has a network management importance as it helps to identify the dense zones of the network and the non well covered zones. Hence, it may lead to redeployment of more nodes in some zones for a better coverage.

We propose that each sensor node computes its own network density. There are two main reasons behind that: the first one is that each node has its dedicated view of the network (which is limited to its neighbors). The second and most important reason is the fact that for a specific task, the needs for a cooperation is between the sensor nodes of the same zone (geographical part of the network) and not farther nodes. For simplicity's sake, we propose the equation (3) to compute this density (D). In this equation, we compute the percentage of the real density compared to the theoretical density (both of them are explained later on), i.e., a density bigger than 100% could exist in the case of a very dense zone (in this case, the tendency of the sensor node will be toward the selfishness, hence to preserve its battery). It is important to note here, that if the density is greater than 200% it will be limited to this value to avoid an overweight estimation of density. Otherwise, if the density computed by a node is equal to 0%, it means that for example the node is disconnected from the network.

$$\begin{aligned} \text{D=} \quad & \frac{real density(RD)}{theoretical density(TD)} \\ \text{where,} & \\ \text{RD=} \quad & \frac{N_{real}}{(\pi \times r^2)} \\ \text{and,} & \\ \text{TD=} \quad & \frac{N_{theoretical}}{(\pi \times r^2)} \\ \text{hence,} & \\ \text{D=} \quad & \frac{N_{real}}{N_{theoretical}} \end{aligned} \tag{3}$$

Where r is the radio range of the sensor node, $N_{theoretical}$ is the theoretical number of nodes and it is given from the ideal distribution of the nodes or the grid distribution (figure 5a). $N_{theoretical}$ corresponds to the number of nodes within the radio range of a reference node (RN). A RN is a node in the center of the area to eliminate the special cases of border nodes.

N_{real} is the number of the one hop neighbor nodes, appearing on the neighbors or routing table of the node in question. N_{real} should be equal to $N_{theoretical}$ in the ideal case. Figure 5b shows an example of randomly distributed nodes to give an idea about real network densities.



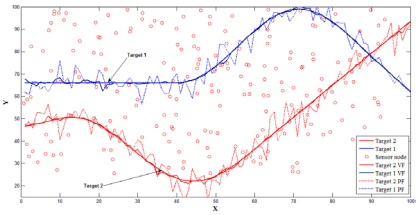
(a) Ideal distribution (b) Random distribution

Fig. 5. Network density comparison

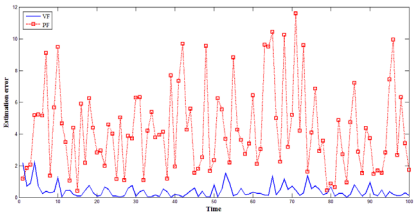
Impact of network density on multi objects tracking

Figures 6 and 7 compare the performance of two multi objects tracking algorithms (PF Djuric et al. (2003) and QVF Mansouri et al. (2009)) in a sparse and dense WSNs, respectively. figures 6a and 7a present the behavior of both algorithms, in tracking the two targets in question. We can observe that both algorithms behave better when the network is dense. That is due to fact that, in dense networks, the number of nodes detecting the object is higher; therefore the estimation of its position and its next position is more reliable.

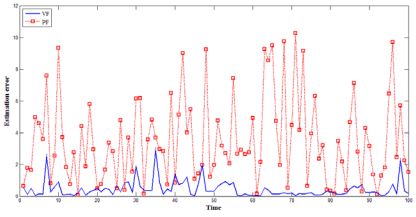
figures 6b,c and 7b,c presents the distance estimation error in both algorithms, for WSNs of 400 nodes and 800 nodes, respectively. In low density network (figure 6b,c), the distance estimation error in VF is in average variable between 0 and 1 meter. On the other side, in higher density (figure 7b,c), the distance estimation error is in average divided by 2, were it is approximately less than 0.5 meters. For PF, the optimization is similar, where the errors in low density (figure 7b,c)) are around 4 meters. Then, these values are approximately divided by two when the network density is increased (figure 7b,c)), thus the distance estimation error become less than 2 meters.



(a) Two objects tracking

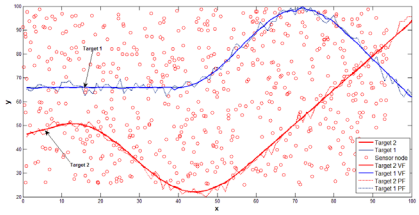


(b) 1st object : error estimation

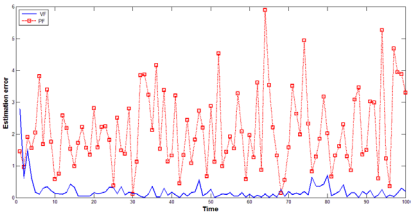


(c) 2nd object : error estimation

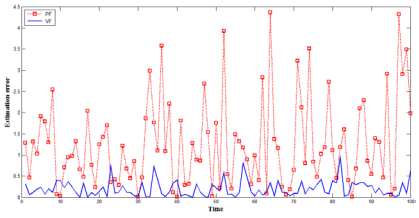
Fig. 6. Multi objects tracking in low density network, 400 nodes



(a) Two objects tracking



(b) 1st object : error estimation



(c) 2nd object : error estimation

Fig. 7. Multi objects tracking in high density network, 800 nodes

3.4 Node position within the network

Another parameter related to the network deployment has also to be studied carefully, due to its importance as we ill explain in this section. This parameter is the position (P) of the agent node in the network. We define three types of node positions: (1) normal, (2) edge and (3) critical. The normal position is the position inside the network where the node has multiple neighbors. This kind of nodes may tend toward the cooperative behavior, to maximize the amount of the important information collected in the network. The edge node (**E** in figure 8) is a node in the border of the network, which has a restricted view of the network limited to only one neighbor.

A node is considered in a critical position (**C** in figure 8) if it connects two parts of the network. That means, if the node runs out of battery, it may divide the network and multiple nodes behind it will become unreachable and in the best case they will require a longer route to communicate their data to the sink. This longer route is expensive in term of energy as the number of hops is increased. For example, in figure 8 , if a **C** node runs out of battery, the network will be divided in two parts.

A good strategy should allow a sensor node in a critical position to decrease its power consumption to maintain the connection between the two parts of the network the longest possible time. Thus, the value of the importance factor of the node position should help the sensor node to apply a selfish behavior and hence, e.g, it should be greater than or equal to the energy or the information importance degree factors.

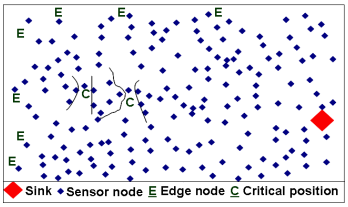


Fig. 8. Nodes' positions in the network

To facilitate the computation of P, we propose a fixed value for each type of node position. These values are 10%, 50% and 100% for the normal, edge and critical, position respectively.

Mean time before first partitioning

The mean time before first partitioning (MTBFP) in WSN, could be measured by the occurred duration before the loss of the first critical node. Thus, in Sardouk et al. (2011), we study a data aggregation method that takes into consideration the position of the sensor node. This method is simulated in two scenarios. The first one takes into consideration the position of the sensor nodes during the data aggregation (IIBC+P). The second scenario (IIBC) supposes that all the sensor nodes are equals.

In figure 9, we present a comparison, in terms of power consumtion, between the both scenarios IIBC and IIBC+P. As we can observe, IIBC+P decreases the average power consumption

of the critical sensor nodes in an important manner. It shows also that more the network is dense more the amount of decreased power is relevant. We can also observe that for 500 nodes, IIBC+P divides by more than 5 the consumption of these nodes compared to IIBC and for 700 and 900 nodes, this optimization remained important where IIBC+P divides the consumption by more than 4. In addition, in a non dense network, the power consumption has been divided by a factor of approximately 3.

Hence, we can deduce from these curves that IIBC+P offers a better power management for nodes in critical positions independently from the network scale and density.

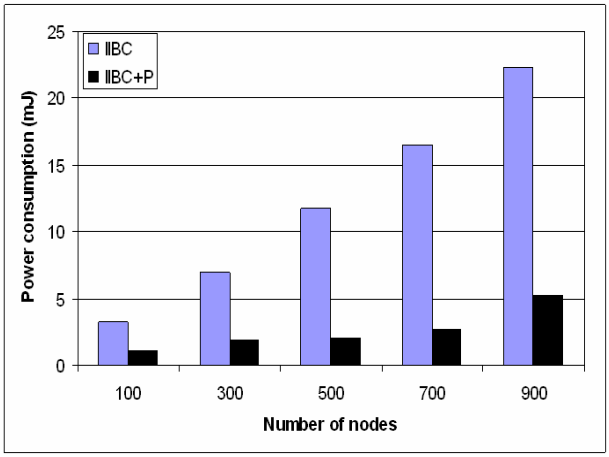


Fig. 9. Average power consumption per node in critical position

4. Information relevance: a study

The information relevance parameters computing is done following the model proposed in Mansouri et al. (n.d.); in which we assume that: i) the sensor measurements are quantized before being transmitted (a quantized proximity sensors is considered), ii) the application is the target tracking.

4.1 Quantized Observation Model

Consider a wireless sensor network, in which the sensor locations are known $\boldsymbol{s}^i = (s_1^i, s_2^i)$, $i = 1, 2, ..., N_s$. We are interested in tracking a target position $\boldsymbol{x}_t = (x_{1,t}, x_{2,t})^T$ at each instant t ($t = 1, ..., N$, where N denotes the number of observations). Consider the activated sensor i, its observation γ_t^i is modeled by:

$$\gamma_t^i = K\|\boldsymbol{x}_t - \boldsymbol{s}^i\|^\eta + \epsilon_t, \tag{4}$$

where ϵ_t is a Gaussian noise with zero mean and known variance σ_ϵ^2. The constants η and K are also assumed to be known. The sensor transmits its observation to the cluster head (CH) only

if the target is detected, which is equivalent to the condition that $R_{min} \leq \|\boldsymbol{x}_t - \boldsymbol{s}^i\| \leq R_{max}$ where R_{max} (resp. R_{min}) denotes the maximum (resp. minimum) distance at which the sensor can detect the target. Based on gathered transmitted sensor information, the cluster head is in charge of processing data in order to track the target. In order to save energy, before being transmitted, the observation is quantized by partitioning the observation space into N_t^i intervals $\mathcal{R}_j = [\tau_j, \tau_{j+1}]$, where $j \in \{1, ..., N_t^i\}$. The number $N_t^i = 2^{L_i}$ denotes the quantization level.

The quantizer is assumed to have an uniform step $\Delta = \frac{\tau_{N_t^i+1} - \tau_1}{N_t^i}$, with the initial and the last thresholds set to $\tau_1 = KR_{min}^{\eta} - \sigma_\epsilon$ and $\tau_{N_t^i+1} = KR_{max}^{\eta} + \sigma_\epsilon$, respectively. The quantization rule is then given by:

$$y_t^i = Q(\gamma_t^i) = d_j \text{ if } \gamma_t^i \in [\tau_j(t), \tau_{j+1}(t)] \tag{5}$$

where, the normalized d_j is given by $d_j = \frac{\tau_j(t) + \frac{\Delta}{2}}{\tau_{N_{t+1}}(t) - \tau_1(t)}$, and $Q()$ is the quantization function. Figure 10 depicts a simple example for the quantized observation model.

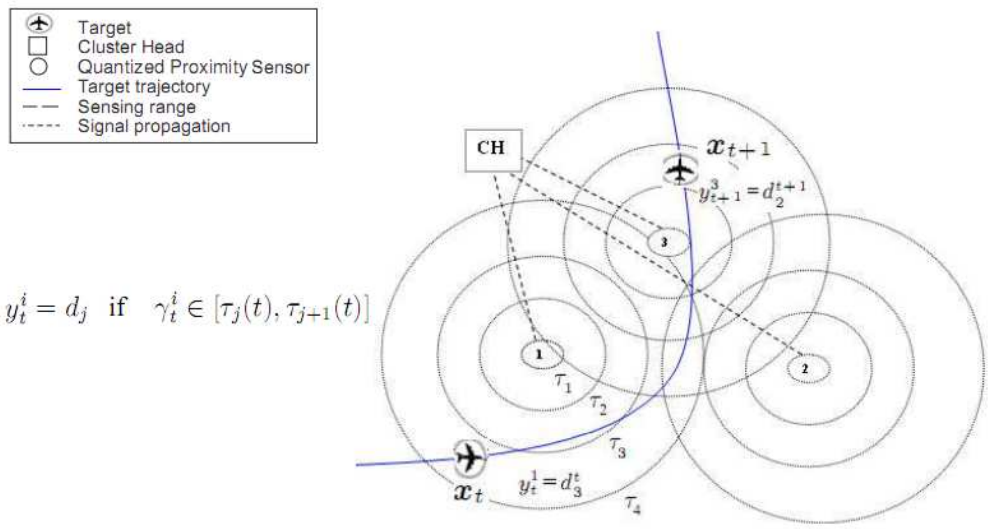


Fig. 10. The quantized observation model is described by a simple example. With respect to the first sensor, the target is within its sensing range at instant t. Observation y_t^1 is thus transmitted to the CH. However the second sensor keeps silent. The situation at instant t+1 can be similarly deduced.

Then, the signal received by the CH from the sensor i at the sampling instant t is written as,

$$z_t^i = \beta_t^i . y_t^i + n_t \tag{6}$$

where $\beta_t^i = r_i^{\lambda}$ is the i-th sensor channel attenuation coefficient at the sampling instant t, r_i is the transmission distance between the i-th sensor and the CH, λ is the path-loss exponent and n_t is a random Gaussian noise with a zero mean and a known variance σ_n^2. Figure 11 summarizes the transmission scheme occurring during the data processing.

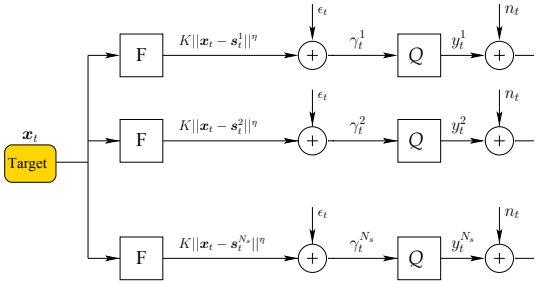


Fig. 11. Illustration of the communications path-ways in a WSN: The 1^{st} sensor makes a noisy reading γ_t^1. The quantized measurement $y_t^1 = Q(\gamma_t^1)$ with L_t^1 bits of precision is sent to the CH. The measurement z_t^1 is received by the CH, it is corrupted by an additive white Gaussian noise n_t.

The next section is devoted to the mutual information parameter computing.

4.2 Parameters that measure the information relevance of sensor measurements

The main idea of these parameters is to define the basic parameters that may influence the relevance of the sensors cooperation, which are: (1) information content that can be transferred from candidate sensor i; $\boldsymbol{MI}(\boldsymbol{x}_t, z_t^i)$ (detailed in 4.2.1, (2) the Fisher information matrix; $\boldsymbol{FI}(\boldsymbol{x}_t, z_t^i)$ (detailed in 4.2.2) and the Kullback Leibler distance (KLD), which is detailed in 4.2.3.

4.2.1 Computation of the Mutual Information function

The mutual information function is often used to measure the efficiency of a given information. The MI function is a quantity measuring the amount of information that the observable variable $\boldsymbol{z}_t$ carries about the unknown parameter $\boldsymbol{x}_t$. The mutual information between the observation z_t^i and the source $\boldsymbol{x}_t$ is proportional to Mansouri et al. (2009):

$$MI(\boldsymbol{x}_t, z_t^i) \propto p(z_t^i \mid \boldsymbol{x}_t) \log(p(z_t^i \mid \boldsymbol{x}_t)) \tag{7}$$

The likelihood function (L) is expressed as,

$$L(\boldsymbol{s}^i) = p(z_t^i | \boldsymbol{x}_t) = \sum_{j=0}^{N_i^j - 1} p\Big(\tau_j(t) < \gamma_t^i < \tau_{j+1}(t)\Big) \mathcal{N}\Big(h_t^i d_j, \sigma_\epsilon^2\Big) \tag{8}$$

where

$$p\Big(\tau_j(t) < \gamma_t^i < \tau_{j+1}(t)\Big) = \int_{\tau_j(t)}^{\tau_{j+1}(t)} \mathcal{N}\Big(\rho_{\gamma_t^i}(\boldsymbol{s}^i), \sigma_n^2\Big) d\gamma_t \tag{9}$$

is computed according to the quantization rule defined in (5), in which

$$\rho_{\gamma_t^i}(\boldsymbol{s}^i) = K\|\boldsymbol{x}_t - \boldsymbol{s}^i\|^\eta, \tag{10}$$

4.2.2 Fisher information matrix

The fisher information (FI) matrix is a quantity measuring the amount of information that the observable variable z_t^i carries about the unknown parameter $\boldsymbol{x}_t$. The FI matrix elements at the sampling instant t are given by:

$$\begin{aligned}\Big[FI(\boldsymbol{x}_t, \boldsymbol{s}^i, N_t^i)\Big]_{l,k} &= E_{z_t^i|\boldsymbol{x}_t}\left[\frac{\partial \log(p(z_t^i|\boldsymbol{x}_t))}{\partial \boldsymbol{x}_{(l,t)}} \frac{\partial \log(p(z_t^i|\boldsymbol{x}_t))}{\partial \boldsymbol{x}_{(k,t)}}\right] \\ &\quad (l,k) \in \{1,2\} \times \{1,2\}\end{aligned} \tag{11}$$

where z_t^i denotes the observation of the i-th sensor at the sampling instant t, $\boldsymbol{x}_t = [x_1, x_2]^T$ is the unknown 2×1 vector to be estimated, and $E_{z_t^i|\boldsymbol{x}_t}[.]$ denotes the expectation with respect the likelihood function $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$, which is given by

$$p(z_t^i|\boldsymbol{x}_t) = \sum_{j=0}^{N_t^i - 1} p\Big(\tau_j(t) < \gamma_t^i < \tau_{j+1}(t)\Big) \mathcal{N}\Big(\beta d_j, \sigma_\epsilon^2\Big) \tag{12}$$

Then, the derivative of the log-likelihood function can be expressed as,

$$\begin{aligned}\frac{\partial \log(p(z_t^i|\boldsymbol{x}_t))}{\partial \boldsymbol{x}_{l,t}} = \frac{\eta K}{\sqrt{2\sigma_n^2}}(x_{l,t} - s_{l,i})\|x_{l,t} - s_{l,i}\|^{\eta-2} \times \sum_{k=1}^{N_t^i}\Big[\exp\left(-\frac{1}{2}\frac{(\tau_k - \rho_{\gamma_t^i}(\boldsymbol{x}_t))^2}{\sigma_n^2}\right) - \\ \exp\left(-\frac{1}{2}\frac{(\tau_{k+1} - \rho_{\gamma_t^i}(\boldsymbol{x}_t))^2}{\sigma_n^2}\right)\Big] \times \exp\left(-\frac{1}{2}\frac{(z_t(k) - d_k)^2}{\sigma_\epsilon^2}\right) / \sum_{k=1}^{N_t^i}\Big[\operatorname{erfc}\left(\frac{\tau_k - \rho_{\gamma_t^i}(\boldsymbol{x}_t)}{\sqrt{2\sigma_n^2}}\right) - \\ \operatorname{erfc}\left(\frac{\tau_{k+1} - \rho_{\gamma_t^i}(\boldsymbol{x}_t)}{\sqrt{2\sigma_n^2}}\right)\Big] \times \exp\left(-\frac{1}{2}\frac{(z_t(k) - d_k)^2}{\sigma_\epsilon^2}\right)\end{aligned} \tag{13}$$

Substituting expression (13) in (11), the FI matrix is easily computed by integrating over the likelihood function $p(z_t^i|\boldsymbol{x}_t)$ at the sampling instant t.

4.2.3 Computing of the Kullback Leibler distance (KLD)

In certain problems, we would like to measure the distance between two statistical models. For example, this distance can be used in evaluating the training algorithm or classifying the estimated models Juang & Rabiner (1985). The Kullback-Leibler distance or the relative entropy arises in many contexts as an appropriate measurement of the distance between two distributions. The KLD between the two probability density functions p and $\widehat{p}$ is defined as Cover & Thomas (2006):

$$KLD(p||\widehat{p}) = \int p \log \frac{p}{\widehat{p}} \tag{14}$$

For hidden Markov models, the distribution function is very complex, and practically it can be only computed via a recursive procedure; the "forward/backward" or "upward/downward" algorithms Rabiner (1989); Ronen et al. (1995). Thus there is no simple closed form expression for the KLD for these models. Commonly, the Monte-Carlo method is used to numerically approximate the integral in (14) as:

$$KLD(p||\widehat{p}) = \boldsymbol{E}_p(\log(p) - \log(\widehat{p})) \tag{15}$$

5. conclusion

In this chapter, we have studied the parameters that may influence the performance of the WSN. We have started by the sensor nodes characteristics as battery, processor speed, storage capacity and radio communication. The sensor nodes types have been classified according to the probable applications as terrestrial, underground, underwater and multimedia. Indeed, the needed sensor node characteristics change from an application to another, as e.g., the high communication capacity needed in underwater applications to deal with the acoustic signal propagation problems, the battery optimization in the context of large scale terrestrial application, or the storage and processing problems to treat the captured images, videos and sounds in a multimedia WSN. In addition, the simulations have shown the importance of adjusting the transmitting power of the sensor nodes to reduce the estimation error in target tracking while maintaining the power consumption of the sensor nodes.

Later on, we have discussed the impact of the network deployment on the WSNs' performance, in terms of data accuracy and optimal lifetime maximization. This chapter has focused mainly on the case of random distribution/deployment of nodes, as the pre-planed deployments are generally adapted to some performance levels. We have shown through successive simulations the importance of the network density on reducing the distance estimation error, in the context of multi objects tracking. The simulations have proved also, the importance of taking into consideration, in any proposal, the position of each sensor nodes within the network. E.g., by applying special behaviors to sensor nodes in critical positions, we can maximize the occurred duration before the first network partitioning, which could help to optimally maximize the WSN lifetime.

Finally, this chapter has dedicated an important part to the processing of the information that we may have in the WSN. We have studied the parameters that could help to measure the relevance of the sensor nodes measurement. From these parameters, we have detailed the computation of the mutual information function, the fisher information matrix and the computation of the Kullback Leibler distance. We have also presented a computation model related to these parameters, which is the quantized observation model.

6. References

Akyildiz, I. F., Melodia, T. & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks, *Comput. Netw.* **51**(4): 921–960.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey, *The International Journal of Computer and Telecommunications Networking* **38**(4): 393–422.

Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A. & Hu, F. Y. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards, *Computer Communications* **30**(7): 1655–1695.
URL: *http://dx.doi.org/10.1016/j.comcom.2006.12.020*

Chang, J.-H. & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks, *IEEE/ACM Trans. Netw.* **12**(4): 609–619.

Chatzigiannakis, I., Kinalis, A. & Nikoletseas, S. (2008). Efficient data propagation strategies in wireless sensor networks using a single mobile sink, *Comput. Commun.* **31**(5): 896–914.

Cheng, L., Chen, Y., Chen, C. & Ma, J. (2009). Query-based data collection in wireless sensor networks with mobile sinks, *IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, ACM, New York, NY, USA, pp. 1157–1162.

Cover, T. & Thomas, J. (2006). *Elements of information theory*, John Wiley and sons.

Cui, S., Goldsmith, A. & Bahai, A. (2005). Energy-constrained modulation optimization, *IEEE Transactions on Wireless Communications* **4**(5): 2349–2360.

Djuric, P., Kotecha, J., Jianqui, Z., Yufei, H., Ghirmai, T., Bugallo, M. & Miguez, J. (2003). Particle filtering, *IEEE Signal Processing Magazine* **20**(5): 19–38.

Gay, D., Welsh, M., Levis, P., Brewer, E., von Behren, R. & Culler, D. (2003). The nesc language: A holistic approach to networked embedded systems, *In Proceedings of Programming Language Design and Implementation (PLDI*, pp. 1–11.

Heidemann, J., Ye, W., Wills, J., Syed, A. & Li, Y. (2006). Research challenges and applications for underwater sensor networking, *Proceedings of the IEEE Wireless Communications and Networking Conference*, IEEE, Las Vegas, Nevada, USA, pp. 228–235.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. & Pister, K. (2000). System architecture directions for networked sensors, *SIGPLAN Not.* **35**(11): 93–104.

Ian F., A. & Erich P., S. (2006). Wireless underground sensor networks:research challenges, *Ad Hoc Networks* **4**: 669–686.

Juang, B. & Rabiner, L. (1985). A probabilistic distance measure for hidden Markov models, *AT&T Bell Laboratories technical journal* **64**(2): 391–408.

Li, M. & Liu, Y. (2007). Underground structure monitoring with wireless sensor networks, *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 69–78.

Mansouri, M., Ouchani, I., Snoussi, H. & Richard, C. (2009). Cramer-Rao Bound-based adaptive quantization for target tracking in wireless sensor networks, *IEEE Workshop on Statistical Signal Processing (SSP)*.

Mansouri, M., Snoussi, H. & Richard, C. (n.d.). Robust target tracking with quantized proximity sensors, *2010,IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*.

Ok, C.-S., Lee, S., Mitra, P. & Kumara, S. (2009). Distributed energy balanced routing for wireless sensor networks, *Comput. Ind. Eng.* **57**(1): 125–135.

Platon, E. & Sei, Y. (2008). Security software engineering in wireless sensor networks., *Progress in Informatics* .

Polastre, J., Hill, J. & Culler, D. (2004). Versatile low power media access for wireless sensor networks, *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, ACM, New York, NY, USA, pp. 95–107.

Pompili, D., Melodia, T. & Akyildiz, I. F. (2006). Deployment analysis in underwater acoustic wireless sensor networks, *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, pp. 48–55.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications inspeech recognition, *Proceedings of the IEEE* **77**(2): 257–286.

Ronen, O., Rohlicek, J. & Ostendorf, M. (1995). Parameter estimation of dependence tree models using the EM algorithm, *IEEE Signal Processing Letters* **2**(8): 157–159.

Sardouk, A., Rahim-Amoud, R., Merghem-Boulahia, L. & Gaiti, D. (2011). Power-aware agent-solution for information communication in wsn, *Telecommunication Systems* **48**. To appear.

Shaylor, N., Simon, D. N. & Bush, W. R. (2003). A java virtual machine architecture for very small devices, *SIGPLAN Not.* **38**(7): 34–41.

Sohraby, K., Minoli, D. & Znati, T. (2007). *Wireless Sensor Networks, Technology, Protocols, and Applications*, WILLEY.

Stavros, T. & Leandros, T. (2006). Optimal deployment of large wireless sensor networks, *IEEE Transactions on Information Theory* **52**(7): 2935–2953.

Sun (2008). Sun^{TM} Small Programmable Object Technology (Sun SPOT) Theory of Operation, *Technical report*, Sun Microsystem, Sun Labs.

Ye, W., Heidemann, J. & Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks.
URL: *citeseer.ist.psu.edu/ye01energyefficient.html*

Yick, J., Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey, *Computer Networks* **52**(12): 2292–2330.

3

Smart Environments and Cross-Layer Design

L. Ozlem KARACA and Radosveta SOKULLU
Dokuz Eylul Univesity, Ege University
Turkey

1. Introduction

In the last decade we have witnessed a really unpredicted boom in the number and variety of applications based on wireless sensor networks (WSN). From environment monitoring and military applications, to health care and event tracking applications, both the diversity and complexity of the nodes themselves and their networked applications have increased immensely (Yick et al., 2008). A combination of consumer demand for more efficient integrated systems and a steep drop in the price of hardware fuelled by manufacturing process improvements has resulted in a noticeable upward cycle of research in the field of networks that not only sense the data but also provide automated reaction to specific situations known as Wireless Sensor and Actuator Networks (WSAN) (Akyildiz & Kasimoglu, 2004). "Smart environments" are discussed as the next step in these evolutionary developments in intelligent systems automation related to utilities, construction, industry, home and transportation. The "smart environment" is defined as one that is "able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment".

The WSN, which are in the heart of the "smart environments" consist of densely deployed microsensor nodes that continuously observe certain physical phenomenon. The existing abundance of WSN applications can be divided into two major groups based on the nature of the supported applications: WSN for monitoring and WSN for event detection/tracking. A major common feature is that both exploit the collective effort of nodes which have computing, transmitting and sensing capabilities. From the user point of view the main objective of WSN is to reliably detect or collect, and estimate event features based on the collective information provided by all sensor nodes. From the engineering design point of view, the main challenge for achieving this objective is posed by the severe energy and processing constraints of the low-end wireless sensor nodes. The collaborative sensing notion of WSN, which is achieved by the networked deployment of sensor nodes, can potentially be used towards overcoming the characteristic challenge of WSN, i.e., resource constraints. To this end, there has been a significant amount of research effort to develop suitable networking protocols in order to achieve communication with maximum energy efficiency. Because of the strict demands of WSN as compared to wired networks and Ad-Hoc networks, the design goals of such system are different from the traditional approaches. The suitability of one of the foundations of networking, the OSI layered protocol architecture, is coming under close scrutiny from the research community. It is repeatedly

argued that although layered architectures have served well for wired networks, they are not particularly suitable for wireless sensor networks. That is why the notion for a different approach, called cross-layer design, has come into existence.

Generally speaking, cross-layer design refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains. This is unlike layering, where the protocols at the different layers are designed independently (Srivastava & Motani, 2005). Cross-layer design stands as the most promising alternative to inefficient traditional layered protocol architectures allowing researchers to take into consideration different factors like the scarce energy and processing resources of WSNs, joint optimization and design of networking layers and last but not least overall performance evaluation. Accordingly, an increasing number of recent papers have focused on the cross-layer development of wireless sensor network protocols (Melodia et al., 2006). Recent papers (Cui et al., 2005); (Fang & McDonald, 2004); (van Hoesel et al., 2004); (Vuran et al., 2005) reveal that active cross-layer interactions and integration incorporated in the design techniques can bring about significant improvement in terms of energy conservation. The reasons have been summarized as follows:

- The significant overhead of layered protocols results in high inefficiency.
- Recent empirical studies necessitate that the properties of low power radio transceivers and the varying wireless channel conditions should be included in the protocol design.
- The severe restrictions on capabilities such as storage, processing and especially energy of the wireless sensor nodes make active interaction between different protocol layers mandatory.
- The event-centric approach of WSNs requires application-aware communication protocols.

It is obvious that the necessity has emerged for creating a new model that will inherently take into consideration the abovementioned specifics and restrictions of WSN.

Examining the literature in the area of cross-layer design, the following important observations can be made (Srivastava & Motani, 2005). First, there are several interpretations of cross-layer design. This is probably because the cross-layer design effort has been made rather independently by researchers from different backgrounds, who work on different layers of the stack. Second, some cross-layer design proposals build upon other cross-layer designs, hence some more fundamental issues (coexistence of different cross-layer design proposals, when cross-layer design proposals should be invoked, what roles the layers should play, etc.) are not addressed directly. Third, the question of how cross-layer interactions may be implemented has not been examined sufficiently; therefore the relation between the performance viewpoint and implementation concerns is weak. Furthermore, the wireless medium allows richer modalities of communication than wired networks. For example, nodes can make use of the inherent broadcast nature of the wireless medium and cooperate with each other. Employing modalities like node cooperation in protocol design also calls for cross-layer design.

Another very important aspect is related to the realization of the idea - cross-layer design proposals realized by different ways and manner exist in literature. Some of them focus on the idea of how actions in one layer affect other layer or layers (Wang & Abu-Rgheff, 2003); (Sichitiu, 2004). Studies exist also that consider the combined actions in two or three layers (Melodia et al., 2006); (Akyildiz et al., 2006); (Lee, 2006). However a cross-layer solution

generally decreases the level of modularity, which may lead to decoupling between design and development process, making it more difficult to design further improvements or introduce innovations. Moreover, it increases the risk of instability that can be caused by unintended functional dependencies, which are not easily foreseen in a non-layered architecture. Issues like these should be especially considered when trying to create and overall model or framework reflecting the inherent features and requirements of WSN.
Although a consistent amount of recent papers have focused on cross-layer design and improvement of protocols for WSNs, a systematic methodology to accurately model and leverage cross-layer interactions is still missing. Furthermore, the definition of a suitable, encompassing both performance and implementations issues cross-layer design (CLD) framework is required to unify the abundant research in WSN. Towards this aim we investigate the few suggested so far proposals for CLD frameworks which have quite different features and implementation methods focusing on the performance improvement and the consequent risks of a cross-layer design approach.
In this chapter we first introduce the cross-layer protocol design methodology for WSN and WSAN and review some major sources in literature. We focus on the concept of CLD frameworks, as a new emerging approach contrasting the well known conventional layered approach of protocol design. Our first aim is to investigate the ongoing work in the area of CLD framework, put that work in perspective, and consolidate the existing results and insights. Our second aim is to define some major criteria for comparing such frameworks and identify their pros and cons in terms of adaptivity, power efficiency, complexity, channel property orientation and fault tolerance.
From here on the chapter is organized as follows. In Section 2 we overview the concept of cross-layer design and the necessity for the development of CLD frameworks. In Section 3 we provide a definition of CLD framework and present a brief survey of the existing CLD frameworks in literature. Further elaborating on that subject in Section 4 we propose a set of criteria relevant to the evaluation of CLD frameworks and provide a detailed comparison of the discussed frameworks. Finally in Section 5 we provide a look ahead by discussing WSAN and the protocol design issues they pose. The chapter is concluded with some open research issues that we foresee for the development of a unified approach to protocol design in sensor networks suitable for smart environments.

2. Cross-Layer Design and Frameworks

To understand the concept of the cross-layer design and CLD frameworks, first the definition of layered frameworks should be elaborated. A layered architecture, like the seven-layer open systems interconnect (OSI) model (Stallings, 2006), divides the overall networking task into layers and defines a hierarchy of services to be provided by the individual layers. The services at the layers are realized by designing protocols for the different layers. The architecture restricts direct communication between nonadjacent layers; communication between adjacent layers is limited to procedure calls and responses.
Alternatively, protocols can be designed by violating the reference architecture, for example, by allowing direct active information exchange between protocols at nonadjacent layers or sharing variables between layers. Such violation of the layered architecture is what is known as the most popular definition of cross-layer design with respect to the reference architecture (Srivastava & Motani, 2005). There exist a number of studies that discuss and

evaluate the cross-layer design approach from different angles and formulate different positions on its applicability and possible disadvantages (Srivastava & Motani, 2005); (Melodia et al., 2006); (Zhang & Zhang, 2008); (Raisinghani & Iyer, 2004); (Wang & Abu-Rgheff, 2003); (Zhang & Cheng, 2003). However, the work of Srivastava and Montani (Srivastava & Motani, 2005), stands out as one of the most completed classifications available. The article presents detailed definitions and classification of cross-layer design and related interlayer interactions and the authors dutifully argue that they present a "taxonomy for classifying the existing cross-layer proposals and clarify the different interpretations of cross-layer design". Fig.1 summarizes their suggested taxonomy. They classify the possible methods for realizing cross-layer design in 6 groups and present examples for each one. The suggested taxonomy takes into consideration the interlayer interactions and their direction as well as the possible merging of layers up to the point where a totally holistic structure can be achieved (called "vertical calibration").

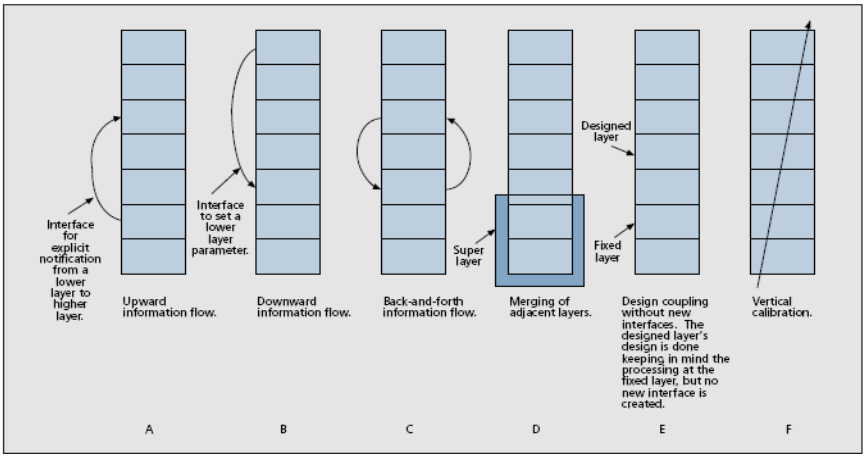


Fig. 1. Illustrating the different kinds of cross-layer design proposals. The rectangular boxes represent the protocol layers (Srivastava & Motani, 2005).

Another considerable attempt to put the discussion on cross-layer design on a well structured ground is given in (Melodia et al., 2006). The authors suggest a systematic methodology to model and leverage cross-layer interaction based on the assumption that the design of networking protocols for multi-hop sensor networks can be interpreted as the joint solution of resource allocation problems at different protocol layers. Thus they classify the proposals available in literature based on the number of protocol layers involved and the layers in the classical OSI model they try to replace. The focus is on expected performance improvement and the risks involved in the cross-layer approach. It is clearly stated that cross-layer solutions decrease the level of modularity and significantly increase the risk of instability brought by unforeseen functional dependencies and a joint solution is required.
(Zhang & Zhang, 2008) stress on the fact that cross-layer design allows active communication between different layers which ultimately can result in significant performance gains. Some of the new trends in wireless networking such as cooperative communication and networking, opportunistic transmission and real system performance

evaluation are discussed in light of QoS support for multihop sensor networks. The interaction between protocols at different layers is examined from the point of view of different system parameters controlled at distinct layers. For instance, it is argued that power control and modulation adaptation in the physical layer can affect the overall system topology, while scheduling and channel management in the MAC layer will affect the space/time reuse in the whole network. By using a general framework (Fig.2) they illustrate the interaction ideas and point out that all controls can have a multiple impact. (1) in Fig.2 illustrates the fact that assignment of channels to certain network interfaces changes the interference between neighboring channels. The authors conclude by pointing out that in order to achieve joint optimization of the whole system it is absolutely necessary to consider that all controls do cross different layers.

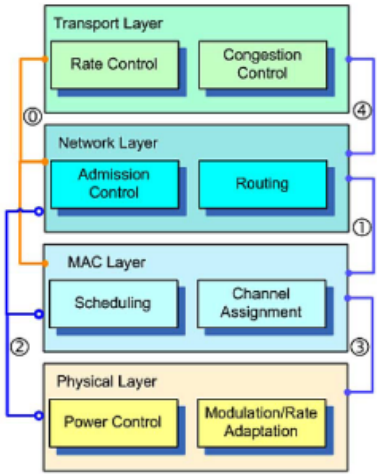


Fig. 2. Cross-layer framework and interaction among layers (Zhang & Zhang, 2008).

The experience gained through both scientific studies and experimental work in WSNs revealed important interactions between different layers of the network stack. These interactions are especially important for the design of communication protocols for WSNs.

The purpose of design principles is to organize and guide the placement of functions within a system. Design principles impose a structure on the design space, rather than solving a particular design problem. This structure provides a basis for discussion and analysis of trade-offs, and suggests a strong rationale to justify design choices. The arguments would also reflect implicit assumptions about technology options, technology evolution trends and relative cost tradeoffs. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology (Carpenter, 1996).

The general description of a framework states that it is a “basic conceptual structure” used to solve or address complex issues. A framework can be defined as an extensible structure for describing a set of concepts, methods and technologies necessary for a complete product design and manufacturing process. Regarding the CLD framework we can say that it should incorporate and reflect the inherent characteristics and specifics of WSN, and address the

major issues of performance and implementation in a joint manner for providing enhanced operation, energy efficiency and extending the lifetime of the network. As discussed before, numerous cross-layer solutions have been proposed so far taking into consideration a single or only a few, (mostly a combination of two or three) of the parameters of the WSN. Unfortunately the changes made affect other layers and might give rise to totally unpredicted situations and problems. Even if these situations and problems do not arise every time, in a different application, the suggested approach most probably will not provide the same functionality and optimization (Kawadia & Kumar, 2005); (Shakkottai et al., 2003); (Zhao & Sun, 2007).
To summarize, it is important to consider and evaluate the suggested cross-layer approaches in light of a basic conceptual structure, which is independent of the specific application and can provide adaptivity to system changes. In the next section, we continue with a survey, discussion and evaluation of the CLD frameworks suggested by different researcher teams.

3. Cross-Layer Design (CLD) Framework Proposals

To achieve understanding of WSN protocol design in terms of constituting CLD frameworks, we investigate four different CLD framework proposals. We examine each of them, in this section and give details of these proposals and their main features.

3.1 TinyCubus

Known applications of WSN fall into different classes and based on this the possible approaches to building a CLD framework can be subdivided into two major groups. The first one is using generic components and definitions while the second is using several more specific components or entities for each different class of applications. In (Marrón et al., 2005a) the architecture of a generic framework is presented, since its internal structure is the same independently of whether or not it is intended for all classes or just a certain number of applications.
The architecture of TinyCubus presents a single generic framework that can support very different application requirements even with contradictory requirements like environmental monitoring or target tracking. Its aim is to provide the necessary infrastructure to support the complexity of a specific WSN system architecture. TinyCubus consists of a Data Management Framework, (DMF) a Cross-Layer Framework, (CLF) and a Configuration Engine (CE). (Marrón et al., 2005b) The Data Management Framework allows the dynamic selection and adaptation of system and data management components. The Cross-Layer Framework supports data sharing and other forms of interaction between components in order to achieve cross-layer optimizations. The Configuration Engine allows code to be distributed reliably and efficiently by taking into account the topology of sensors and their specific assigned functionality.
The overall architecture of TinyCubus mirrors the requirements imposed by the two applications namely CarTalk 2000 (Tian & Coletti, 2003); (Morsink et al., 2003) and Sustainable Bridges (Marrón et al., 2005c) and the underlying hardware. It has been developed with the goal of creating a totally generic and fully reconfigurable framework for sensor networks. As shown in Fig. 3, TinyCubus is implemented on top of TinyOS using the nesC programming language, which allows for the definition of components that contain functionality and algorithms. The applications register their requirements and components with TinyCubus and are executed by the framework.

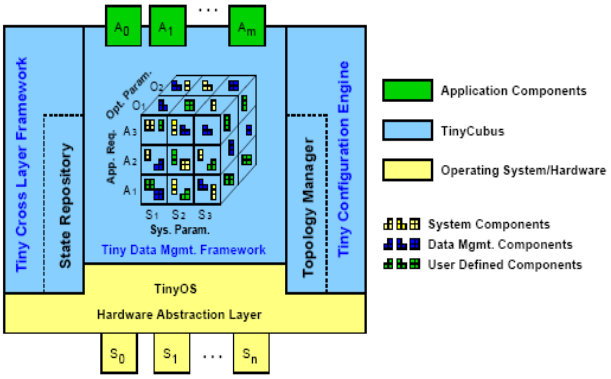


Fig. 3. Architectural components in TinyCubus (Marrón et al., 2005b).

The major design goal of TinyCubus is to support different application schemes easily and to do so it uses a generic framework. Despite all the differences, many applications obviously have some commonalities. Therefore, it is possible to simplify the development of both applications – and of others that share some properties with them.
Below the three major components of the TinyCubus Framework are discussed in more detail:

1. **Tiny Cross-Layer Framework**: The goal of the Tiny Cross-Layer Framework is to provide a generic interface to support parameterization of components using cross-layer interactions. The Tiny Cross-Layer Framework provides support for both parameter definition and custom code execution. This framework uses a specification language that allows for the description of the data types and information required and provided by each component. This cross-layer data is stored in the state repository. To deal with custom code, the cross-layer framework makes use of TinyCubus' ability to execute dynamically loaded code.
    a. State Repository: The cross-layer framework acts as a mediator between components. Cross-layer data is not directly accessed from other components but stored in the state repository. Thus, if a component is replaced (e. g., to adapt to changing requirements), no component that uses the old component's cross-layer data is affected by the change, given that the new component also provides the same or compatible data.
    b. Custom Code: The approach used in this study does not extend the interface of all components between two interacting ones. Instead, it provides support for the execution of application-specific code in lower-layer components via callbacks.
2. **Tiny Configuration Engine**: The Tiny Configuration Engine makes possible installation of new components, or swapping certain functions if necessary, by distributing and installing code in the network. Its goal is to support the configuration of both system and application components using cross-layer information about the functionality assigned to the nodes.
    a. Topology Manager: The topology manager is responsible for the self-configuration of the network and the assignment of specific roles to each node. A role defines the function of a node based on properties such as hardware capabilities, network neighborhood, location etc. Examples for

roles are SOURCE, AGGREGATOR, and SINK for aggregation, CLUSTERHEAD, GATE- WAY, and SLAVE for clustering applications as well as VIBRATION to describe the sensing capabilities of a node.

b. Code Distribution: Most existing approaches that distribute code in sensor networks do it by replacing the complete code image. However, most of the time only a single component needs to be updated or replaced. To avoid wasting energy by sending complete code image, configuration engine only transmits the components that have changed and integrates them with the existing code. The code distribution depends on the role of the node. Code updates only send to those nodes that belong to a given role and need this code update.

3. **Tiny Data Management Framework**: The goal of the Tiny Data Management Framework is to provide a set of standard data management and system components and to choose the best set of components based on three dimensions, namely system parameters, application requirements, and optimization parameters. The cube of Fig.1, called 'Cubus', represents the conceptual management structure of the Tiny Data Management Framework. When developing a suitable algorithm, at first, influencing factors called system parameters, such as density or mobility of the network is considered. Secondly, application requirements, such as reliability requirements, additionally restrict the set of possible algorithms. Finally, the algorithm is selected that fulfills best some optimization criteria, e. g., minimal energy consumption.

The strongest point in this framework proposal is its high adaptivity, the fact that it can be used for a number of different classes of applications. However, this comes at the price of high complexity and very general consideration of the wireless medium modalities.

3.2 DMA-CLD and the Optimization Agent Based Framework

The Optimization Agent Based (OAB) Framework (Lee, 2006) which is an extension of the cross-layer interaction approach suggested as the Dynamic Multi-Attribute Cross-Layer Design (DMA-CLD) constitutes a different class of framework for WSNs. It is based on the idea of systematically organizing the interactions between the layers by means of defining an optimization agent, serving as a core repository or database where essential information is maintained temporarily and exchanged across the protocol stack.

The DMA-CLD approach (Safwat, 2004), is proposed for cross-layer interactions in wireless ad-hoc and sensor networks to allow multiple, and possibly conflicting (single-layer, cross-layer, nodal, and networking) objectives to be met concurrently. While preserving the OSI layered structure, DMA-CLD allows interactions both upwards and downwards in the stack, i.e. information from the network layer can be passed both to higher or lower layers like the application and the MAC layers. It utilizes the Analytic Hierarchy Process (AHP) for making multiple, and possibly conflicting decisions. Thus the DMA-CLD can be viewed as a multi-objective framework that can be extended to accommodate any number of objectives and can relate to any number of OSI layers. It considers the network as a whole and reflects the objectives of selected "best network performance" on the parameters of the single node. DMA-CLD framework accepts a set of routes in the network, which are chosen to optimize the network performance according a given criteria ("high remaining battery capacity", "reliable packet delivery", etc.), as input.

The main idea of DMA-CLD is presented in Fig. 4.

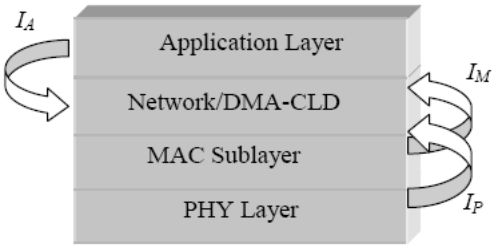


Fig. 4. The DMA-CLD framework and the associated cross-layer interactions (Safwat, 2004).

The key point involved in this approach is choosing multiple routes depending on a comparison matrix which includes the objectives listed precedence. It alleviates congestion by using multiple routes. The routes are ranked according to the Analytic Hierarchy Process (AHP). Putting together the information passed from the application, MAC and PHY layer a reciprocal pairwise comparison matrix C = [ci, j] is constructed for the multiple attributes (equation 1).

$$ci, j = \frac{1}{cj, i}, \forall i, j \in \Omega \tag{1}$$

where $\Omega \neq \varphi$ is the set of objectives. DMA-CLD computes a priority eigenvector via which each objective is assigned a priority. The eigenvector indicates how well each route satisfies each objective. The system also considers route outage. It is calculated by:

$$P_O = 1 - e^{-\frac{\gamma T}{\bar{\gamma}}} \tag{2}$$

where P_o is the link outage probability when the SNR threshold is γ_T and the average SNR is γ. The "route outage" value can be used by inter-layer feedback mechanism on the PHY layer. Thus, the operation of the DMA-CLD approach can be summarized as follows:

- The DMA-CLD is executed at the network layer. There the routes are ranked based on inter-layer feedback (provided by the interfaces I_A, I_M, I_P) and information from intermediate nodes and the first M paths are used for simultaneous load-balanced routing.
- The I_M interface is in charge of relaying MAC-specific information, such as the number of one-hop neighbors and the contention index, to the network layer.
- Information pertaining to the physical layer and the channel conditions, which is reflected in calculating the route outage, is carried to the network layer via the I_P interface.
- The application layer dynamically constructs the "pairwise attribute comparison matrix" taking into account the application requirements and network conditions such as traffic type, transmission delay bound, and transmission delay jitter bound. Then the reciprocal matrix C is constructed and conveyed to the network layer via the I_A interface.

The ideas involved in DMA-CLD were further extended in the OAB Framework, presented in (Lee, 2006). The major contribution of OAB is combining the inter-layer interactions as described in DMA-CLD in the form of a core repository, namely Optimization agent. The structure of the suggested framework is given in Fig. 5.

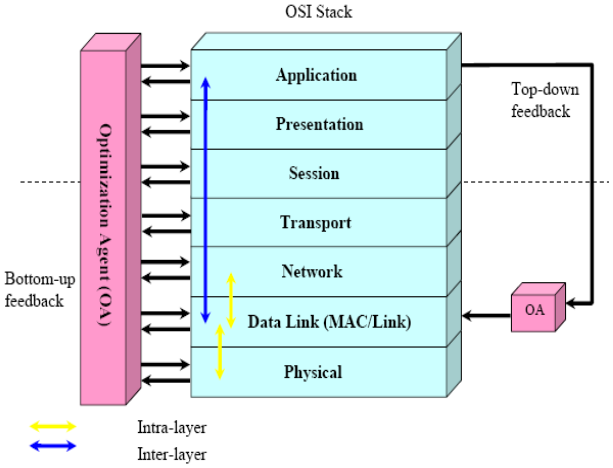


Fig. 5. The interactions of layers in Optimization Agent based design (Lee, 2006).

In the OAB framework the authors categorize the interactions between layers in two general groups: intra-layer (between adjacent layers) or inter-layer interactions (across two or more adjacent/nonadjacent layers). Both can be executed bottom up or top down.

- Bottom up interactions represent the typical feedback mechanism used in control systems. For example, information about the channel conditions at the physical layer is used at the link layer to adapt its error control mechanisms or at the application layer to adapt its sending rate.
- Top down interactions can be described as sending messages for the normal operation or data flow. An example is the sending of urgent messages for prioritized traffic from the application layer to the network layer or sending information from the MAC layer for tuning the transmission range at the PHY layer.

The structure of the OAB provides a framework that can accommodate changes or modifications to the protocol stacks for different network requirements or applications. It presents a generalization of a number of approaches that intend to optimize the performance between adjacent layers (e.g. MAC and network layers) (Liu et al., 2004); (Alonso et al., 2003). It extends the cross-layering process to all protocol layers as critical information kept in the OA can be exchanged across all layers and thus the performance is jointly optimized.

When compared to other frameworks the DMA-CLD and its extension OAB framework provide a direct possibility to take into consideration both channel oriented parameters and power efficiency by defining suitable objectives that influence the decision at the network layer. However the selection of the inputs for the reciprocal pairwise matrix is a very sensitive issue and the involved computational resources are considerable as the decisions

have to be taken in real time. Also the mechanism of accessing the information in the suggested OA and possible concurrency issues or race conditions have to be further elaborated as they pose a potential pitfall.

3.3 Horizontal Framework

In their work (Hakala & Tikkakoski, 2006), the authors suggest reducing the size and functions of the protocol stack and propose an additional cross-layer management entity to make application programming easier by simplifying the protocol stack in a way to better suit the limited resources available in WSNs. The role of the cross-layer management entity in this study is to offer a shared data structure and to take care of sensor network specific functions, like topology management and power saving. It also provides additional services that applications and other layers in the protocol stack can use. Data structures, which are in common use, are also implemented in the cross-layer management entity. So the two major entities, Application and Protocol Stack are responsible for the application-specific data transmission.

The cross-layer implementation provides reduced computational and memory requirements - not all the information needs to be transmitted between application interfaces and protocol layers. The other advantage is that the architecture also allows the implementation of the application and protocol stacks to be as simple as possible, since they are practically free of the tasks related to network management.

While taking into consideration some of the sensor network's special needs, it is obvious that there is a necessity of different solutions to be used. The system proposed uses horizontal architecture instead of the vertical model. Fig. 6 illustrates the major idea and components of the suggested horizontal CL framework for WSNs Above the physical layer and data link layer which are kept like in the classical structure, the architecture branches into two parallel areas. The ***Application*** and the ***Protocol Stack*** are responsible for the application-specific data transmission and the ***Cross-Layer Management*** (CLM) ***Entity*** takes care of network management. The CLM Entity is further divided into two parts: Management Entity, and Shared Data Structures.

The Management Entity is made up of one or more parallel modules, each of which takes care of a task affecting the operation of the sensor network node. Examples of these tasks include network management based on listening beacon messages, implementing a control algorithm that improves power saving characteristics, selecting efficient data transmission routes and so on.

The CLM entity is responsible for tasks directly related to the operation of the network but also general purpose tasks that are common to most WSN applications. Some of these, representing important modules in the CLM entity are summarized below:

- **Network configuring and topology management** –Topology management is an important cross-layer issue that is included in the CLM entity. It is vital to monitor the state of the surrounding network, for example, battery charges in neighboring nodes, network control traffic including beacon messages or other control messages. Using the information provided by the CLM entity, resources of the network can be employed effectively.

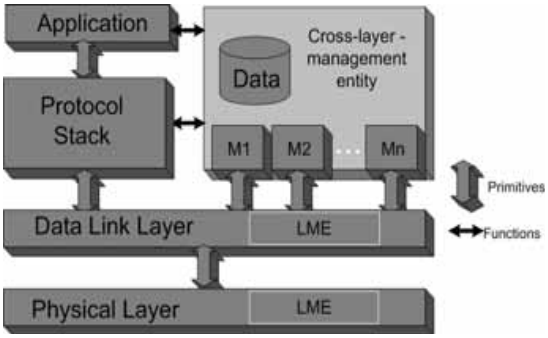


Fig. 6. Horizontal cross-layer architecture (Hakala & Tikkakoski, 2006).

- **Providing optimal data transmission routes:** Routing in the WSN is a major factor in providing efficient network operation. In a lot of cases multi-hop and more power efficient methods might be sought then the general flooding algorithm. Deciding in the optimal route affects both the operation of the single node and its duty cycle and the topology of the whole network so it is considered one of the main modules in the CLM entity.
- **Providing optimal power mode selection for the node:** This includes tasks as moving the node into power saving mode or providing other power related solutions whenever feasible:
    - For the implementation of short duty cycles, the mechanism such as on/off type switching can be used. To extend the lifetime of a battery-powered device into many years, the duty cycle must be as short as possible.
    - Selection of the node's optimal transmitting power is also classified as a power saving issue. Listening consumes more energy than sending, because the receiver must be kept on independent of whether there is any traffic on the channel or not. However, energy can be saved by adjusting the transmitter power. This also provides that disturbances to other nodes are minimized.
- **Sharing data structures:** Lot of the operations in the network as self-configuration, routing information exchange, power saving etc. are interrelated. For this reason they cannot be easily confined to any particular layer. To minimize memory and computational requirements, the authors suggest the use of the so called Shared Data Structures. An example of such usage is adjusting the optimal broadcast power knowing the neighbor's data. However, Sharded Data Structures have to be very clearly defined as there might be unforeseen dependencies.
- **Coding/decoding:** Coding/decoding is a general purpose operation is not dependent on the protocol stack used. Therefore, it can be done in the CLM entity. Algorithms used in coding may include, among others, different compression and encryption algorithms.

As can be deducted from the discussion presented above the main idea of the Horizontal Framework is to simplify the protocol stack and separate certain tasks as modules of the CLM entity, thus making application programming easier. The low stack reduces the data

transfer between the different layers. At the same time, the reduced header information by means of the CLM entity results in a reduced number of bits to be transmitted. Power consumption in data transmission is directly proportional to the length of the broadcasted frame, so the system ensures extending network lifetime. The interface between the CLM entity and the Application/Protocol Stack employs the client/service principle. The CLM entity can provide certain services that the layers in the protocol stack and the application can use. Usually, the function of communication in this interface is to perform a certain task, for example the updating of Shared Data Structures. Because the application program can be freed from the tasks related to network management and some general purpose tasks, it is possible to have a very simple application program. The system also allows the use of the same sensor network structure for a great number of different applications.
The Horizontal framework provides high degree of adaptivity to different applications while at the same time involves much less complexity then the TinyCubus framework. The suggested management entity directly interacts with the MAC layer, with the network and application layer providing duty cycle control, topology control and other solutions to extent the overall lifetime of the network. However it does not define how modifications in the Shared Data Structures should be taken into account. The dependencies between the modules and the suggested common data structures might bring out unexpected complicacy. In the example presented by the authors, two management modules are proposed – the power saving and the topology control module. They do provide the required efficiency related to the example at hand (CiNet) but for other applications the number of these modules might have to be increased resulting in a much higher complexity.

3.4 XLM

XLM (cross-layer module) (Akyildiz et al., 2006) is a unified cross-layer module which is developed to achieve efficient and reliable event communication in WSNs with minimum energy expenditure. XLM merges common protocol layer functionalities into a single cross-layer module for resource-constrained sensor nodes. The operation of the XLM is devised based on a new notion, which the authors define as "initiative determination". It is the core of the XLM and implicitly incorporates most of the the inherent communication functionalities required for the successful operation of a general application oriented WSN. Based on the initiative concept, XLM performs received based contention, local congestion control, and distributed duty cycle operation in order to realize efficient and reliable communication in WSN.
The basis of communication in XLM is built on initiative concept. In this concept, each node decides whether join a network and participate a communication or not according to the initiative value. Consequently, a completely distributed and adaptive operation is deployed. The next-hop in each communication is not determined in advance. Instead, an initiative determination procedure is used for each node to decide on participating in the communication.
Operation based on the initiative concept in (Akyildiz et al., 2006) can be summarized as follows: A node starts transmission by broadcasting an RTS packet to indicate its neighbors that it has a packet to send. Upon receiving an RTS packet, each neighbor of node **i** decide to participate in the communication or not. This decision is given through initiative determination. The initiative determination is a binary operation where a node decides to

participate in communication if its initiative is 1. Denoting the initiative as I, it is determined as follows:

$$I = \begin{cases} 1, if \begin{cases} \xi_{RTS} \geq \xi_{Th} \\ \lambda_{relay} \leq \lambda_{relay}^{Th} \\ \beta \leq \beta^{\max} \\ E_{rem} \geq E_{rem}^{\min} \end{cases} \\ 0, otherwise \end{cases} \quad (3)$$

The initiative determination value is calculated based on four variables. Each of them represents a necessary threshold value that should be satisfied. The initiative is set to 1 if all four conditions declared above are satisfied. Each condition in inequality (3) constitutes certain communication functionality. The first condition ensures that reliable links are to be constructed and for this purpose, it requires that the received signal to noise ratio (SNR) of an RTS packet, ξ_{RTS}, is above some threshold ξ_{Th} for a node to participate in the communication. The second and third conditions are used for local congestion control. The second condition prevents congestion by limiting the traffic a node can relay. The third condition ensures that the node does not experience any buffer overflow and hence, also prevents congestion. The last condition ensures that the remaining energy of a node E_{rem} stays above a minimum value, $E_{min\ rem}$. This constraint guarantees even distribution of energy consumption. The cross-layer functionalities of XLM are summarized in these constraints defining the initiative of a node to participate in communication.

Each node performs distributed duty cycle operation. The value of the duty cycle is denoted by δ and defines the ratio of the time a node is active. Each node is implemented with a sleep frame with length TS sec. As a result, a node is active for δ × TS sec and sleeps for (1 − δ) × TS sec. There are two main duties according to which sensor nodes can be classified: source duty and router duty. The source duty refers to the nodes with event information that need to transmit their packets to the sink; hence these types of nodes can select their rates based on the congestion in the network. The router duty refers to the nodes that forward the packets received from other nodes to the next destination. These nodes indicate their initiative on accepting new flows through their path to the destination. Based on these duties, each node determines its initiative to participate in the transmission of an event as explained above.

When a node wants to send a packet, it first listens to the channel. If the channel is idle, the node broadcasts an RTS packet, which contains the location of the sensor node i and the location of the sink. By getting the packet, other nodes in networks, decide whether or not they are located in a feasible region or in an infeasible region. The node located nearer to sink is “in feasible region”, otherwise it is “in infeasible region”. Only nodes located in feasible region initiate the procedure, nodes located far are switched to sleep mode to save energy. If a node decides to participate in the communication, it performs receiver contention. Following the receiver contention procedure node i receive a CTS packet from a potential receiver and send a DATA packet indicating the position of the winner node in the header so the other nodes stop contending and switch to sleep. Since each time only a small

number of nodes contend in the selected "priority regions" the collision probability is small in XLM.
Two sources of traffic are considered as an input to the buffer of each node:

- Generated packets: The sensing unit of a node senses the event and generates the data packets to be transmitted by the sensor node during its source duty. It is referred to these packets as the generated packets. For a node i, the rate of the generated packets is denoted by λ_{ii}.
- Relay packets: As a part of its router duty, a node also receives packets from its neighbors to forward to the sink due to multi-hop nature of sensor networks. These packets are referred as the relay packets. The rate at which a node i receives relay packets from a node j is denoted as λ_{ji}.

The main idea of XLM cross-layer congestion control is to regulate the congestion. XLM has two main congestion control measures:

- In router duty - enabling the sensor node to decide whether or not to participate in the forwarding of the relay packets based on its current load arising from its relaying functionality
- In source duty - explicitly controlling the rate of the generated data packets.

For realizing congestion control, besides regulating the relaying functionality by the initiative determination, the XLM allows local congestion control by directly regulating the amount of traffic generated and injected to the network at each node.
This framework presents a novel approach in considering a number of network and physical layer requirements by combining them in a very simple structure. However it does not include any fault tolerant mechanisms and being predominantly a network layer based solution does not directly address any issues at the application layer. It also implicitly assumes that all nodes have exact information about their own location and centralized information about the location of the sink.
After this overview of the suggested in literature examples of CLD Frameworks, we proceed, in the next section with a discussion of the relation between WSN application requirements and the functionality of a basic conceptual protocol structure that would meet the specifics and limitations of WSN protocol design.

4. Evaluation of the Existing Frameworks

After suggesting a possible unified approach to comparing diverse WSN application, the Application Comparison Matrix, in the section above, our discussion continues with an attempt to define suitable criteria for evaluating CLD frameworks. Further on in this section we propose a detailed comparison of the CLD frameworks surveyed in section 3.

- **Adaptivity:** The adaptivity evaluates the extent to which a framework can easily and in a fine grain manner adapt itself to the changes in the requirements of heterogeneous applications, to different hardware platforms and to different network topologies. As can be seen from the selected applications, sometimes the differences in their requirements can be even conflicting. For example the Sustainable Bridges application (Marrón et al., 2005a; 2005b; 2005c; 2005d) implies a pushed based data model while the Car Talk 2000 (Tian & Coletti, 2003; Morsink et al., 2003) needs a pull based one. In some very specific oriented applications, like for example Forest Fire Detection (CRUISE 2007) nodes might perform very simple

tasks and the required hardware might be greatly simplified, while in others like Sense-R-Us (Lachenmann et al., 2005) the need for diverse information collection and its management might require more sophisticated hardware platforms and functionality. Last but not least changes can occur because of the highly erratic nature of the wireless channel which reflects directly on the network topology and connectivity.

- **Power efficiency:** The most restricted resource in wireless sensor networks is the power of the nodes. It is very important how the suggested framework takes this issue into account. In some frameworks like for example the XLM the power efficiency is considered in a totally distributed manner, at the single node level. On the other hand in the Horizontal Framework this issue is considered both at the node level, by introducing a special management module called the "power saving module" and at the network level by the so called "topology control module". Thus by introducing different modules, the Horizontal Framework provides possibilities for versatile and fine grained control over the power consumption in the node iteself and in the network as a whole. In this respect the TinyCubus provides the most detailed approach but of course at the price of very high complexity.
- **Channel-oriented:** Wireless channel is inherently unsteady. The frameworks that take into consideration this feature can be classified as channel-oriented. They allow for fine tuning of the network operation and management involving in a fairly direct way the channel characteristics.
- **Fault tolerance:** There are many sources that might alter the successful transmission of information and the efficient operation of the network as a whole. Faults might originate because of the mobility of the nodes, fluctuations of the channel, excessive channel utilization due to high density deployments etc. Measures should be taken to minimize the effect of such phenomena and their effect on the network. The fault tolerance criterion takes into account how such issues are covered in the suggested framework.
- **Complexity:** A proposed framework might take into consideration all possible cases and specifics related to a large number of applications but this would result in a structure too difficult to implement and manage. The complexity is an important implementation oriented parameter that has to be taken into account when evaluating the CLD framework.

The design goals and main concerns of the frameworks discusses above are quite different and each has distinctive features, advantages and disadvantages from a specific point of view. Based on the criteria specified we classified the existing frameworks and the results are presented in the Table 1. below:

Property	TinyCubus	DMA-CLD	Horizontal	XLM
Adaptivity	■■■■	■■	■■	■
Channel-oriented	■■■■	■■	■■■	■■■■
Power efficiency	■■	■■	■■■■	■■■■
Fault tolerance	■■	■■	■■	■
Complexity	■■■■	■■■	■■	□

□ Not important ■Little ■■ Medium ■■■High ■■■■ Very important

Table 1. Frameworks comparison table.

TinyCubus aims to provide a framework that can easily and in a fine grain manner adapt itself to the changes arising from heterogeneous applications, to different hardware and to different network operation. The topology manager in the TinyCubus framework and the role-based code distribution algorithm are used to provide dynamic code distribution and allow very high degree of adaptivity. This framework can be applied quite successfully to develop both applications like Sustainable Bridges and Forest Fire Detection as well as more complex interaction-based ones like the Sense-R-U and CarTalk 2000. In (Marrón et al., 2005a) it is proven that the role-based code distribution algorithm reduces the messages sent to nodes which need update information compared to general flooding. Suitably selected algorithms can be applied for regulating the duty cycle for sending and receiving mode allowing medium to high degree of energy efficiency. Also, mobility of the nodes and partially the specifics of the transmission channel/environment can be taken into consideration by distributing suitable code using the CE. Even though not explicitly mentioned in the article, with some further effort, fault tolerance issues can be incorporated. However, on the other hand, the TinyCubus, being so detailed and encompassing, is far more complex when compared to other frameworks. From implementation point of view it presents a real challenge. The complexity evaluation based on the number of messages to be exchanged for distributing new code relies on a single and very restricted example which does not justify the general case.

The DMA-CLD and also the OAB frameworks present an interesting view for creating a "common entity" used to simplify the traditional protocol stack and provide more efficient network operation. It builds on the general direction of the research in CL design and optimization so far that evolves around inter-layer and intra-layer interactions and parameter exchange. The functions of the existing layers are kept intact, while the data structures and available data are unified in a common entity. Thus it can provide high degree of channel-oriented operation because the common access to data about the channel conditions can be used directly by other layers to optimize performance at node and network level. Also certain degree of interoperability will be ensured as the layered stack is preserved. Even though existing work in CL design based on optimization of the operation of two or more layers, proves that such type of solutions do bring overall energy efficiency the suggested approach has some pitfalls. First of all, the access to the OA is a potential source of problems and can bring about additional complexity instead of reducing complexity. Second, race conditions will be difficult to track and deal with. Last but not least the suggested approach does not allow for efficient and adequate to WSNs solution of some interlayer functions as topology control and fault tolerance. On the whole, even though a certain degree of optimization can be achieved the DMA-CLD and the related OAB framework do not seem to provide high adaptivity neither from implementation nor from performance point of view. If we consider the applications mentioned in section 4 it is clear that this framework has to be further modified based on the "class" of applications addressed. For example, applications like Sustainable Bridges and Forest Fire Detection can be developed based on a subset of this framework optimized for environmental monitoring while applications like CarTalk 2000 and Sense-R-U might result in unforeseen complications and problems due to the more intricate and generic information interaction involved.

A different way of separating a "common entity" from the traditional protocol stack is presented in the idea of the Horizontal framework. In this case the separation is based on

functions not on data structures. The Horizontal framework provides a separation of the functions currently covered by the different layers of the OSI model by selecting some that are not definitely related to a fixed layer and creating a new "horizontal" or "cross-layer" entity called CLM entity. This new entity has a modular structure in itself where modules are roughly corresponding to different tasks that might be related directly to network operation (topology management, energy efficient routing etc.) or might be more general and related to the single node (duty cycle determination, switching between different power modes at the node level etc.). The Data Link Layer and the Physical Layer are preserved but some of their general purpose functions are transferred to modules in the CLM entity. As a result of this organization the Horizontal Framework provides a simplification of the application/protocol stack and makes programming easier. It provides a high degree of adaptivity in a simplified structure and allows for different approaches to dealing with power efficiency issues both at the node and network level. Fault tolerance is not directly resolved. A major advantage is that it tries to balance the advantages of CL and traditional design by preserving partially the layered architecture. However, from implementation point of view the interoperability between the modules in the CLM is under question especially if their number is increased (the authors illustrate their idea with two modules). Further more the boundary between which operations or issues should be separated from the Physical and Data link and included as modules in the CLM and those which should be kept is not clearly defined. This also leads to implementation problems. However we believe that a further elaboration in this direction is very promising and might lead to resolving in an optimized way both the performance and the implementation issues. We can support this idea by using the Horizontal Framework as a generic development platform for the applications discussed. As the Sustainable Bridges and Forest Fire Detection have similar optimization parameters including similar modules in the CLM to realize these functions will provide the required adaptivity. On the other hand the addition of cross-layer module handling mobility issues can easily take into account the additional application requirements raised by adding a mobile robot in the Forest Fire Detection scenario. Furthermore, elaboration on the additional functions required by the CarTalk2000 and Sense-R-U applications can be handled partially in the application layer of the simplified stack and partially by adding new modules in the CLM. Thus it is obvious that without significant increase in the complexity new diverse application requirements can be addressed.

A very untraditional approach is presented in the XML framework. It starts from scratch and defines a totally new architecture based on the communication model and the requirements specific to WSNs. It redefines the principle of network operation based on a totally distributed approach. Each node takes a decision of participating or not participating in the network operation based on specific locally (including single node level and immediate neighborhood level) evaluated criteria. Such a conception is very straight forward and simple both from performance evaluation and implementation point of view. While it provides very high degree of adaptivity regarding different applications it does take for granted a certain high hardware standard. Nodes are aware of their location and have comparatively high computational abilities. Still this adaptivity does not come at the price of higher complexity as is the case with the other mentioned frameworks and especially TinyCubus. It resolves in an elegant way the issues of power efficiency and relation to the dynamically changing channel conditions but does not take into

consideration fault tolerance. It allows for possible extensions of the selected set of parameters to include fault tolerance. Thus XLM presents a very new direction in CLD framework design which requires further research for understanding its implementation implications. Generically, the XML framework should be able to answer both the monitoring type of applications (Sustainable Bridges and Forest Fire Detection) and the more interactive ones (CarTalk 2000 and Sense-R-U). Unfortunately the authors do not provide any details on its relation to specific parameters of the application layer so it is difficult to make any remarks on that point.

5. From WSN to "smart environments"

We have so far concentrated mainly on the issues of cross-layer design related directly to WSNs. However, the future "smart environments" do not only collect information from the environment. As the definition was given in the introduction of this chapter they will "acquire and apply knowledge about the environment to improve the users' experience". Thus not only sensing nodes will be required but also "acting" nodes, known as "actuators" or "actors". While the sensor nodes are very low-power, low-cost sensing devices with very limited communication and processing capabilities the actor nodes are more resource rich nodes, equipped with better communication abilities (more processing power, larger transmission range) and longer battery life. These networks as defined in (Akyildiz & Kasimoglu, 2004) are known as Wireless sensor and actuator networks -WSAN (Fig. 7). Furthermore, while there might be hundreds or thousands of sensor nodes, very densely deployed in a given area, such a dense deployment is not expected for actor nodes. The authors discuss single actor and multi actor networks where the number of actuating devices will be strongly dependent on the specific application and the environment conditions.

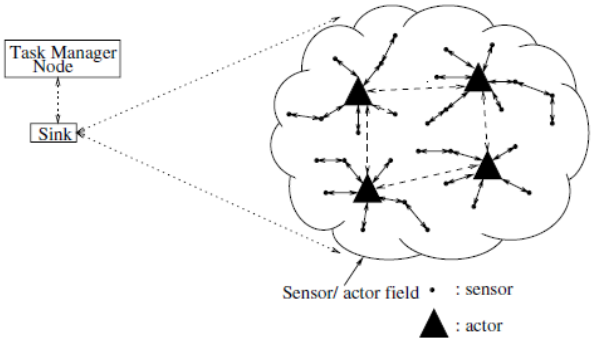


Fig. 7. The physical architecture of WSANs (Akyildiz & Kasimoglu, 2004).

WSAN have two unique features, which clearly differentiate them from WSNs: real time requirement and coordination. The real time requirement comes from the fact that WSAN are expected to immediately respond to a certain event i.e. in case of forest fire actions should be initiated immediately in order to reduce scale of damage. The coordination requirement has two aspects: one provides transmission of the event features from the

sensors to the actor nodes while the other is related to the coordination among the actor nodes themselves and the optimization of their actions.
In the survey the authors present a very detailed analysis of the specifics, requirements and open research issues related to WSAN. Together with the structure and functionalities of the future WSAN networks the authors discuss the questions of protocol design for these networks and its relation to cross-layer design. Akyildiz et al. argue that the presence of actor nodes makes protocol design even more complicated as additional operational issues like efficient communication between sensors and actors and effective coordination between actors in a multi actor network make the restrictions stricter and even protocols suitable for WSNs might be rendered insufficient They suggest a new protocol model for WSAN that is three dimensional and inherently cross-layered (Fig. 8).

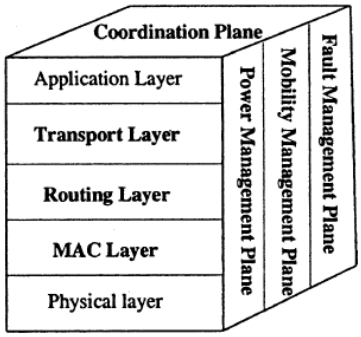


Fig. 8. WSAN protocols stack (Akyildiz & Kasimoglu, 2004).

The suggested model consists of three planes: communication plane, management plane and coordination plane. The communication plane is responsible for realizing the communication between the nodes. The data received by a node at the communication plane is submitted to the coordination plane to decide how the node should react to this data. The management plane in turn is responsible for monitoring the operation of the network and controlling the sensor and actor nodes. Important issues as mobility management, power management and fault tolerance are handled by the management plane. The coordination plane is more related to the actor nodes as they have to collaborate very efficiently with each other in order to perform a certain task, working sequentially or concurrently. It is stated that the realization of WSANs will need to satisfy more severe constraints and specific requirements introduced by the coexistence of sensor and actor nodes. A major research issue is the definition of a framework to characterize the protocol design and the suggested planes. The authors also stress on the fact that the cross-layer approach is the way to provide effective sensing, data transmission and acting.

6. Conclusion

In this chapter we have tried to discuss and summarize different issues related to cross-layer design, the new unconventional protocol design approach that has been suggested to meet the challenges and restrictions posed by the newly emerging networks like WSN and WSAN. These networks are based on small but intelligent devices (smart sensor nodes) that

can sense the environment, collect data and transfer data, if necessary react to a specific event. Furthermore the operation of the network is realized as a result of the collaborative action of large numbers (few tens to thousands) of nodes. Such networks behave quite differently from the traditional IP networks: first because of the inherently unstable and unpredictable nature of the wireless channel through which the multi-hop communication is realized, second due to the great limitations of the nodes in both capacity and power and third, due to the fact that they are highly application-centric and rely on the collaborative operational model to realize a specific task. Thus, unlike conventional networks they have their own design and resource constraints. Resource constrains include the limited amount of energy available to the nodes the short communication range, the low bandwidth and very limited storage and processing. Design constraints are based on the application and may vary as the applications themselves vary from environment monitoring to health care and event detection and tracking. Furthermore, WSAN introduce questions of coordination between actors and sensors.

Numerous studies have proved that the traditional layered protocol design approach (the OSI model) is not suitable to meet these constraints and specifics. Many researchers argue that a new holistic approach is required. In this line a number of cross-layer solutions, that allow interaction between protocols at different layers have been suggested and proved to be more suitable to the protocol design for WSNs. Benefiting from the interaction between different layer higher efficiency and prolonged network lifetime can be achieved. However the advocates of cross-layer design argue that such approaches are very dangerous as they damage the modularity of the design and can result in a number of unforeseen and unwanted effects.

In this chapter we have discussed the definition of cross-layer design approach, the suggested methods and classifications in the existing literature involving cross-layer interactions as well as the problems and challenges involved. Furthermore we have explained the necessity for creating a conceptual structure for protocol design that will suit the requirements and restrictions of WSNs. A review of the few suggested so far CLD frameworks, including the TinyCubus, DMA-CLD, OAB and XLM frameworks was given. By defining criteria for their evaluation we have contrasted and compared these suggestions. The chapter was concluded with a look towards the future: from wireless sensor networks and cross-layer design issues to the "smart environments" realized by wireless sensor and actor networks.

Finally we hope that this work will throw additional light on issues related to the cross-layer design and CLD frameworks and provide a background for a future unified approach to protocol design in WSN and WSAN that researchers may want to address as they move forward.

7. References

Akyildiz, I. F.; Su, W.; Sankarasubramaniam. Y. & Cayirci, E. (2002). A Survey on Sensor Networks. *IEEE Communications Magazine,* Vol. 40, No. 8, (August 2002), (102-116), ISSN: 0163-6804

Akyildiz, I. F. & Kasimoglu, I. (2004). Wireless sensor and actor networks: research challenges. *Ad Hoc Networks,* Vol. 2, No. 4, (October 2004), (351-367), ISSN: 1570-8705

Akyildiz, I. F.; Vuran, M. C. & Akan, O. B. (2006). A Cross-Layer Protocol for Wireless Sensor Networks. *Proceedings of Conference on Information Sciences and Systems,* pp. 1102 - 1107, ISBN 1-4244-0349-9, Princeton, NJ, March 2006, Information Sciences and Systems (CISS), Princeton

Alonso, L.; Ferrus, R. & Agusti, R. (2003). MAC-PHY enhancement for 802.11b WLAN systems via cross-layering. *Proceedings of IEEE VTC-Fall,* pp. 776 - 780, ISSN : 1090-3038, Orlando, FL, October 2003

Carpenter, B. (1996). Architectural Principles of the Internet, RFC 1958, June 1996. [Online]. Available: http:/www.rfc-editor.org/rfc/rfc1958.txt

CRUISE, (2007). European IST project CRUISE, Deliverable no.:D112.1, *Report on WSN applications, their requirements, application-specific WSN issues and evaluation metrics*, IST-027738/ CRUISE

Cui, S.; Madan, R.; Goldsmith, A. & Lall, S. (2005). Joint routing, MAC, and link layer optimization in sensor networks with energy constraints, *Proceedings of IEEE ICC 2005,* pp. 725 - 729, ISBN 0-7803-8938-7, May 2005

Fang, Y. & McDonald, A. B. (2004). Dynamic codeword routing (DCR): a cross-layer approach for performance enhancement of general multi-hop wireless routing, *Proceedings of IEEE SECON 2004*, pp. 255 - 263, ISBN 0-7803-8796-1, October 2004

Hakala, I. & Tikkakoski, M. (2006). From vertical to horizontal architecture - a cross-layer implementation in a sensor network node. *Proceedings of First International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense),* Article No:6, ISBN:1-59593-427-8, Nice, France, May 2006

Kawadia, V., & Kumar, P. R. (2005). A cautionary perspective on cross-layer design. *IEEE Wireless Communications Magazine*, Vol. 12, No. 1, (February 2005), (3-11), ISSN: 1536-1284.

Lachenmann, A., Marron, P.J., Minder, D., & Rothermel, K., (2005). An Analysis of Cross-Layer Interactions in Sensor Network Applications, *Proceedings of the Second International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 121-126, ISBN: 0-7803-9399-6, December 2005.

Lee, L. T. (2006). Cross-layer design and optimization for wireless sensor Networks. *MSc Thesis, Naval Postgraduate School,* (March 2006), Monterey California

Liu, Q.; Zhou, S. & Giannakis, G. B. (2004). Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links. *IEEE Transactions on Wireless Communications,* Vol. 3, No. 5, (September 2004), (1746 - 1755), ISSN: 1536-1276

Marrón, P. J.; Lachenmann, A.; Minder, D.; Hähner, J.; Sauter, R. & Rothermel, K. (2005a). TinyCubus: A Flexible and Adaptive Framework for Sensor Networks, *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*, pp. 278-289, ISBN 0-7803-8801-1, February 2005

Marrón, P. J.; Minder, D.; Lachenmann, A. & Rothermel, K. (2005b). TinyCubus: A Flexible and Adaptive Cross-Layer Framework for Sensor Networks. *4. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", Technical Report TR 481, Computer Science Department,,* (March 2005), (49 - 54), Zurich, Switzerland

Marrón, P. J.; Saukh, O.; Krüger, M.; & Große, C. (2005c). Sensor Network Issues in the Sustainable Bridges Project, *Proceedings of the European Projects Session of the Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, January 2005

Marrón, P. J.; Minder, D.; Lachenmann, A. & Rothermel, K. (2005d). TinyCubus: An Adaptive Cross-Layer Framework for Sensor Networks. *it - Information Technology,* Vol. 47, No. 2, (2005), (87 - 97), ISSN: 18125638

Melodia, T.; Vuran, M. C. & Pompili, D. (2006). The State-of-the-art in Cross-layer Design for Wireless Sensor Networks, *Lecture Notes in Computer Science (LNCS)*, Vol. 3883/2006, (May 2006), (78–92), ISSN 0302-9743

Morsink, P.; Hallouzi, R.; Dagli, I.; Cseh, C.; Schafers, L.; Nelisse, M.; & Bruin, D. D. (2003). Cartalk 2000: Development of a cooperative adas based on vehicle-to-vehicle communication, *Proceedings of the 10th World Congress on Intelligent Transport Systems and Services (ITS)*, Madrid, Spain, November 2003.

Raisinghani, V. T. & Iyer, S. (2004). Cross-layer design optimization in wireless protocol stacks, *Computer Communication*, Vol. 27, No.8, (May 2004), (720—724), ISSN: 0140-3664.

Safwat, A. M. (2004). A novel framework for cross-layer design in wireless ad-hoc and sensor networks, *Proceedings of IEEE GlobeCom Workshops*, pp. 130-135, ISBN: 0-7803-8798-8, December 2005

Shakkottai, S.; Rappaport, T. S. & Karlsson, P. C. (2003). Cross-layer design for wireless networks. *IEEE Communications Magazine,* Vol. 41, No. 10, (October 2003), (74 - 80), ISSN: 0163-6804

Sichitiu, M. L. (2004). Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks, *Proceedings of IEEE INFOCOM 2004*, pp. 1740 - 1750, ISSN: 0743-166X, March 2004

Sokullu, R., & Karaca, L. O., (2009). Simple and Efficient Cross-Layer Framework Concept for Wireless Sensor Networks, *Proceedings of 12th International Symposium on Wireless Personal Multimedia Communications*, pp.40, Sendai, Japan, September 2009.

Srivastava, V. & Motani, M. (2005). Cross-layer design: A survey and the road ahead. *IEEE Communications Magazine,* Vol. 43, No. 12, (December 2005), (112 - 119), ISSN: 0163-6804

Stallings, W. (2006). *Data and Computer Communications,* Prentice Hall, ISBN 13: 9780132433105

Tian, J. & Coletti, L. (2003). Routing approach in CarTALK 2000 project, *Proceedings of 12th IST Mobile and Wireless Communications Summit 2003*, Paper No. 1047, Aveiro, Portugal, June 2003

van Hoesel, L.; Nieberg, T.; Wu, J. & Havinga, J. M. (2004). Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Communications,* Vol. 11, No. 6, (December 2004), (78-86), ISSN: 1536-1284

Vuran, M. C.; Gungor, V. B. & Akan, O. B. (2005). On the interdependency of congestion and contention in wireless sensor networks, *Proceedings of SENMETRICS '05*, pp.136 – 147, San Diego, CA, July 2005

Wang, Q., & Abu-Rgheff, M. A. (2003). Cross-Layer Signalling for Next-Generation Wireless Systems. *Proceedings of IEEE Wireless Communications and Networking Conference 2003 (IEEE WCNC 2003*, pp. 1084-1089, ISSN : 1525-3511, New Orleans, LA, USA, March 2003

Yick, J., Mukherjee, B., & Ghosal, D., (2008). Wireless sensor network survey, *Computer Networks,* Vol.52, No.12, (August 2008), (2292-2330), ISSN: 1389-1286.

Zhang, Y., & Cheng, L., (2003). Cross-Layer Optimization for Sensor Networks, *Proceedings of 3 rd New York Metro Area Networking Workshop*, New York City, September 2003.

Zhang, Q., Zhang, Y. (2008). Cross-Layer Design for QoS Support in Multi-hop Wireless Networks. *Proceedings of IEEE 2008*, pp.64-76, ISSN : 0018-9219, January 2008.

Zhao, N. & Sun, L. (2007). Research on Cross-Layer Frameworks Design in Wireless Sensor Networks, *Proceedings of the Third International Conference on Wireless and Mobile Communications ICWMC '07*, pp. 50a - 50a, ISBN 0-7695-2796-5, Guadeloupe, March 2007.

4

Artificial Intelligence for Wireless Sensor Networks Enhancement

Alcides Montoya[1], Diana Carolina Restrepo[2]
and Demetrio Arturo Ovalle[2]
[1]*Physics Department,*
[2]*Computer Science Department, National University of Colombia - Campus Medellin*
Colombia

1. Introduction

Whereas the main objective of Artificial Intelligence is to develop systems that emulate the intellectual and interaction abilities of a human being the Distributed Artificial Intelligence pursues the same objective but focusing on human being societies (O'Hare et al., 2006). A paradigm in current use for the development of Distributed Artificial Intelligence is based on the notion of multi-agent systems. A multi-agent system is formed by a number of interacting intelligent systems called agents, and can be implemented as a software program, as a dedicated computer, or as a robot (Russell & Norving, 2003). Intelligent agents in a multi-agent system interact among each other to organize their structure, assign tasks, and interchange knowledge.

Concepts related to multi-agent systems, artificial societies, and simulated organizations, create a new and rising paradigm in computing which involves issues as cooperation and competition, coordination, collaboration, communication and language protocols, negotiation, consensus development, conflict detection and resolution, collective intelligence activities conducted by agents (e.g. problem resolution, planning, learning, and decision making in a distributed manner), cognitive multiple intelligence activities, social and dynamic structuring, decentralized administration and control, safety, reliability, and robustness (service quality parameters).

Distributed intelligent sensor networks can be seen from the perspective of a system composed by multiple agents (sensor nodes), with sensors working among themselves and forming a collective system which function is to collect data from physical variables of systems. Thus, sensor networks can be seen as multi-agent systems or as artificial organized societies that can perceive their environment through sensors.

But, the question is how to implement Artificial Intelligence mechanisms within Wireless Sensor Networks (WSNs)? There are two possible approaches to the problem: according to the first approach, designers have in mind the global objective to be accomplished and design both, the agents and the interaction mechanism of the multi-agent system. In the second approach, the designer conceives and constructs a set of self-interested agents whose then evolve and interact in a stable manner, in their structure, through evolutionary techniques for learning. The same difficulty applies when working with a WSN perspective seen from the

perspective of DAI. Can the principles, algorithms and application of Distributed Artificial Intelligence be used to optimize a network of distributed wireless sensors? Is it possible to implement a solution that enables a sensor network to behave as an intelligent multi-agent system? From a perspective of multi-agents, artificial societies, and simulated organizations, how must a distributed sensor network be installed in an efficient manner and achieve the proposed objectives of taking measures of physical variables by itself? What are the union points between Distributed Artificial Intelligence and Wireless sensor networks? The fundamental idea is this chapter is to propose a model that enables a highly distributed sensor network to behave intelligently as a multi-agent system.

2. Wireless Sensor Networks

A Sensor Network (SN) is a system that consists of thousands of very small stations called sensor nodes. The main function of sensor nodes it is to monitor, record and notify a specific condition at various locations to other stations. Also, a SN is a group of specialized transducers with a communications infrastructure intended to monitor and record conditions at diverse locations. Commonly monitored parameters are temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, chemical concentrations, pollutant levels and vital body functions.
Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. Although these devices have a very little capability on their own they have substantial processing capabilities when they are working as an aggregate, (CRULLER et al., 2004). Each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust (Romer & Mattern, 2004). A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station). It is important to underline that SNs are subject to more severe power constraints than PDAs, mobile phones, or laptops. The whole network is usually under the administration of one controller: the base station. The main functionality of the base station is to act as gateway to another network, and is a powerful data processor and storage center. Advances in microelectronics and wireless communications have made WSNs the predict panacea for attacking a host of large-scale decision and information processing tasks. The applications for WSNs are varied, typically involving some kind of monitoring, tracking, or controlling. Specific applications include habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes. A number of WSNs have been deployed for environmental monitoring (Davoudani et al., 2007). Many of these have been short lived, often due to the prototype nature of the projects. Wireless sensor networks have been developed for machinery Condition-Based Maintenance (CBM) since they offer significant cost savings and enable new functionalities.
Although a number of new WSN systems and technologies have been developed, a number of new problems or challenges are yet to be solved or improved on. Examples of such problems are optimal routing strategies, lifespan of the WSN, lifetime of the nodes are often very limited, reconfigurability without redeployment, etc.
Finally, since WSNs become popular there is not a common platform. Some representative designs have broader users and developer communities, such as Berkeley Motes, which was

the first commercial motes platform. However, many research labs and commercial companies prefer to develop and produce their own devices since a sensor node is a processing unit with basic components. Some platforms are: Mica Mote (http://www.xbow.com), Tmote Sky (http://www.moteiv.com), BTnode(http://www.btnode.ethz.ch/), Waspmote(http://www.libelium.com/products/waspmote), Sun Spot(http://www.sunspotworld.com/SPOTManager/), G-Node (http://sownet.nl/index.php/en/products/gnode), TIP series mote (http://www.maxfor.co.kr/), among others.

3. Artificial Intelligence and Multi-Agent Systems

Classical Artificial Intelligence aimed at emulating within computers the intellectual and interaction abilities of a human being. The modern approach to Artificial Intelligence (AI) is centered around the concept of a rational agent. An agent is anything that can perceive its environment through sensors and act upon that environment through actuators (Russell & Norving, 2003).An agent that always tries to optimize an appropriate performance measure is called a rational agent. Such a definition of a rational agent is fairly general and can include human agents (having eyes as sensors, hands as actuators), robotic agents (having cameras as sensors, wheels as actuators), or software agents (having a graphical user interface as sensor and as actuator). From this perspective, AI can be regarded as the study of the principles and design of artificial rational agents.
However, agents are seldom stand-alone systems. In many situations they coexist and interact with other agents in several different ways. Examples include intelligent Web software agents, soccer playing robots, e-commerce negotiating agents, computer vision dedicated agents, and many more. Such a system that consists of a group of agents that can potentially interact with each other is called a Multi-Agent Systems (MAS), and the corresponding subfield of AI that deals with principles and design of multi-agent systems is called Distributed AI (DAI).

4. Wireless Sensor Networks and Artificial Intelligence

An intelligent sensor is one that modifies its internal behavior to optimize its ability to collect data from the physical world and communicates it in a responsive manner, to a base station or to a host system. The functionality of intelligent sensor includes: self-calibration, self-validation, and compensation. The self-calibration means that the sensor can monitor the measuring condition to decide whether a new calibration is needed or not. Self-validation applies mathematical modeling error propagation and error isolation or knowledge-based techniques. The self-compensation makes use of compensation methods to achieve a high accuracy. The types of artificial intelligence techniques widely used in industries are: Artificial Neural Network (ANN), Fuzzy Logic and Neuro-Fuzzy. Intelligent sensor structures embedded in Wireless Sensor Networks result in wireless intelligent sensors. The use of Artificial intelligence techniques plays a key role in building intelligent sensor structures. Main research issues of the WSNs are focused on the coverage, connectivity network lifetime, and data fidelity. In the recent years, there has been an increasing interest in the area of the Artificial Intelligence and Distributed Artificial Intelligence and their methods for solving WSNs constrains, create new algorithms and new applications for WSNs. Resource management is an essential ingredient of a middleware solution for WSN. Resource management includes initial sensor-selection and task allocation as well as runtime adaptation of allocated task/resources. The parameters to be optimized include energy, bandwidth, and network lifetime. In this par-

ticular case Distributed Independent Reinforcement Learning proposed the use of collective intelligence in resource management within WSNs (Shah et al., 2008). Finally, intelligent networking and collaborative systems are also proposed as components for WSNs' enhancement.

5. Multi-Agent Based Simulation

MABS refers to the simulation aim at modeling the behavior of agents in order to analyze their interactions and consequences of their decision making process. Hence, a global result is closely determined by agents' interactions. In practice, MABS models are used to represent and understand social systems (Conte et al., 1998), moreover to evaluate new strategies of improvement and politics on different kind of systems. Due to MABS is a recently area, there are actually few techniques and tools for its development. In fact, some contributions come from system simulation, software engineering and agent-oriented software engineering (AOSE). Facing this constrain, a methodology was proposed by GIDIA research group from National University of Colombia, which defines several stages and artifacts for every phase of a software lifecycle (Moreno et al., 2009). This methodology allows the representation of main characteristics of the distributed system, including key aspects such as organization, reasoning, communication, and coordination mechanism, among others. The main function of WSN simulators is to emulate a WSN operation and simulate entire characteristics of hardware for each node in simulated WSN, instead of providing strategies to do a deployment. The fundamental idea is to propose a model that enables a highly distributed sensor network to behave intelligently as a multi-agent system. It is important to note that most simulators are used to simulate a specific system, be a MAS or a WSN, but not both of them. Besides, it is needed to identify the relationships existing between agents and sensor nodes for getting intelligence from the multi-agent system and monitoring from the WSN. From WSNs' point of view, MABS provides understanding on WSNŠs performance and network autonomous capabilities when acting as an agents society. In this case, agents collaborate together to save and improve resources within the WSN. Finally, MABS can highly contribute to define deployment strategies and operation politics related to the simulated application.

6. Multi-agent Model proposal

Model proposal is a Multi-Agent hybrid model to simulate the deployment of software agents over any WSN, this is done by a layered architecture that utilizes deterministic models of hardware with agent based intelligence, in order to evaluate different strategies, such as different agents for a specific application. It utilizes mobile agents to control network resources and facilitate intelligence. In order to get this, it is used principal deterministic models for WSN performing, such as, protocol model, which comprises all the communication protocols and their operation usually depends on the state of the physical platform of nodes, physical model, which represents the underlying hardware and measurement devices, media model, which links the node to the "real world" through a radio channel and one or more physical channels, battery model that is responsible for checking if the node has exhausted its battery through computing power consumption of the different components, among others (Egea-Lopez et al., 2006). Moreover, it is added the topology and physical variables according to the application that is going to be simulated. Then, it is used software agents to perform all tasks required by the application study case.

6.1 Simulation Models for WSN

Present simulation models try to represent how a WSN works. For example, Egea-Lopez at al., in Egea-Lopez et al. (2006) have proposed a general simulation model taking into account current components of a WSN simulator. Hence, there are several deterministic models to represent hardware, environment, power, radio channels, among others. These models are useful in the way of knowing about how a WSN performs in a real life but they do not offer the potential of evaluating different strategies of deployment, moreover, the simulation nodes number is really far of a real network, due to scalability is affected by all required processing to simulate complete hardware.

Later, a new propose is presented by Cheong in Cheong (2007). Some strengths of this work are the use of different simulation tools whose are already defined for WSN Levis et al. (2003), and it permits a directed implementation from simulation. However, Cheong proposes a programming paradigm based on actors, whose are a concept between objects and agents. Actors are objects with data flow for communication, but they are not aware of its environment neither able to take decisions for acting.

Another approach is presented by Wang and Jiang in Wang et al. (2006), where is presented a strategy to control and optimize resources in a WSN through mobile agents. Optimization of resources such as, power, processing and memory of devices is done, but it is not defined how devices and agents are related for getting this optimization.

6.2 Model Proposal

It is proposed a Multi-Agent hybrid model to simulate the deployment of software agents over any WSN, this is done by a layered architecture that uses deterministic models of hardware with agent based intelligence, in order to evaluate different strategies, such as different agents for a specific application.

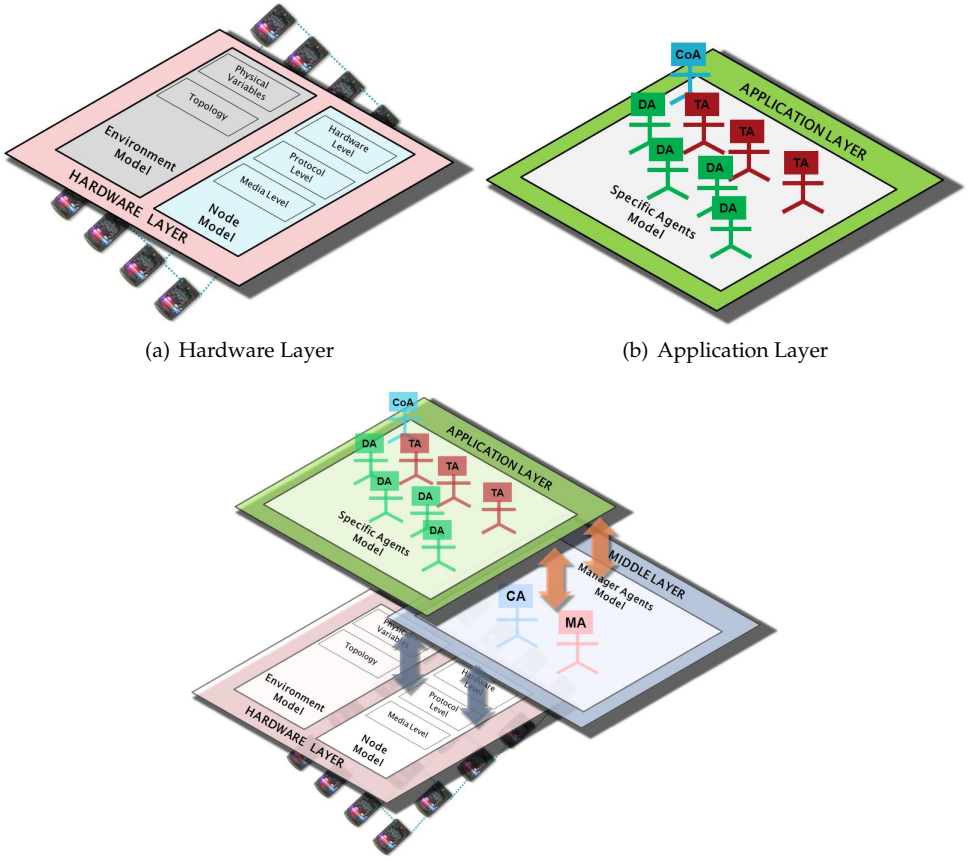
We aim to utilize mobile agents to control network resources and facilitate intelligence. In order to get this, it is used the principal deterministic models specified by Egea-Lopez et al. (2006), these models set features, such as, platform of nodes, power consumption, radio channel and media. Moreover, it is added the topology and physical variables according to the application that is going to be simulated. Finally, it is used software agents to perform all tasks required by the application study case. Below is presented three different layers that let to perform intelligence through agents over a WSN.

6.2.1 Hardware Layer

The hardware layer is responsible to specify all components that are related to characteristics provided by hardware and the environment where network is going to be deployed. Most models of this layer are already defined by the present WSN simulators. Below it is introduced some models that specify these components.

- Node Model: This model has been specified before by Egea-Lopez et al. (2006), where a node is divided by protocols, hardware and media. Protocols operation depends on hardware specifications and comprises all communications protocols of a node. Hardware represents the underlying platform and measurement devices. And media, links the node to the Şreal worldŤ through a radio channel and one or more physical channels, connected to the environment component.
- Environment Model: This model includes principal variables of physical area where the network is going to be deployed. The sensors of a node have to be able to sense these variables otherwise the agents of higher layers will not be executed. Besides, this

model specifies the topology, i.e. the structure of how the nodes are organized, there are different topologies to a WSN such as square, star, ad-hoc, irregular Piedrahita et al. (2010).



(a) Hardware Layer

(b) Application Layer

(c) All Layers

Fig. 1. Hardware, Application Layers and Complete Model Proposal

6.2.2 Middle Layer

The middle layer is responsible to attach a WSN with the needed agents for a specific application. Hence this layer has two agents that perform control and resources manage.

- Manager resources Agent (MA): It is a specialized mobile agent that takes decisions about controlling resources of memory and power. It is aware of required charge for an agent performs a task, and denies or admits to execute an agent. This is an agent that takes decisions based on a BDI model Georgeff et al. (1998). Moreover, it says if a group of tasks can be executed in keeping with the specified hardware.

- Capturing Agent of physical variables (CA): It is a mobile agent that is aware of physical variables according to a specific application. It takes decisions about propagation and transmitting of these variables.

6.2.3 Application Layer

The application layer represents specific study case or application for which the WSN is going to be deployed. Therefore this layer has agents that perform application required tasks.

- Coordinator Agent (CoA): It is an agent aware of required tasks by a study case so it has a queue of application tasks. Hence, it manages, organizes and negotiates them, for being executed by a TA successfully. Also, it takes decisions based on a BDI model.
- Tasks Agent (TA): It is a reactive agent that performs tasks assigned by a CoA, as long as CoA said it had to be.
- Deliberative Agent (DA): It is a mobile agent that takes decisions based on a BDI model too. It does not need that a CoA manages, organizes and negotiates its tasks, it does by its own. Accordingly, it performs a set of tasks to achieve its own goal or a goal established by a MAS which it belongs to.

It is a specific treatment for an application multi-agent system, due to not all sensor nodes platforms can perform a rational agent i.e. for a simple application there is a group of TA with a CoA that manages and coordinates entire system, and for a complex application there is a group of DA that interact to achieve a global goal.

6.3 Interaction Process

First of all, the CoA(or a DA, depending of required type agents) starts the process for assigning a task, it has the belief that a task needs to be done, it has this belief because there is a tasks list related to the application. Its desire consist of ensure that a task is done successfully by a TA. Then, its first intention is to interact with MA and to ask task feasibility.
Now, MA beliefs about its hardware characteristics and charge task, and its desire consist to inform if there are enough resources to do the task, for this reason its intention is reasoning if charge task processing fits on available resources. It informs true or false.
If MA answer is true, CoA second intention is to create an instance of a TA, and assign this task. Finally, its last intention is to be sure that the task was done then it asks to TA, if it is done and depending on this answer it starts with another task or the same.
In the case of DA multi-agent system any DA starts the interaction process with agents in the middle layer. MA beliefs about its hardware characteristics and charge on a plan (task group). If MA confirms available resources, the DA starts its process, otherwise it waits until get an affirmation from MA.
Taking into account above process, we introduce some theoretical formula to determinate global battery discharge (see Equation 1 and 2) and memory usage (see Equation 3 and 4), for a time period in the simulation.

$$B_{(t)} = B_{(t-1)} - P(CoA)(MA) - P(TA)L_{(t-1)} \tag{1}$$

$$B_{(t)} = B_{(t-1)} - P(DA)(MA) - P(DA)L_{(t-1)} \tag{2}$$

Where $B_{(t)}$ is the battery state at time t, $P(CoA)(MA)$ and $P(TA)$ are the processing of CoA and MA agents and TA agent respectively and $L_{(t-1)}$ is the task charge. For equation 2 $P(DA)$

and $P(MA)$ are the processing of DA and MA agents and $L_{(t-1)}$ is the plan charge. These tasks and plans are negotiated in a specified order, and constantly repeating.
For Memory usage ($M_{(t)}$), the formula required to perform or not a task or a plan,

$$M_{(t)} = M_{(t-1)} - P(CoA)(MA) - P(TA)L_{(t-1)} + P(TA)L_{(t-2)} \tag{3}$$

$$M_{(t)} = M_{(t-1)} - P(DA)(MA) - P(DA)L_{(t-1)} + P(DA)L_{(t-2)} \tag{4}$$

7. Conclusions and future work

The principles, algorithms and application of Distributed Artificial Intelligence can be used to optimize a network of distributed wireless sensors. The Multi-Agent System approach permits WSN optimization using rational agents to get this achievement.
It is possible to implement a solution that enables a sensor network to behave as an intelligent multi-agent system through the proposed model due to it utilizes multi-agent systems together with layered architecture to facilitate intelligence and simulate any WSN, all needed is to know the final application, where the WSN is going to be deploy. Also, a layered architecture can provide modularity and structure for a WSN system. Moreover, proposed model emphasizes about how a WSN works and how to make it intelligent.
From a perspective of multi-agents, artificial societies and simulated organizations, a distributed sensor network can be installed in an efficient manner and achieve the proposed objectives of taking measures of physical variables by itself with different types of rational agents that can be reconfigured to fit any kind of application and measures, also to fit the most appropriate strategy to achieve requirements of physical variables monitoring.
Further work to do is testing model using a real WSN. Some study cases of multi-agent systems for specific applications are required to do a complete testing. A useful tool to use is the Solarium SunSPOT emulator. This emulator makes available a realistic testing to develop and test SunSPOT devices without requiring hardware platform. After this testing finishes, the model could be performed over a real WSN of SunSPOT devices.

8. Acknowledgments

This work presents the results of the researches carried out by GIDIA (Artificial Intelligence Research & Development Group) and GICEI (Scientific & Industrial Instrumentation Research Group) at the National University of Colombia - Campus Medellin, as advance of two research projects co-sponsored by DIME (Research Direction of National University of Colombia at Medellin Campus) and COLCIENCIAS (Colombian Institute of Science and Technology) respectively entitled:"Intelligent Hybrid System Model for Monitoring of Physical variables using WSN and Multi-Agent Systems" with code 20201007312 and "Development of a model of intelligent hybrid system for monitoring and remote control of physical variables using distributed wireless sensor networks" with code 20201007027.

9. References

Cheong, E. (2007). Actor-oriented programming for wireless sensor networks.
Conte, R., Gilbert, N. & Sichman, J. (1998). MAS and social simulation: A suitable commitment, *Multi-Agent Systems and Agent-Based Simulation*, Springer, pp. 1–9.

CRULLER, D., Estrin, D. & Srivastava, M. (2004). Overview of sensor networks, *Computer* **37**(8): 41–49.

Davoudani, D., Hart, E. & Paechter, B. (2007). An immune-inspired approach to speckled computing, *Artificial Immune Systems* pp. 288–299.

Egea-Lopez, E., Vales-Alonso, J., Martinez-Sala, A., Pavon-Marino, P. & Garcia-Haro, J. (2006). Simulation scalability issues in wireless sensor networks, *IEEE Communications Magazine* **44**(7): 64.

Georgeff, M., Pell, B., Pollack, M., Tambe, M. & Wooldridge, M. (1998). The belief-desire-intention model of agency, *Intelligent Agents V. Agent Theories, Architectures, and Languages: 5th International Workshop, ATAL'98, Paris, France, July 1998. Proceedings*, Springer, pp. 630–630.

Levis, P., Lee, N., Welsh, M. & Culler, D. (2003). TOSSIM: Accurate and scalable simulation of entire TinyOS applications, *Proceedings of the 1st international conference on Embedded networked sensor systems*, ACM, p. 137.

Moreno, J., Velásquez, J. & Ovalle, D. (2009). Una Aproximación Metodológica para la Construcción de Modelos de Simulación Basados en el Paradigma Multi-Agente, *Avances en Sistemas e Informática* **4**(2).

O'Hare, G., O'Grady, M. & Marsh, D. (2006). Autonomic wireless sensor networks: Intelligent ubiquitous sensing, *proceeding of ANIPLA 2006, International Congress on Methodologies for Emerging Technologies in Automation*, Publisher, University La Sapienza, Rome, Italy.

Piedrahita, A., Montoya, A. & Ovalle, D. (2010). Performance Evaluation of an Intelligent Agents-based Model in WSN with irregular topologies.

Romer, K. & Mattern, F. (2004). The design space of wireless sensor networks, *IEEE Wireless Communications* **11**(6): 54–61.

Russell, S. & Norving, P. (2003). *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs,.

Shah, K., Kumar, M., Inc, S. & Addison, T. (2008). Resource management in wireless sensor networks using collective intelligence, *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008*, pp. 423–428.

Wang, X., Wang, S. & Jiang, A. (2006). Optimized deployment strategy of mobile agents in wireless sensor networks, *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, Vol. 2.

Part 2

Network protocols, architectures and technologies

5

Broadcast protocols for wireless sensor networks

Ruiqin Zhao, Xiaohong Shen and Xiaomin Zhang
Northwestern Polytechnical University
P.R.China

1. Introduction

Future network is all about an integrated global network based on an open-systems approach. Integrating different types of wireless networks with wireline backbone networks seamlessly and the convergence of voice, multimedia, and data traffic over a single IP-based core network will be the main focus of 4G. With the availability of ultrahigh bandwidth of up to 100 Mbps, multimedia services can be supported efficiently. Ubiquitous computing is enabled with enhanced system mobility and portability support, and location-based services and support of ad hoc networking are expected. Fig. 1 illustrates the networks and components within the future network architecture. It integrates different network topologies and platforms. There are two levels of integration: the first is the integration of heterogeneous wireless networks with varying transmission characteristics such as wireless LAN (Local Area Network), WAN (Wide Area Network), and PAN (Personal Area Network) as well as mobile ad hoc networks; the second level includes the integration of wireless networks and fixed network-backbone infrastructure, the Internet and PSTN (Public Switched Telephone Network).

Recent advancement in wireless communications and electronics has enabled the development of low-cost sensor networks. WSN are composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. A wireless sensor network can be used in a wide variety of commercial and military applications such as inventory managing, disaster areas monitoring, patient assisting, and target tracking.

The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source. The issue of energy-efficient communication in WSN has been attracting attention of many researches during last several years. Broadcasting is a common operation that allows the node in WSN to share its data efficiently among each other. Broadcasting can be used for network discovery to initiate the configuration of the network, to discover multiple routes between a given pair of nodes, and to query for a piece of desired data in a network (N. B. Chang & M. Liu, 2007). In wireless sensor networks, broadcasting can serve as an efficient solution for the sensors to share their local measurements among each other due to the robustness and the effectiveness of the protocol.

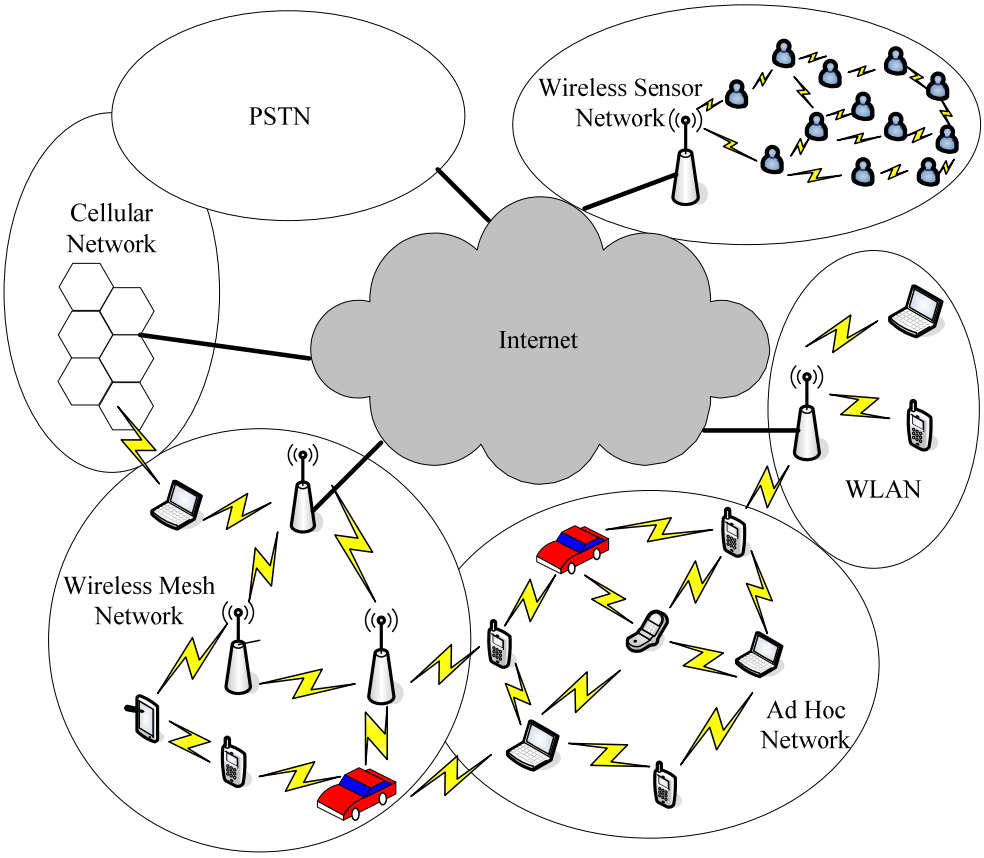


Fig. 1. Future network

The traditional way of broadcast in WSN is flooding, which is the straightforward and obvious way. When a source node has a packet to broadcast in the network, it sends the packet to all of its neighbors. Then each node that has received the packet for the first time will rebroadcast the packet to its neighborhood, which leads to the participation of all the nodes in broadcasting the packet. Thus, the traditional flooding which also is known as ordinary broadcast mechanism (OBM), results in serious redundancy, collision and contention, and referred to as broadcast storm problem (S Y Ni et al., 1999). The formation of the broadcast storm problem is due to the redundancy of rebroadcast which results in the serious contention and collision. Moreover, the reduction of the redundancy of rebroadcast is also the requirement of energy-saving in WSN. In networks where each node is assumed to have a fixed level of transmission power, less rebroadcasts means less energy consumed with the assumption that the energy needed by receiving is much less than the energy consumed by transmitting. To save as much energy as possible for each node in the network, the broadcast algorithm should make as less nodes as possible participate in the rebroadcast of the broadcasted message (R.Q. Zhao et al.,2007). Therefore, reduction of

rebroadcast redundancy is significant. A satisfying broadcast strategy should be able to reduce the broadcast redundancy effectively, not only for the saving of bandwidth, but also for the saving of energy, as both bandwidth and energy are valuable resources in WSN. While reduction of rebroadcast redundancy is not the only metric for a good broadcast protocol. There is another metric used for evaluating performance of broadcast protocols called reachability, which indicates the coverage rate of a broadcast algorithm.

With the aim of solving the broadcast storm problem and maximizing the network life-time, we propose an efficient broadcast algorithm—Maximum Life-time Localized Broadcast (ML2B) for WSN, which possesses the following properties:

a) ***Localized algorithm.***
   Localized algorithm is distributed algorithm which achieves a desired global objective with simple local behaviors. Each node makes the decision of rebroadcast based on its one-hop local information, e.g. its own position, its one-hop neighbors' information and energy left in its battery. Distributed design of broadcast routing is required by the essence of WSN. However, many proposed broadcast approaches were not distributed, such as those approaches selecting rebroadcast nodes based on a constructed broadcast tree which could not be maintained by each node using only its own local information. ML2B need not maintain any global topology information, thus resulting in much less overhead in WSN.

b) ***Energy-saving approach.***
   It is designed with the aim of minimizing energy required per broadcast task and maximizing network life-time. ML2B is not based on constructing a minimum energy tree which may cause much overhead to maintain the tree. It selects rebroadcast nodes by considering the coverage efficiency and the left energy of the node together to maximize life-time of the whole network. Using the rule of less rebroadcasts results less total energy consumed, ML2B cuts down the total energy consumption in broadcast routing by reducing the redundancy of rebroadcast largely which is capable of relieving the broadcast storm problem synchronously.

c) ***Degree adaptive broadcast strategy.***
   To reduce the redundancy of rebroadcast, nodes with large degree will be selected with higher priority as forward nodes in ML2B. The degree we use in this paper is the number of left neighbors that have not been covered by the former forward node or by the broadcast originator. Therefore, the rebroadcast of nodes with high degree brings high efficiency of the rebroadcast and great reduction of broadcast redundancy.

d) ***)Fault tolerant algorithm.***
   For the multi-path and fading effects of the wireless channel, or some sensor nodes may fail or be blocked due to physical damage or environmental interference, protocols used in WSN should be robust. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures. ML2B uses a self-selection mechanism to choose nodes that will rebroadcast next from nodes that were able to receive the packets without errors.

The remainder of this chapter is organized as follows. Firstly we make a survey of energy efficient broadcast protocols for wireless sensor networks in Sections 2. Secondly we propose an efficient broadcast protocol for WSN in Sections 3 and 4. It optimizes broadcasting by reducing redundant rebroadcasts and balancing the energy consumption among all nodes. Simulation is done in section 5 to verify the proposed mechanism.

Simulation results show that the proposed broadcast protocol can prolong the network life-time of WSN effectively. Finally, in Section 6 we draw the main conclusions.

2. Related Works

The straightforward way of broadcast is flooding. The advantage of flooding is its simplicity and reliability. However , for its large amount of redundant rebroadcast, flooding will cause serious packets collision, bandwidth waste , and battery energy exhaustion, which are referred to as broadcast storm problem (S Y Ni et al., 1999).

Various approaches have been proposed to solve the broadcast storm problem of flooding for wireless multi-hop networks. Some methods are designed with the aim of alleviating the broadcast storm problem by reducing redundant broadcasts. As in (J. Wu & F. Dai, 2004) ; (M. T. Sun &T. H. Lai, 2002); (W. Peng & X. C. Lu, 2000), each node computes a local cover set consisting of as less neighbors as possible to cover its whole 2-hop coverage area by exchanging connectivity information with neighbors. These methods require each node know its k-hop (k >=2) neighbor information. To maintain the fresh k-hop (k >=2) neighbor information, these broadcast methods result in heavy overhead on WSN, and they consume much energy at each node. Some methods (S Y Ni et al., 1999); (M. Lin et al., 1999) select forward node based on probability, which cannot guarantee the reachability of the broadcast.

Many proposed energy-saving broadcast methods are centralized, which require the topology information of the whole network. They try to find a broadcast tree such that the energy cost of the broadcast tree is minimized. Some methods(J.E. Wieselthier et al., 2000); (P.J. Wan et al., 2001); (M. Cagalj et al., 2002); (D. Li et al., 2004) are based on geometry or graph information of the network to compute the minimum energy tree.

Since the centralized method will cause much overhead in wireless sensor network, some localized versions of the above algorithms have been proposed recently. The algorithm in (M. Agarwal et al., 2004) reduces energy consumption by taking advantage of the physical layer design. (W.Z. Song et al., 2006) proposed a scheme for each node to find the network topology in a distributed way. However the algorithm proposed in (W.Z. Song et al., 2006), also requires each node to maintain the network topology, and the overhead is obviously more than a localized algorithm. The method proposed in (F. Ingelrest & D. Simplot-Ryl., 2005) requires that each node must be aware of the geometry information within its 2-hop neighborhood. It results in more control overhead and energy cost than the thorough distributed algorithm that requires only local one-hop information.

Two types of broadcasting protocols(J.-P. Sheu g et al., 2006) are proposed for wireless sensor networks. The two broadcasting protocols, are called one-to-all and all-to-all broadcasting protocols. And the protocols are proposed for five fixed and regular WSN topologies. An energy-saving broadcast method using cooperative transmission in WSN is proposed in (Y.-W. Hong & A. Scaglione, 2006). The cooperation is provided through a system called the Opportunistic Large Array (OLA) where network broadcasting is done through signal processing techniques at the physical layer. In (X. Hui et al., 2006), the practical models for power aware broadcast in wireless ad hoc and sensor networks are analyzed. Some literatures deal with the query execution in large sensor networks, e.g. (J.-P. Sheu et al., 2007); (C. R. Mann et al., 2007). These proposed protocols are designed to

facilitate any type queries for data content and services over a specific geographic region in large population, high-density wireless sensor networks. Several robust data delivery protocols (F. Ye et al., 2005); (Miklós Maróti, 2004) have been proposed for large sensor networks to disseminate data to interested sensors. GRAdient Broadcast (F. Ye et al., 2005) addresses the problem of robust data forwarding to a data collecting unit using unreliable sensor nodes with error-prone wireless channels. A Broadcast Protocol for Sensor networks (BPS) is proposed in (A. Durres i&V. Paruchuri, 2007). BPS uses the location of each node to broadcast packets in a distributed way.

3. System Model

The WSN can be abstracted as a graph $G(V,E)$, in which V is the set of all the nodes in the network and E consists of edges presented in the graph. An edge $e=(u,v)$, $e \in E$ exists if the Euclidean distance between node u and v is smaller than r, where r is the radius of the coverage of nodes. We assume all links in the graph is bidirectional, and the graph is in a connected state. Given a node i, time t is recorded since it receives the broadcasted message for the first time, and $t=0$. The energy left in battery of node i is represented by $e(i,t)$. $l(i,t)$ is defined as the Euclidean distance between node i and the up-link forward node $uf(i,t)$ which sends the broadcasted message.

We assume each node knows its own position information by means of GPS or other instruments. Each node also obtains its one-hop neighbors' information which is available in most location-aided routing (F. Ingelrest & D. Simplot-Ryl, 2005) of the ad hoc or sensor networks. Energy left in battery also needs to be provided at every node locally.

For $\forall i \in V$, several variables are defined as follows:

- Neighbor $nb(i)$, is the node that can communicate directly with node i. It is the one-hop neighbor of node i.
- Neighbor set $NB(i)$, is the set of all neighbors of node i.
- Uncovered set $UC(i,t)$, consists of one-hop neighbors that have not been covered by a certain forward node of the broadcasted message or the broadcast originator, before t.
- Degree $d(i,t)$, is the number of nodes belonging to $UC(i,t)$ at t. $d(i,t)$ implies the rebroadcast efficiency of node i. If $d(i,t)$ is below a threshold before its attempt to rebroadcast the broadcasted message, node i could abandon the attempt.
- Up-link forward node $uf(i,t)$, is the $nb(i)$ that rebroadcasts or broadcasts the message which is received by node i at t $(0 \le t \le D(i))$. Before $t > D(i)$, node i may receive several copies of the same broadcasted message from different up-link forward nodes($D(i)$ is the add delay of node i).
- Down-link forward node $df(i,t)$, is the $nb(i)$ that rebroadcasts the message at t $(t > D(i))$, after it has received the message from node i. If node i has not rebroadcasted the message at $t = D(i)$, it will not have any down-link forward node. That is to say, only the forward node has down-link forward node, though except for broadcast originator node each node owns up-link forward node.

- Up-link forward set $UF(i,t)$, is the set of all up-link forward nodes of node i before t. If it has received the same broadcasted message for k times before t $(t > D(i))$, its up-link forward set can be expressed as:

$$UF(i,t) = \{uf(i,t_0), uf(i,t_1), uf(i,t_2) \ldots uf(i,t_{k-1})\}, \ (k \geq 1) \tag{1}$$

(where $t_0, t_1, t_2 \ldots$, and t_{k-1} $(t_{k-1} \leq t)$ records the time node i received the 1st, 2nd, 3rd …, and k th copy of the same broadcasted message).

- Down-link forward set $DF(i,t)$, consists of all down-link forward nodes of node i before t. Nodes that have not been selected as forward node have an empty down-link forward set. While the down-link forward set of forward node i with k' down-link forward nodes is given as follows:

$$DF(i,t) = \begin{cases} \{df(i,t_0), df(i,t_1), df(i,t_2) \ldots df(i,t_{k'-1})\}, & k' \geq 1 \\ \varnothing, & k' = 0 \end{cases} \tag{2}$$

(where $k' = 0$ means no rebroadcast is initiated by the rebroadcast of node i).

4. Maximum Life-time Localized Broadcast (ML2B) Algorithm

4.1 Design for Add-Delay $D(i)$

Utilization of add-delay in broadcast protocols is to reduce the redundancy of nodes' rebroadcast and energy consumption. When node i receives a broadcasted message for the first time, it will not rebroadcast it as OBM. It delays a period of add-delay $D(i)$ before its attempt to do the rebroadcast. Even when $D(i)$ expires, the node will not rebroadcast it urgently until the node degree $d(i, D(i))$ is larger than the abandoning threshold n. During the period time of $0 \leq t < D(i)$, $\forall$ node i could abandon its attempt to rebroadcast the message as soon as its node degree $d(i,t)$ is equal to or below the threshold, thus reducing the rebroadcast redundancy and energy consumption largely.

Nodes with larger add-delay have a higher probability of receiving multiple copies of a certain broadcasted message, before their attempt to rebroadcast. Each reception of the same message decreases the node degree, thus making nodes with large add-delay rebroadcast the message with little probability. While nodes with little add-delay may rebroadcast the message quickly. We assign little add-delay or no-delay to nodes with high rebroadcast efficiency and enough left energy, large add-delay to nodes with large rebroadcast redundancy. To formulate the rebroadcast efficiency, two metrics are presented as follows:

$$f_d(i) = \frac{a - d(i,0)}{a}, \qquad (0 \leq f_d(i) \leq 1) \tag{3}$$

$$f_l(i) = \frac{r - l(i,0)}{r}, \qquad (0 \le f_l(i) \le 1) \tag{4}$$

Formula (3) is the node degree metric, and formula (4) is the distance metric. a is the maximum node degree, r is the radius of nodes' coverage. It can be induced from the two formulas that less $f_l(i)$ or $f_d(i)$ results in higher rebroadcast efficiency.

To maximize the network life-time, we present the third metric----energy metric for selecting proper rebroadcast nodes. If the left energy at a node is smaller than an energy threshold, it refuses to forward the broadcasted message. Otherwise, the node calculates the add-delay based on formula (5) where $E^{'}$ is the maximum energy when battery is full, and E_T is the energy threshold which is used to prevent nodes with little energy from dying. The selection of E_T 's value affects the performance of ML2B. Too large value will bring low redundancy, but may result in low reachability simultaneously. Too small value, on the other hand, could not prevent the premature crash of nodes with less energy left which may affect the connectivity of WSN. Hence, there is tradeoff in the selection of E_T 's value.

$$f_e(i) = \frac{E^{'} - e(i,0)}{E^{'} - E_T}, \qquad (E_T \le e(i,0) \le E^{'}) \tag{5}$$

ML2B first introduces a new metrics for the selection of rebroadcast node in WSN. It incorporates the three metrics presented above together to select rebroadcast nodes with goals of obtaining low rebroadcast redundancy, high reachability, limited latency, and maximized network life-time. We propose two different ways to combine node degree, coverage rate and left energy metrics into a single synthetic metric, based on the product and sum of the three metrics, respectively. If the product is used, then synthetic metric of delaying the attempt to rebroadcast the broadcasted message is given by formula (6). The sum, on the other hand, leads to a new metric shown by formula (7) by suitably selected values of the three factors: α , β and λ .

$$f^{pro}(d(i,0), l(i,0), e(i,0)) = f_d(i) f_l(i) f_e(i) \tag{6}$$

$$f^{sum}(d(i,0), l(i,0), e(i,0)) = \alpha f_d(i) + \beta f_l(i) + \lambda f_e(i) \tag{7}$$

Nodes with minimized $f(d(i,0), l(i,0), e(i,0))$, rebroadcast the message with the least latency. We compute the add-delay with the following formula:

$$D(i) = D.f(d(i,0), l(i,0), e(i,0)) \tag{8}$$

(where D defines the maximum add-delay, $f(d(i,0), l(i,0), e(i,0))$ is the synthetic metric shown by formula (6) or (7)). Hence, based on formulas: (3)–(8), we can get product and sum versions of add-delay are:

$$D^{pro}(i) = \frac{D[a - d(i,0)][E^{'} - e(i,0)][r - l(i,0)]}{(E^{'} - E_T)ar} \tag{9}$$

$$D^{sum}(i) = D(\frac{\alpha[a - d(i,0)]}{a} + \frac{\beta[r - l(i,0)]}{r} + \frac{\lambda[E^{'} - e(i,0)]}{E^{'} - E_T}) \tag{10}$$

4.2 Algorithm Description

ML2B is a delay based broadcast protocol, where add-delay $D(i)$ is synthetically calculated based on the only one-hop local information at each node, thus making it a truly distributed broadcast algorithm. The final important goal of a broadcast routing algorithm is to carry broadcasted messages to each node in network with as less rebroadcast redundancy as possible, satisfied reachability and maximized life-time of network. ML2B is designed with the idea in mind. Let s be the broadcast originator, the algorithm flow for $\forall$ node $i \in \left(V - \{s\}\right)$ may be formalized as follows:

- Step 0: Initialization: $j = -1$, $D(i) = D$, $UF(i) = \varnothing$.
- Step 1: If node i has received broadcasted message M_s, go to step 2; else if $j \geq 0$, go to step 7, else the node is idle, and stay in step 1.
- Step 2: Check the node ID of originator S and the message ID. If M_s is a new message, go to step 3; else, node i has received the message before, then let $j = j + 1$, and go to step 4.
- Step 3: Let $t = 0$, and the system time begins. Let $j = 0$, where j indicates the times of the repeated i's reception of M_s. Let

$$UC(i,0) = NB(i) \tag{11}$$

  Thus, node degree $d(i,0)$ equals the number of all its neighbors. If $e(i,0)$ is smaller than an energy threshold E_T, node i abandons its attempt to rebroadcast, and go to step 9.
- Step 4: Let $t_j = t$, and use p_{t_j} to mark the previous-hop node of M_s. p_{t_j} transmitted M_s at t_j. We assume the propagation delay can be omitted. Then we get:

$$uf(i,t_j) = p_{t_j} \tag{12}$$

  p_{t_j} is the j th up-link forward node of node i. Add p_{t_j} to up-link forward set $UF(i)$ at last.
- Step 5: Based on the locally obtained position of $uf(i,t_j)$, node i computes the geographical coverage range of $uf(i,t_j)$ which is expressed as $C(i,t_j)$. Then it updates $UC(i,t_j)$ by deleting nodes that locate in $C(i,t_j)$ from $UC(i,t_j)$. Based on the updated $UC(i,t_j)$, node i could find out its degree $d(i,t_j)$. If $d(i,t_j) \leq n$, it abandons its attempt to rebroadcast, and go to step 9; else if $j > 0$ go to step 7.

- Step 6: $j = 0$ means node i has received M_s for the first time. It calculates its add-delay $D(i)$ based on three factors: $d(i,0)$, $l(i,0)$ and $e(i,0)$. $l(i,0)$ equals the Euclidean distance between node i and $uf(i,0)$. $d(i,0)$ has been calculated by step 5, and $e(i,0)$ can be obtained locally. When we get the value of the three parameters, the add-delay can be obtained using formula (9) or (10).
- Step 7: Check the current time t : if $t < D(i)$, go to step 1; else let $d(i,t) = d(i,t_j)$.
- Step 8: If $d(i,t_j) \le n$, node i abandons its attempt to rebroadcast; else rebroadcasts M_s to all its neighbors.
- Step 9: the algorithm ends.

Option for the value of abandoning threshold n affects the rebroadcast redundancy and reachability. There is a tradeoff between the two performance metrics, in which large n leads to low reachability, while little one may not achieve as low broadcast redundancy as large n could achieve. The value of abandoning threshold can be selected depending upon the scenarios and applications of WSN.

5. Performance Evaluation

To verify the proposed ML2B, we made lots of simulations using NS-2 (NS-2, 2006) which is a network simulator supported by DARPA and NSF, with an 802.11 MAC layer. We study the performance of ML2B in the simulated wireless ad hoc networks. Nodes in the wireless multi-hop network are placed randomly in a 2-D square area. For all simulation results, each broadcast stream consists of packets of size 512 bytes and the inter arrival time is uniformly distributed around a mean rate varying from 2 packets-per-second (pps) to 10 pps depending upon the simulation scenarios.

In the all simulations made in this paper, we use the formula (10) to calculate the add-delay for each node by selection that $\alpha = 0,\ \beta = f_d(i)/2, \lambda = f_d(i)/2$. The abandoning threshold and energy threshold used in our simulations are configured as $n = b/5$ and $E_T = E'/100$, where b is the average number of neighbors of nodes.

5.1. Performance Metrics Used in Simulations

We consider four performance metrics:

- ***Saved rebroadcast (SRB):*** $(x - y)/x$, where x is the number of nodes that receive the broadcasted message, and y is the number of nodes that rebroadcasts the message after their reception of the message.
- ***Reachability (RE):*** x/z , where z is the number of all nodes in the simulated connected network. So RE is also known as the coverage rate.
- ***Maximum end-to-end delay (MED)***: the interval form the time the broadcasted message is initiated to the time the last node in the network receiving the message.
- ***Life-time (LT):*** the interval from the time the network is initiated to the time the first node dies.

The saved rebroadcast (SRB) and reachability (RE) metrics were utilized to evaluate the performance of broadcast algorithms by most of the proposed broadcast approaches (S Y Ni et al., 1999) ; (D. Katsaros &Y. Manolopoulos, 2006) ; (F. Ingelrest & D. Simplot-Ryl, 2005) etc.

5.2. Simulation Results

Performance Dependence on the Network Scale

To study the performance of ML2B under different network scales, we design four scenarios by placing randomly different number of nodes separately in squares areas of different size, to maintain a same node density under different network scales. The packets generation rate in this experiment is 2 pps. As illustrated in Fig. 2 and Fig. 3, ML2B achieves high saved rebroadcast without sacrificing the reachability and maximum end-to-end delay under varying network size. According to expectation, maximum end-to-end delay increases with the increased network scale. From Fig. 3 we can see that the network with 10 nodes has a higher SRB than other cases. That is because 10 nodes randomly placed in a 300m×300m square may be within a node's coverage area which is larger than the area of the square (radius of a node's coverage is 250m). The trend of SRB in the left larger scale networks becomes flat, due to the same node density.

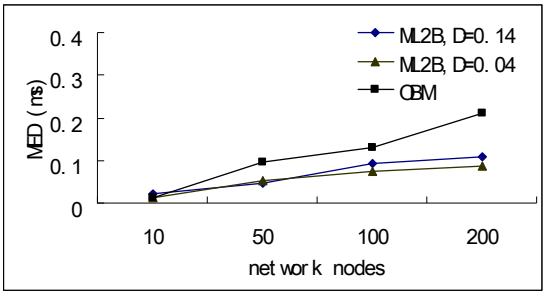


Fig. 2. MED dependence on network scale.

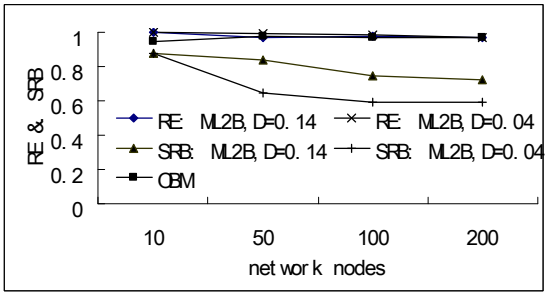


Fig. 3. SRB &. RE dependence on network scale

Performance Dependence on Node Density

We made many experiments to study the ML2B performance dependence on node density. For the reason of limited pages, we give the results of the network consisting of 50 nodes, which is shown by Fig. 4 and Fig. 5. The packets generation rate here is 2 pps. Results illustrated by Fig. 5 shows saved rebroadcast of ML2B fall with the decrease of node density. That is because the theoretical value of the saved rebroadcast depends upon the node density. Large density causes big SRB, and ideal SRB will be zero when the node density is below a certain threshold, which is not the main issue of this paper.

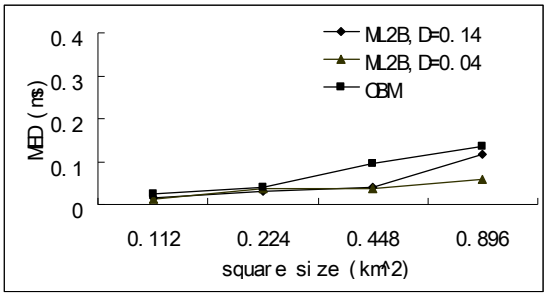


Fig. 4. MED dependence on node density

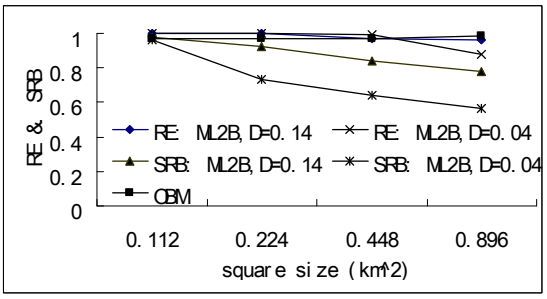


Fig. 5. SRB &. RE dependence on node density

We also compare the performance of ML2B with maximum add-delay $D = 0.14$ s and $D = 0.04$ s. From Fig. 2–Fig. 5 it is clear that the former outbalanced the latter in SRB and RE. And both of them have less MED than the OBM in all circumstances. Therefore, in the following experiments we set $D = 0.14$ s.

Performance Dependence on Packets Generation Rate

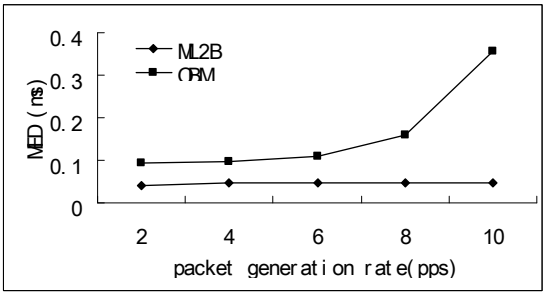


Fig. 6. MED dependence on network load

We study the influence of network load on network performance by varying the packets generation rate from 2 pps to 10 pps. Simulation results in Fig. 6, Fig. 7 show that increased network load incurs little impact on ML2B, however leads to increased MED in OBM. ML2B

maintains nearly as high RE as OBM and, simultaneously achieves SRB with a value larger than 80%, which reveals the superiority of ML2B over OBM.

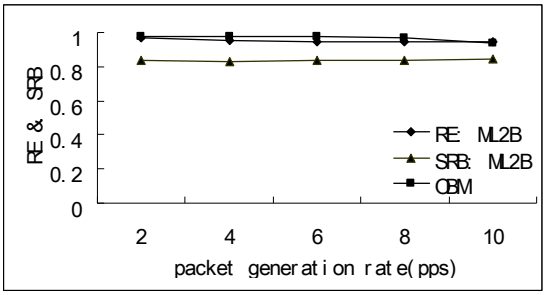


Fig. 7. SRB &. RE dependence on network load

It can be summarized from the above simulations that, ML2B achieves high saved rebroadcast without sacrificing the reachability and maximum end-to-end delay under all circumstances. It is beyond our expectation that ML2B, which has delayed the rebroadcast for an interval of $D(i)$, obtains a smaller maximum broadcast end-to-end delay than OBM that has not delayed rebroadcast. For the different add-delay values for different nodes in ML2B greatly alleviates and avoids the contention and its resulting collision problem that persecutes OBM seriously. In ML2B, nodes rebroadcast the message with less contention for the communication channel, thus making ML2B achieve a smaller maximum end-to-end delay than OBM. In a word, ML2B could effectively relieve the broadcast storm problem.

Life-Time Evaluation

Fig. 8 shows the network life-time of OBM and ML2B under the same scenario, in which each node's initial energy is uniformly distributed between 0.5 J (joule) and 1.0 J. The first and last node dies separately at 32.48 s and 33.62 s in OBM. After 33.62 s no node dies due to malfunction of the broadcast caused by the unconnectivity of WSN due to the too much dead nodes. While in ML2B, they happen at 73.05 s and 95.0 s separately. Life-time is defined as the interval from the time WSN was initiated to the time the first node died. Obviously, ML2B has more than doubles the useful network life-time compared with OBM.

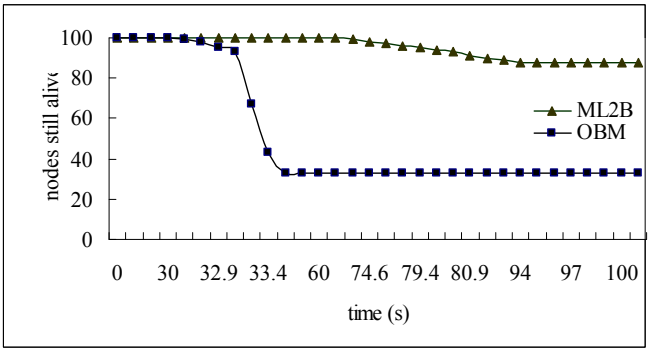


Fig. 8. Number of nodes still alive in the network of 100 nodes

We break the whole simulation time into many small time steps which also are called as rounds. Broadcast originator broadcasts each packet to other nodes in the network during each round. Table.1 shows the network life-time by round with different initial energy, which manifests ML2B obtains much longer network life-time than OBM under different initial energy.

Energy (J/node)	Protocol	Life-Time (rounds)
0.25	ML2B	192
	OBM	45
0.5	ML2B	245
	OBM	91
1.0	ML2B	407
	OBM	195

Table 1. life-time using different amount of initial energy

6. Conclusion

This paper focused on the broadcasting design of wireless multi-hop networks. When a node has packets to broadcast in the network, the broadcast protocol should route these packets to all nodes in the network with little overhead, latency, and consumed energy. To alleviate the broadcast storm problem and simultaneously maximize the network life-time, we propose a new and efficient broadcast protocol-----Maximum Life-time Localized Broadcast (ML2B) for WSN such as wireless ad hoc and sensor networks. ML2B is featured by the following properties: effective reduction of the rebroadcast redundancy, adaptation to node degree, energy conservation, and synthetic consideration of node degree, coverage rate and left energy when selecting rebroadcast nodes. ML2B is based on add-delay strategy which is adopted from the delay-based geographical routing (M. Mauve et al., 2001); (B. Blum et al., 2003) in wireless ad hoc networks. However, the add-delay strategy used in ML2B is different from that used in the geographical routing. The main goal of add-delay here is to select applicable rebroadcast nodes to achieve high broadcast efficiency without sacrificing the network life-time. We also proposed two methods to calculate the add-delay.
To further reduce the rebroadcast redundancy and maximize the network life-time, ML2B has defined two thresholds: abandoning threshold and energy threshold. The former makes nodes with little uncovered neighbors abandon their rebroadcast, and the latter makes nodes with very little energy left in their batteries refuse to rebroadcast messages. The two thresholds could save a number of unused rebroadcasts, decrease the needed total energy for a message broadcast, and extend the network life-time consequently.
Simulations results have verified the effectiveness of ML2B through different ways, which manifest that ML2B achieves high saved rebroadcast with lower maximum end-to-end delay than OBM without sacrificing the reachability under all circumstances. And simultaneously, it has more than doubles the useful network life-time compared with OBM.
However, there are still some works left in ML2B. E.g., the formulas for the add-delay calculation may also needs some improvements. We only simulate the sum version the synthetic metric for the selection of broadcast nodes. The product version synthetic metric

shown by formula (9) will be investigated and simulated in the future work to evaluate its performances.

7. Acknowledgements

The first author would like to acknowledge helpful discussion and solid support from the co-authors of the chapter.
This work was supported by NPU Foundation for Fundamental Research (NPU-FFR-JC201004).

8. References

A. Durresi, V. Paruchuri, "Broadcast protocol for energy-constrained networks," IEEE Transactions on Broadcasting, vol. 53, no. 1, Mar. 2007, pp. 112-119.

B. Blum, T. He, S. Son, J. Stankovic. IGF: A State-free Robust Communication Protocol for Wireless Sensor Networks. Department of Computer Science, University of Virginia, USA, Tech. Rep. CS-2003-11, 2003.

C. R. Mann, R. O. Baldwin, J. P. Kharoufeh, and B. E. Mullins, "A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks," Telecommunication System, vol. 35, no. 1-2, Jun. 2007, pp. 67–86.

D. Li, X. Jia, and H. Liu, "Minimum energy-cost broadcast routing in static ad hoc wireless networks," IEEE Transactions on Mobile Computing, vol. 3, no. 2, Apr.-Jun. 2004.

F. Ingelrest and D. Simplot-Ryl, "Localized broadcast incremental power protocol for wireless ad hoc networks," Proc. IEEE ISCC, 2005.

F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient broadcast: a robust data delivery protocol for large scale sensor networks," Wireless Networks, vol. 11, no. 3, May 2005, pp. 285–298.

J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," Proc. IEEE INFOCOM, 2000.

J.-P. Sheu, C.-S. Hsu and Y.-J. Chang, "Efficient broadcasting protocols for regular wireless sensor networks," Wireless Communications and Mobile Computing, vol. 6, no. 1, 2006, pp. 35–48.

J.-P. Sheu, S.-C. Tu, and C.-H. Yu, "A distributed query protocol in wireless sensor networks," Wireless Personal Communications, vol. 41, no.4, Jun. 2007, pp. 449–464.

J. Wu and F. Dai, "A generic distributed broadcast scheme in ad hoc wireless networks," IEEE Transactions on Computers, vol. 53, no. 10, Oct. 2004, pp. 1343-1354.

M. Agarwal, J. H. Cho, L. Gao, and J. Wu, "Energy efficient broadcast in wireless ad hoc networks with hitch-hiking," Proc. IEEE INFOCOM, 2004.

M. Cagalj, J.P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," Proc. MOBICOM, 2002.

Miklós Maróti, "Directed flood-routing framework for wireless sensor networks," Proc. IFIP International Federation for Information, LNCS 3231, 2004, pp. 99–114.

M. Lin, K. Marzullo and S. Masini, "Gossip versus deterministic flooding: Low packet overhead and high reliability for broadcasting on small networks." UCSD Tech. Rep. 0637, 1999.

M. Mauve, J. Widmer, H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. IEEE Network. pp. 30-39, Nov. /Dec. 2001.

M.-T. Sun and T.-H. Lai, "Location aided broadcast in wireless ad hoc network systems," Proc. IEEE WCNC, 2002, pp. 597- 602.

N. B. Chang, and M. Liu, "Controlled flooding search in a large network," IEEE/ACM Transactions on Networking, vol. 15, no. 2, Apr. 2007, pp. 436-449.

NS-2 Network Simulator , http://isi.edu/nsnam/ns/index.html. Jun. 2009

P.J. Wan, G. Calinescu, X.Y. Li, and O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," Proc. IEEE INFOCOM, 2001.

R.Q. Zhao et al., Maximum Life-time Localized Broadcast Routing in MANET. Lecture Notes in Computer Science, 2007, 4672(1): 193–202.

S Y Ni, Y C Tseng, Y S Chen, J P Sheu. The Broadcast Storm problem in a Mobile Ad Hoc Network. Proceedings of the Fifth Annual ACM/ IEEE International Conference on Mobile Computing and network .Washington: IEEE, pp. 151–162, 1999

W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," Proc. MOBIHOC, 2000, pp. 129-130.

W.Z. Song, X. Y. Li, and W. Z. Wang, "Localized topology control for unicast and broadcast in wireless ad hoc networks," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 4, 2006, pp. 321-334.

X. Hui, M. Jeon, S. Lei, N. Yu, J. Cho, and S. Lee, "Impact of practical models on power aware broadcast protocols for wireless ad hoc and sensor networks," Proc. IEEE Workshop on SEUS–CCIA, 2006.

Y.-W. Hong and A. Scaglione, "Energy-Efficient Broadcasting with Cooperative Transmissions in Wireless Sensor Networks," IEEE Transactions on Wireless Communications, vol. 5, no. 10, Oct. 2006, pp. 2844-2855.

6

Routing Protocol with Unavailable Nodes in Wireless Sensor Networks

Deyun Gao, Linjuan Zhang and Yingying Gong
National Engineering Laboratory for Next Generation Internet Interconnection Devices, School of Electronic and Information Engineering, Beijing Jiaotong University Beijing 100044, P.R.China

1. Introduction

With the rapid development of modern microelectronic technology, wireless communication technology, signal processing technology, and computer network technology, wireless sensor networks (WSNs) has become one of the most important and the most basic technologies of information access (Jennifer Yick, 2008). WSNs have been widely used in military, environment monitoring, medicine care and transportation control. Routing protocol is one of the key support technologies in WSNs and the performance of routing protocols significantly impact the performance of the entire network (Khan & Javed, 2008).

In wireless sensor networks (WSNs), some unavailable areas often are formed because some sensor nodes become unavailable due to energy exhausted, congestion, or disaster (Fang et al., 2006; Jafarian & Jaseemuddin, 2008). Multi-path routing protocol is one of the mechanisms to solve or alleviate the above problems. Data delivery over multiple paths can help balance network load and extend the life time of entire network. Generally, multiple paths in the routing protocols can be classified into two categories: disjoint multiple paths and joint multiple paths (Ganesan et al., 2001). Disjoint multiple paths can be furthermore classified into node-disjoint multiple paths and link-disjoint multiple paths. In the node-disjoint multiple paths, each path is independent and has no affect on each other. Apparently, it is better to choose node-disjoint multiple paths for data delivery in the designed routing protocol if possible.

Most of multi-path routing protocols in wireless ad hoc networks are extended from classical single path routing protocols. For example, split multi-path routing (SMR) is based on the dynamic source routing (DSR) protocol and ad hoc on demand multi-path distance vector routing (AOMDV) extends the ad hoc on-demand distance vector routing (AODV) protocol (Lee & Gerla, 2001; Marina & Das, 2001). Similarly, as its special type, most of multi-path routing protocols in WSNs are extended from the ones in wireless ad hoc networks and at the same time take account of different factors such as energy, QoS, security, congestion, and etc. There are many papers to consider energy efficiency when designing multi-path routing protocols in WSNs (KIM et al., 2008). They mainly select multiple paths based on the link cost function consisting of both the node residual energy level and hop count. In (Huang & Fang, 2008), Xiaoxia Huang and Yuguang Fang proposed a probabilistic modeling of link state for wireless sensor networks. Based on this model, an approximation of local multi-path routing algorithm is explored to provide soft-QoS under multiple constraints, such as delay and reliability. Yunfeng Chen and Nidal Nasser proposed to select multiple paths between one

sink and multiple sources with the consideration of reducing collision occurred at nodes that are receiving and forwarding packets on behalf of the source nodes in order to improve QoS (Chen & Nasser, 2008). The same authors proposed an secure and energy-efficient multi-path routing protocol (SEER) (Nasser & Chen, 2007). Besides of using multiple paths alternately for communication between two nodes to prolong the lifetime of the network, SEER is resistive some specific attacks that have the character of pulling all traffic through the malicious nodes by advertising an attractive route to the destination. In (Toledo & Wang, 2006), Alberto Lopez Toledo and Xiaodong Wang proposed to use network coding to achieve an adaptive equivalent solution to the construction of disjoint multi-path routes from a source to a destination. It exploits both the low cost mesh-topology construction, such as those obtained by diffusion algorithms, and the capacity achieving capability of linear network coding. Jenn-Yue Teo, Yajun Ha, and Chen-Khong Tham proposed a heuristics-based interference-minimized multi-path routing (I2MR) protocol that increases throughput by discovering and using maximally zone-disjoint shortest paths for load balancing and a congestion control scheme that is able to adjust the loading rate of the source dynamically (Teo et al., 2008).

However, the existed multi-path routing protocols can not provide mechanisms to cross around the unavailable areas particularly during the routing building procedure or later data delivering procedure. Because in WSNs the states of sensor nodes or areas are changing due to many factors, it is important to consider all of these factors and situations when designing the routing protocols. In this chapter, we propose a new micro sensor multi-path routing protocol (MSMRP) to avoid crossing the unavailable areas based on the micro sensor routing protocol (MSRP) previously developed by us (Gao et al., 2009). We firstly define the unavailable areas that may be formed due to kinds of reasons such as energy exhausted, disaster and so on, which can be detected by kinds of sensors through some predefined settings. Then we design several new routing packets and routing tables to help building multiple paths based on the MSRP. In particularly, we propose a neighbor node table exchanging mechanism that can help build an alternate route around the unavailable areas and try to avoid the multiple paths intersect. When a sensor node becomes unavailable during the route reply (RREP) forwarding procedure, its precursor node will try to find the alternate route to forward the RREP to the destination with the help of above mechanism. It also can help balance the network load, improve the transmission efficiency and routing stability with multi-path transmission, which furthermore decreases the unavailable areas' forming and enlarging. Finally, we implement the proposed protocol in the real sensor nodes and set up a testbed to conduct detail experiments. The experimental results show that MSMRP can perform well to build up multiple paths to avoid the unavailable areas.

This chapter is organized as follows. Section 2 describes the MSRP routing protocol. Section 3 introduces the definitions of unavailable and available areas, and presents the details of the MSMRP including new added message formats and operation mechanisms. Section 4 introduces our developed sensor node's hardware architecture. Section 5 presents the software architecture, operation mechanisms of some standard interfaces of the connector module and adaptive data processing scheme. Section 6 shows the experimental performance results of WSNs implementing MSMRP. Some important conclusions are drawn in Section 7.

2. Micro Sensor Routing Protocol

Based on AODV, we designed Micro Sensor Routing Protocol (MSRP)for IEEE802.15.4 based sensor network. In the following, we firstly describe the protocol stacks of IPv6 sensor node designed. Then, we present the details of MSRP.

2.1 Protocol Stack of IPv6 sensor node

Fig. 1 shows protocol stack of IPv6 sensor node designed by us. We divide the protocols into fiver layers including application layer, network layer, adaptation layer, data link layer and physical layer. Considering scare resources we simplify the traditional transportation layer (TCP and UDP) and merge them into network layer. Also, we put our MSRP routing protocol into network layer. Specially, we add a new adaption layer. For other layers, it is easy to understand their functions and we do not need to introduce them. Here, we just describe the adaptation layer.

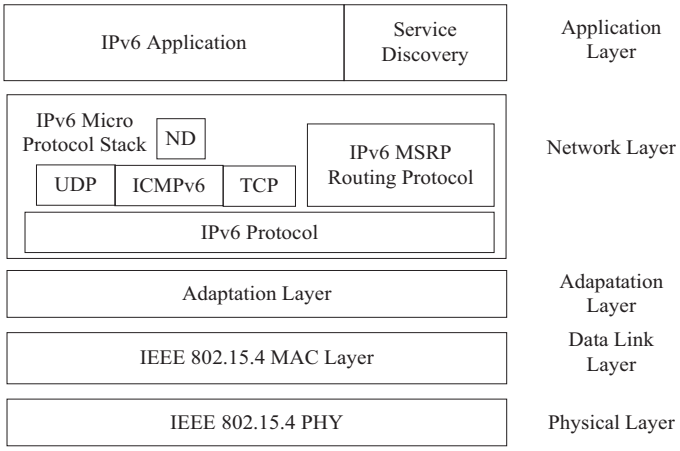


Fig. 1. Architecture of IPv6 Wireless Sensor Node

The adaptation layer lies between IEEE 802.15.4 MAC layer and the network layer. Adaptation layer is used mainly for fragmentation and reassembly. As we use IPv6 in the network layer, the maximum transmission unit (MTU) size for IPv6 packets over IEEE802.15.4 is 1280 octets. However, a full IPv6 packet does not fit in an IEEE802.15.4 frame. IEEE802.15.4 protocol data units have different sizes depending on how much overhead is present. Starting from a maximum physical layer packet size of 127 octets and a maximum frame overhead of 25, the resultant maximum frame size at the media access control layer is 102 octets. Link-layer security imposes further overhead, which in the maximum case (21 octets of overhead in the AES-CCM-128 case, versus 9 and 13 for AES-CCM-32 and AES-CCM-64, respectively) leaves only 81 octets available. This is obviously far below the maximum IPv6 packet size of 1280 octets, and in keeping with Section 5 of the IPv6 specification (Deering & Hinden, 1998), a fragmentation and reassembly adaptation layer must be provided at the layer below IP. Furthermore, since the IPv6 header is 40 octets long, this leaves only 41 octets for upper-layer protocols, like UDP. The latter uses 8 octets in the header which leaves only 33 octets for application data. Thus, there is a need for a fragmentation and reassembly layer.

2.2 Micro Sensor Routing Protocol Packet Format

In order to reduce low-speed IPv6 WSN equipment energy consumption, it is very important to design efficient and streamlined routing protocol packet formats. Considering low-speed wireless network characteristics, we designed our routing protocol with three routing packet formats including routing request (RREQ), routing reply (RREP) and routing error (RERR). We

do not use a Hello mechanism for route maintenance, thereby reducing the routing packet size sent in establishing new routes and maintaining them, which will reduce energy consumption. In the following we take a more descriptive look at these three packet formats.
Fig. 2 and 3 show the Route request packet format and the Route reply packet format respectively.

Type (3 bits)	Reserved (5 bits)	Hops (1 byte)
Source Address (8 bytes)		
Destination Address (8 bytes)		
RREQ_ID (2 bytes)		
MLQI (1 byte)		

Fig. 2. RREQ Packet Format

Type (3 bits)	Reserved (5 bits)	Hops (1 byte)
Source Address (8 bytes)		
Destination Address (8 bytes)		
MLQI (1 byte)		

Fig. 3. RREP Packet Format

The fields in these two packets are the followings.

- Type: 000, 001 for RREQ and RREP message types respectively;
- Reserved: Reserved for future enhancements;
- Hops: Number of nodes RREQ or RREP messages passed from the corresponding source to current Node;
- RREQ_ ID: Unique identifier of RREQ message;
- Source Address: Address of the node which initiated RREQ or RREP;
- Destination Address: Requested route destination node address or Address of the node which initiated RREQ;
- MLQI: Minimum of the Link Quality Indicator (LQI) values between RREQ or RREP source to current node.

Fig. 4 illustrates the route error message format.

- Type: 010, for the Route Error (RERR) message format type;
- No. of Addresses: Number of neighbors which became unreachable as detected by the RERR originator node;
- Unreachable Destination Address n: Addresses of nodes unreachable (Number of addresses depend on “No. of Unreachable Addresses” field, in order to comply with IEEE802.15.4 standard, a IEEE802 .15.4 the size of data packets is not more than 128 bytes, hence one Route Error (RERR) message may Carry up to 4 unreachable addresses);

Type (3 bits)	No. of addresses (2 bytes)	Hops (1 byte)
Unreachable Destination Address 1 (8 bytes)		
Unreachable Destination Address 2 (8 bytes)		
...	...	

Fig. 4. RRER Packet Format

2.3 Routing Tables

Fig. 5 illustrates a routing table entry.

Type (1 bit)	Reserved (7bits)	PAN ID (2 bytes)	
Hop Limit (1 byte)		Time to Expire (1 byte)	Route LQI Value (1 byte)
Destination Address Interface ID (8 bytes)		Next-Hop Address Interface ID (8 bytes)	
Precursor Node Address Interface ID 1 (8 bytes)		Precursor Node Address Interface ID 2 (8 bytes)	
...		.	

Fig. 5. Routing Table Entry

- Type: Used for distinction between two types of equipments: Cluster head (0) and the cluster members (1);
- PAN ID: PAN (Personal Area Network) identifier;
- Hop Limit: No. of hops for this route;
- Time to Expire: The time of the expiration or deletion of this route entry;
- Route LQI value: Minimum LQI value of the Route;
- Destination Address Interface ID: Interface identifier(IEEE 64bit) of the destination node;
- Next-Hop Address Interface ID: Interface identifier(IEEE 64bit) of the next-hop of the route;
- Precursor Node Address Interface ID: Interface identifier(IEEE 64bit) of the previous node in the route (possibly more than one, used to send RERR messages);

In IPv6 WSN, routing protocol must avoid routing loops, reduce invalid data packets, effectively record routes and dynamically adapt to the changes in network topology and improve the information transmission efficiency. In our Micro Sensor Routing Protocol (MSRP) when a source needs to send data packet to unknown destination it will encapsulation and broadcast a RREQ packet. But the intermediate nodes will receive multiple instances of this RREQ packet through multiple paths. If the intermediate node broadcasts each time when this type of RREQ is received, this will create broadcast storms, which will affect the network performance and by increasing energy consumption of nodes it will decrease the network life time. Therefore we use a mechanism which involves a duplicate routing table. Dupe table will be inserted with the RREQ message information with the unique RREQ_ID. If another RREQ

message arrived from the same source (through a different path) with the same RREQ_ID before the entry expiration time, this new packet will be dropped. This mechanism effectively reduces overhead on the network at route establishment phase. Fig. 6 shows a dupe table entry.

RREQ Source Address (8 bytes)	RREQ_ID (2 bytes)	Time to Expire (1 byte)

Fig. 6. Dupe table entry

- RREQ Source Address: Address of the node which initiated one RREQ message;
- RREQ_ID: Unique identifier of RREQ message;
- Time to Expire: The time of the expiration or deletion of a route;

2.4 Route Selection and Decision Making Process of MSRP

MSRP is actually an on-demand routing protocol. When there is a need to send data to a destination, source node launches routing search process to find the corresponding route. This kind of on-demand routing protocol overhead is reduced and suitable to IPv6 WSN with energy saving requirements.

2.4.1 Sending RREQ

In IPv6 WSN, when a node needs to send data to another destination node, first search the local routing table, if no entry to the destination exists, cache current data packets and create RREQ packet, then broadcast the RREQ.

2.4.2 When intermediate nodes receive a RREQ

First when a intermediate node receives a RREQ message, it checks if the destination address is itself, if not, first check its dupe table. If there already exists a similar entry, that means this node received a RREQ from the same source with the same RREQ_ID, in order to reduce LR-WPAN energy consumption and broadcast storms, this duplicate RREQ is dropped. If no entry exists in the duplicate table, route to the source is added to routing table, then if there exists a route to the source, compare the two routes and store the optimum route. If there is a route to RREQ destination then unicast the RREQ to its destination, otherwise broadcast the packet.

2.4.3 When the destination node receive a RREQ

If the node detects that the destination address of RREQ equals its own, then it enters route reply process:
First of all node will put RREQ message in a cache table. Because in the RREQ path determining process, RREQ messages are broadcasted through network and hence the node might receive multiple RREQ messages through multiple paths, as a result it is necessary to wait for a reasonable period of time T, afterwards we apply the route determining function $f(m,h,n)$.

$$f(m,h,n) = Am + Bh + Cn \tag{1}$$

where m is the number of nodes with insufficient energy from source to destination, h is the number of hops from source to destination, n is the number of links with weak LQI between

source to destination. A, B, C parameters are to be determined under different network environments. A, B and C are values produced by non negative integer powers of 2, and must meet the condition $A >> C > B$. For example, in the open area network environment, we can use $A = 256$, $B = 1$, $C = 2$.
Destination node will calculate f value for each RREQ received through different routes from source. Then it will compare f value for current route to source if there is an entry in the routing table, then choose the entry with the lowest f value and start the reply process. Afterwards it unicasts a RREP through optimum path to RREQ source.
Fig. 7 describes the receive route request process.

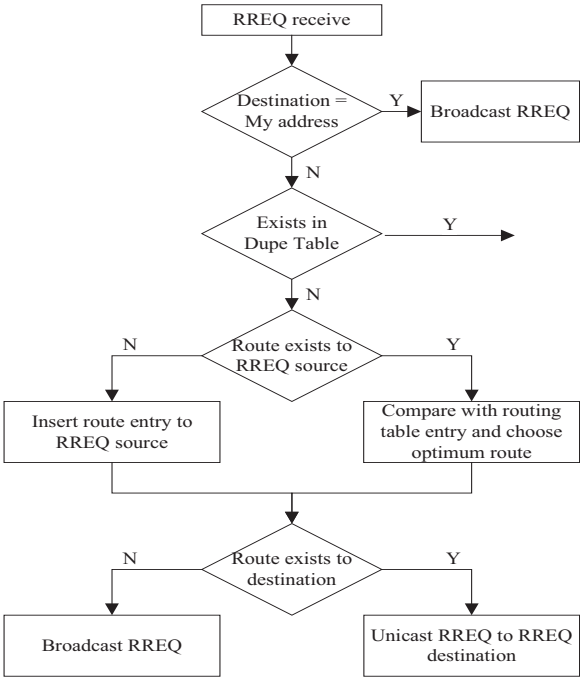


Fig. 7. Receive routing request

2.4.4 Receive RREP

When nodes receive RREP message, the first objective is to determine RREP destination address is itself. If it has established a routing entry to RREP source in the routing table, send the data items in the transmission buffer. Otherwise, it inserts or updates the RREP source address route entry, searchs routing table for the route to RREP destination, and then unicasts RREP towards its destination.

2.5 Route Maintenance and Error Handling Process

Micro Sensor Routing Protocol does not use the traditional maintenance methods like AODV Hello messages. Furthermore IEEE802.15.4 standard uses ACK frames to determine neighbor node reliability, if you do not receive an ACK in certain period of time after sending data,

this means that the neighbors nodes had expired, then save current data to a buffer, once again start the RREQ process, at the same time send a RERR to the Precursor node. It has the advantage of reducing energy consumption and network resource usage of sending and receiving Hello messages, on the other hand low-rate WPAN equipment usually are not very delay sensitive. As an on-demand routing protocol MSRP more effectively performs route maintenance.
When a neighbor node failure is detected, first find routing entries with that node address as the next-hop address. Then get their precursor node address and encapsulate a RERR message, unicast the precursor nodes with RERR, then delete the Routing table entries with that node as the next-hop address from the routing table. When a precursor node receives a RERR message, similarly process unreachable entries in the routing table, until all precursor nodes in this route has been informed about the route expiration.

3. Micro Sensor Multi-Path Routing Protocol

In this section, we firstly define unavailable areas that are formed due to the occurrence of unavailable sensor nodes, which can not provide data forwarding any more. Then, we introduce the MSMRP operation procedures and key modules.

3.1 Available and Unavailable Areas in Wireless Sensor Networks

In wireless sensor networks, the data delivery over some areas are unfeasible maybe because in this area the energy of sensor nodes are exhausted, or there are serious network congestion, or blind spots in the coverage area, or there are sudden disasters where even some nodes are destroyed. We can define such an area as **unavailable area**. Otherwise, the area where the data delivery can be completed feasibly can be defined as **available area**. Fig. 8 shows an example. In the figure, the red area is marked as unavailable area and the remain area is available area.

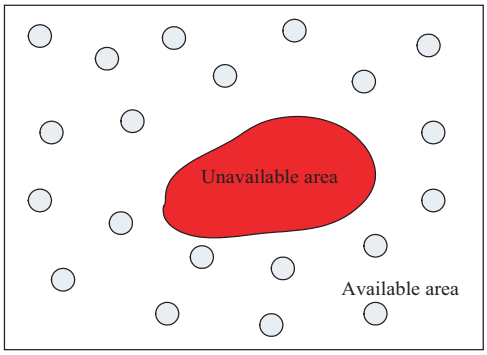


Fig. 8. Available and unavailable areas in wireless sensor networks

These unavailable areas forms because the sensor nodes in these areas becomes unavailable for data delivery. Furthermore, we can divide these unavailable sensor nodes into two categories. **The first category of sensor nodes** are located in the unavailable area. **The second category of sensor nodes** are located in the boarder of the unavailable area and they only have one neighbor node. After it receives the data frame from its neighbor node, it can not

forward out this frame. Each sensor node has a flag bit that is set to "disable" when it becomes a unavailable node.

3.1.1 The First Category of Sensor Nodes

For the classification of the first category of sensor nodes, we can make judgement according to the sensor node's energy, sensory data, network congestion status, and so on under different situations.

For example, the battery voltage is 1.15∼3.7V in our developed sensor node. If the battery voltage of a sensor node is lower than 1.15V, it can not work. Thus, when a sensor node's battery voltage is lower than 1.5v, we should decrease their work load and set it to a unavailable node. In the sensor node, we use a 8-bit digit to represent the energy value and 1.15V∼3.7V can be converted to 0∼255 as shown in Eq. (3).

$$B = (A - 1.15)\frac{255}{3.7 - 1.15} \tag{2}$$

where, A is the battery voltage of a sensor node and B is the converted energy value. When $A = 1.5$V, B is 35, i.e., when the energy value of a sensor node is lower than 35, we set it to a unavailable node.

For the temperature, we assume that the temperature is T, the digital value representing the temperature is SO_T.

$$T = d_1 + d_2 * SO_T \tag{3}$$

where, d_1, d_2 and SO_T of the temperature sensor SHT11 are shown in Tabs. 1 and 2

VDD	$d1[^\circ C]$
5v	-40.00
3.5v	-39.66
3.3v	-39.636
2.5v	-39.55

Table 1. Relationship between VDD (voltage drain drain) and d_1

SO_T	$d2[^\circ C]$
14bit	0.01
12bit	0.04

Table 2. Relationship between SO_T and d_2

For example, we assume that there occurs a disaster such as fire if the temperature is larger than 60°C. If the current VDD is 3.5V, the 14-bit digital SO_T can be calculated as 9966, i.e., if the temperature value in the sensor node is larger than 9966, we set the sensor node as a unavailable node.

We use the number of route entries, queue length, and frame sending rate to judge whether the network load is heavy with three threshold values for them. If any one is larger than the corresponding threshold value, then we think the sensor node is congested and set it to a unavailable sensor node.

3.1.2 The Second Category of Sensor Nodes

After a unavailable sensor node judged with the above mechanism is set, it will immediately send a removing message to its neighbor nodes. The sensor nodes receiving the message will remove the corresponding entries in the neighbor node table and check the number of its neighbor nodes. If it has only one neighbor node in the table, then it will be set to an unavailable node. This kind of unavailable nodes is the second category of sensor nodes.

3.2 MSMRP Routing Protocol Packets and Routing Tables

In order to minimize the energy consumption of the sensor nodes, it is very important to design space efficient and concise routing protocol packet formats and routing tables.

3.2.1 Routing Protocol Packets

In MSMRP, the used routing packets include routing request (RREQ), routing reply (RREP), routing error (RERR), HELLO message, advertisement message of neighbor node table and delete message of neighbor node. Among them, the former three kinds of routing packets can be referred in the MSRP routing protocol (Gao et al., 2009). The latter three kind of messages are explained in the followings.

The first new added type of packet is HELLO message, which includes three fields and is shown in Tab. 3. The field "Type" distinguishes message type. Here it specifies "011" for the HELLO message. The second field "Reserved" is reserved for future enhancements. The third field "Address" is the address of the sensor node that sends out this HELLO message.

Type (3 bits)	Reserved (5 bits)	Address (2 bytes)

Table 3. HELLO message format

The second new added type of packet is the advertisement message of neighbor node table (NDAD), which includes three fields at least and is shown in Tab. 4. The field "Type" specifies "100" for the NDAD message. The second field "Reserved" is reserved for future enhancements. The third field "Address of neighbor node 1" is the address of the first neighbor node. If it includes one more neighbor nodes, their address can be included into the following fields.

Type (3 bits)	Reserved (5 bits)
Address neighbor node 1 (8 bytes)	
Address neighbor node 2 (8 bytes)	
...	

Table 4. Advertisement message of neighbor node table

The third new added type of packet is the delete message of neighbor node (NDDE), which includes three fields and is shown in Tab. 4. The field "Type" specifies "101" for the NDDE message. The second field "Reserved" is reserved for future enhancements. The third field "Address" is the address of the sensor node that sends out the NDDE message.

Type (3 bits)	Reserved (5 bits)	Address (8 bytes)

Table 5. Delete message of neighbor node

3.2.2 Routing Tables

Each sensor node maintains a local routing table (LRT) for packet forwarding and a duplicate (DUPE) routing table to detect duplicate RREQ messages to avoid excessive flooding or control messages during the route discovery process. DUPE table will be inserted with the RREQ message information of a unique RREQ_ID. If another RREQ message arrived from the same source through a different path with the same RREQ_ID before the entry expiration time, this packet will be dropped. LRT and DUPE can also be referred in the MSRP routing protocol (Gao et al., 2009). Another new added table is neighbor nodes table that includes the address of neighbor nodes and is shown in Tab. 6.

Address of neighbor node 1 (8 bytes)	Address of neighbor node 2 (8 bytes)	...

Table 6. Neighbor nodes table

3.3 Operation Procedure of MSMRP

The basic operation procedure of MSMRP includes the followings:

- When the node starts up, it will broadcast a HELLO message to the neighbor nodes. If one neighbor node receives the HELLO message, it will build up a neighbor nodes' table.
- When the source wants to send the collected information to the sink node, it will broadcast a RREQ message. The middle nodes, which received the RREQ message, will broadcast this message until it arrives at the sink node finally.
- After the sink node receives multiple RREQ messages from different nodes, it will select N paths with the minimal hops and save them to a route request table, then send back two route reply messages.
- When a middle node receives the first route reply, it will set a mark in the route entry, which means it is already a forwarding node along the first route.
- If the node receives the second RREP with the same destination node later, it will start the neighbor node table exchanging mechanism to find out a common neighbor node with the previous hop node. And it will set the neighbor node as the next hop, to continue forwarding the second RREP message.
- If the source node receives two RREP messages, it will randomly select a route to send the data or use two routes to balance the network load.

3.3.1 Route Discovery Procedure

As the MSRP, during the route query phase, the source node will broadcast a RREQ message as shown in Fig. 9. When the RREQ is forwarded to a unavailable node, it will be discarded directly. Thus, the inverse route to the source node is not built and later the RREP message sent from sink node will not traverse the unavailable node.

3.3.2 Route Reply Procedure

After some RREQ messages traversing different paths arrive at the sink node finally, the sink node will select two best optimal path to send out the RREP messages, which will arrive at the source node along the inverse route built in the route query procedure.

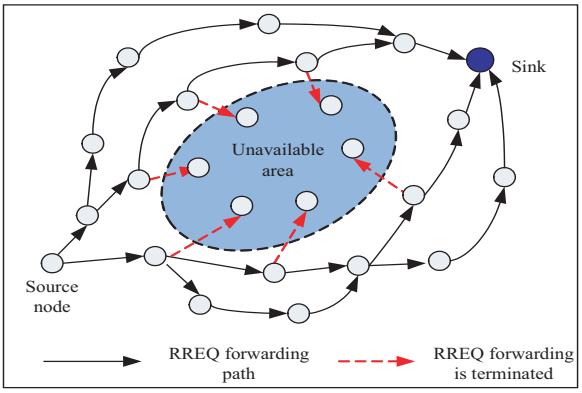


Fig. 9. RREQ message forwarding procedure

The problem that may arise is that there appears a unavailable node along the inverse path. The neighbor nodes of the unavailable node including the previous hop along the inverse path will remove its corresponding route entries, i.e., the inverse path is broken here. The sensor node on previous hop will cache the RREP and try other neighbor node one by one. If its neighbor node has a path to the source node, then it will send ACK to it and the RREP can be forwarded to the source node along a new inverse path. If there is no any neighbor node that has a path to the source node, then it will send NO message to the sink node. And the sink node will select another best path from the remaining paths to send out a RREP message again. The route reply procedure is shown in Fig. 10.

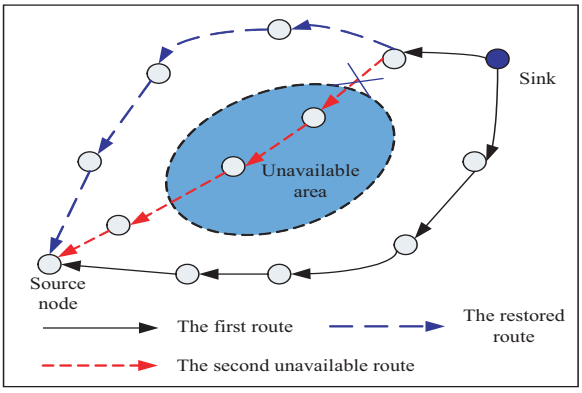


Fig. 10. Route Reply Procedure

3.3.3 Route Maintenance and Error Handling Process

After the route is built up, the source node can send the data along the route. If later some sensor nodes along the route become unavailable nodes, then the data forwarding can not be completed. Therefore, we need to design route maintenance and error handling mechanism to deal with it.

In the RERR process, when a node is notified by the neighbor unavailable node or knows some neighbor node becomes unavailable if its HELLO message dose not been received during a period, it first searches routing entries with that node address as the next-hop address in its LRT. Then get their precursor node address and encapsulate a RERR message, unicast the precursor nodes with RERR, then delete the routing table entries with that node as the next-hop address from the routing table. When a precursor node receives a RERR message, similarly process unreachable entries in the routing table, until all precursor nodes in this route has been informed about the route expiration. In the case where a particular precursor node that also becomes unavailable is detected, our design triggers no further RERR for energy conservation.

3.4 Neighbor Node Table Exchanging Mechanism

In the MSMRP, the multiple paths that can avoid crossing the unavailable area are built with the help of neighbor node table exchanging mechanism.

3.4.1 Neighbor Node Table Building

A node sets up its neighbor node table through the HELLO message broadcasting periodically. When a node starts up, it broadcasts a HELLO message to its neighbors and the node that receives the HELLO message will search its neighbor node table. If the node address information does not exist in its neighbor node table, then it adds the address into its table. If the address exists, then just ignores the HELLO message. This table can be used to help ensure multiple paths disjoint.

3.4.2 Neighbor Node Table Exchanging

After an intermediate node receives a RREP message, it will firstly check up its flag bit to see whether it is an effective node. If it is an unavailable node, it will discard the RREP message directly. Otherwise, it will check whether it is the first received RREP message. If so, it will build up an inverse route to the sink node. Otherwise, it becomes a joint node between two routes. At this time, it will start up the neighbor node table exchanging mechanism to deal with this situation.

When an intermediate node finds out that it is the joint node between two paths after it receives the second RREP message, it will send out its neighbor node table to the precursor node that forwarded the second RREP message to it. The precursor node will find out the common neighbor nodes of them through comparing the received neighbor node table and itself. Following that, the precursor node will send the RREP to a common neighbor node selecting from them. If the selected common neighbor node has a route to the destination node of the RREP message, it will send back an ACK to the precursor node and continue forwarding the RREP message. If the precursor node does not receive the ACK, then it will select another common neighbor node to try again with the repeated procedure. Finally, the second RREP message will arrive at the destination node along the second changed route. However, there maybe not has any common neighbor node that has a route to the destination node. Under this situation, the precursor node will notify the joint node and the joint node will forward the RREP message by itself. Thus, this kind of mechanism can try best to avoid the intersection between two paths.

4. Hardware of the Sensor Node

The designed sensor node should has strong extensibility and is adaptive to the new applications with slight secondary development work. It can improve node flexibility and computation performance.

4.1 Basic Architecture of the Hardware

Fig. 11 shows the hardware architecture of the designed sensor node. The sensor node is divided into main board and expansion board. Processor, wireless communication, power and connector modules, which are the common components for most of applications, are located in the main board. Sensors, memories and other modules, which are changed usually for different applications, are put into the expansion board. Thus, when we reconstruct the sensor node according to users' requirements, this kind of design can reduce the secondary development time and cost.

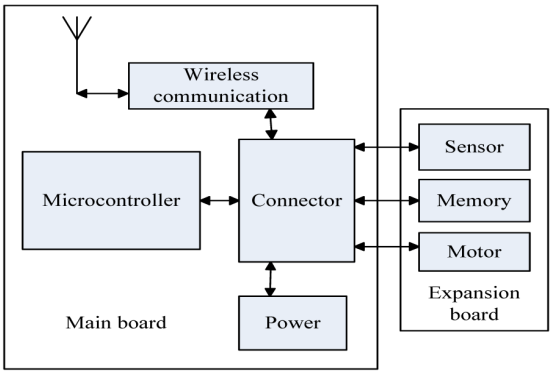


Fig. 11. Hardware architecture of the sensor node

In the following, we briefly introduce each module.

- In the general purpose multi-sensor node, the connector module is responsible to connect other modules and expansion board that is composed of some sensors, memory, and others. If we want to extend any module, then we just need to redesign this module and connect to the connector module. Thus, it is very convenient to extend the functions of sensor node for new applications.
- Microcontroller module is the key of the sensor node, which is in charge of controlling the node, sensor data processing, and etc. It can use the connector module to delivery commands to any module in the node.
- Wireless communication module sends or receives information over wireless link. It is connected to microcontroller through the connector module with SPI interface.
- The sensor module senses and collects the environment information with different type of sensors, and sends these collected information to the microcontroller through the connected module with some standard interfaces.
- The power module provides energy to all of the modules of sensor node, and computes the residual energy for furthermore schedule of the network operation.

4.2 The Connector Module

In our new designed sensor node, we particularly add a new module, connector module. The connector module mainly affects the extensibility of sensor node. The special requirements particularly for the sensors can be put in the expansion board. The sensor module can be re-designed to meet different demands and connect to the main board via the connector module. This kind of design significantly improves the extensibility of sensor networks.

In the connector module, we use a DF9-31 plug-in connection from HRS company. It has 9.9v, 3.3v interferences, SPI, ADC, serial port, and etc. With these connection ports, the sensor node can support many kinds of sensors, radio chips, memorizers, and other devices. The main connection circuit is shown in Fig. 12.

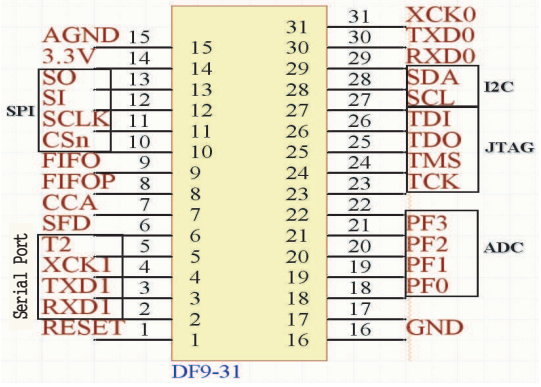


Fig. 12. Interface module schematic of the connector module

It mainly includes the following connection ports.

- SPI interface: It connects to the radio chip.
- Serial Port: It can connect to other terminal for the debugging and communication.
- ADC interface: It connects to some analog sensors for the A/D conversions.
- JTAG interface: It can be used for the software debugging and controlling.
- I2C interface: It connects to some digital sensors and memorizers.
- I/O interface: It also connects to some special digital sensors and memorizers through simulating time sequences.
- 3.3v and 9.9v power interface.

As shown in the figure, "SO, SI, SCLK, CSn" are used for the SPI interface, "TDI, TDO, TMS, TCK" are used for the JTAG interface, "ADC3, ADC2, ADC1, ADC0" are used for the ADC interface, "RXD1, TXD1, XCK1, T2" are used for the serial port, "SCL, ADA" are used for the I2C bus interface, respectively. "GND" and "AGND" are the ground pins. "9v" and "3.3v" provide the power for other components in the node. "RESET" is used for the reset of the microcontroller ATmegal128. "FIFO, FIFOP, CCA, SFD" can be used for the interruption, which connects to the wireless communication module. "XCK0, TXD0, RXD0" are used for the serial port interface. By the way, all of the above ports can be used for I/O interface. Due to the space limitation, Fig. 13 only shows the serial port schematic as an example.

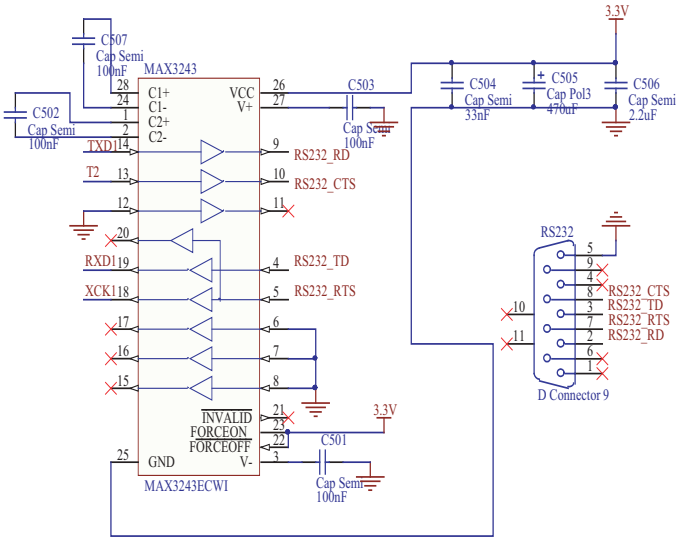


Fig. 13. Serial port schematic

With the above connector module, it improves the extensibility of sensor node. The sensor node can be configured with some common sensors such as temperature/humidity sensor, luminance sensor and accelerometer, support other analog and digital sensors with I2C bus interface and other ports, and connect to the memories that can keep the sensory data and node information. With the redesigned expansion board, the sensor node can easily implement many new functions with new added sensors.

4.3 The Implementation of Sensor node

Our developed sensor node is composed of two parts: main board and expansion board. These two boards are connected through the connector module. The microcontroller module, wireless communication module and power module locates in the main board. The sensor module and some JTAG debug interface locates in the expansion board. The main board is the core of sensor node, it can be connected to some other expansion boards developed according to new requirements. The main board is shown in Fig. 14. In the figure, we indicate the electromagnetic shield cover and the connector module. The electromagnetic shield cover can reduce the electromagnetic interference. The connector module can connect to the expansion board to extend the functions of sensor node.

In Fig. 15, we show out the complete sensor node. The expansion board is put on the right of main board. It has temperature/humidity sensor SHT11, luminance sensor TSL2561, and accelerometer. Through the second development to the expansion board, it is easy to implement the functions to other types of sensors. For example, if we want to add smog sensor into the node, we only need to redesign the expansion board based on the interface functions provided by the connector module, and does not need modify the main board.

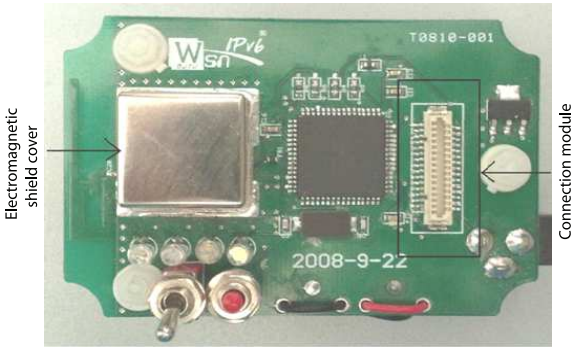


Fig. 14. Main board of the sensor node

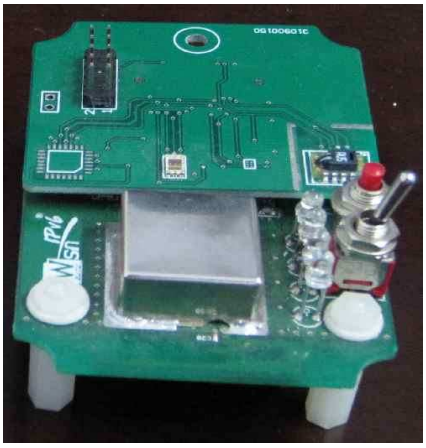


Fig. 15. The picture of sensor node

4.4 The Secondary Development Procedure of the Sensor Node

In our designed general purpose sensor node, it has been configured with temperature/humdity sensor, luminance sensor and accelerometer, which can satisfy the demands of many applications. However, if some applications have special requirements, we need redesign the expansion board according to the characteristics of hardware component and the layout of the connector module. In the designed connector module, it mainly supports SPI, I2C, serial port, I/O ports, with which most of sensors and radio chips can be supported. In the following, we briefly introduce the redesign procedure.

- Choose the hardware components according to the application requirements.
- Determine the interface according to the chosen components and layout of connector module.
- Design the schematic diagram according to the chosen components and interface.
- Design the PCB diagram according to the schematic diagram and the expansion board's requirements.
- Debug the new design modules.

For example, in order to adjust room temperature, we need to add the infrared component into the sensor node to control air conditioner. Firstly, we need to select the type of infrared transceiver according to the power and communication distance. After that, in order to keep the temperature/humidity sensors, we only can choose the free interface from the connector module, which can support connection to the infrared transceiver. In our designed node, we can choose the I/O port for this purpose. Then, we design the schematic diagram and PCB diagram accordingly. Finally, we debug the new design component.

5. Software Architecture of Sensor Node

5.1 The Software Framework of Sensor Node

The software framework of sensor node, which is shown in Fig. 16, is mainly composed of microcontroller software module, wireless communication software module, connector software module, sensor software module. Among them, microcontroller software module is the core of sensor node's software, which includes main process, communication protocol, and some application programs. It is responsible for all software modules' controlling and scheduling. The wireless communication software module is responsible for information exchanging and data delivery between sensor nodes. The connector software module is the key one to help implement the universality and reconfigurability. By the way, power module does not need software to control it so that we do not include it into the software framework. Sensor software module can configure with kinds of sensors to collect information according to application demands. In order to coordinate procedures of different sensors, we also design an adaptive data processing mechanism to work with multiple sensors.

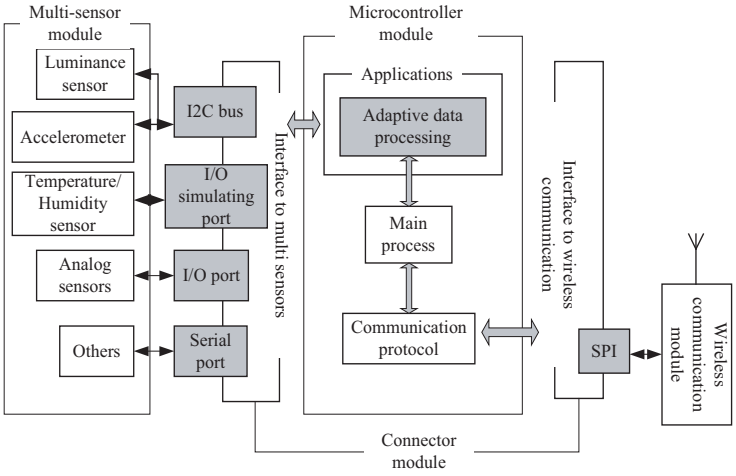


Fig. 16. Software framework of the sensor node

In the following, we firstly introduce some interfaces' operation mechanisms in the connector module. The details of other modules can be referred in Huo et al. (2006). Then, we explain the operation mechanisms of the adaptive data processing.

5.2 Interfaces in the Connector Module

We mainly introduce I2C interface, I/O simulating port, ADC interface, serial port and SPI interface in the connector module.

5.2.1 I2C Interface

In our designed node, the luminance sensor TSL2561 and accelerometer LIS3LV02DQ support I2C bus communications. Following the communication process of I2C bus, the sensor node can complete their sensory data collecting.

We give out the work flow procedure of the sensor TSL2561 as an example. After the sensor TSL2561 starts up, it initializes the I2C bus interface. To prevent the interference from LED lights, it needs to save the LEDs' work status and turn off the LEDs. Then, it sets the control register, time register, interruption register and threshold register of the TSL2561. And it measures the TSL2561 light intensities, compute the actual values of the light intensities. After that, it needs to restore the work status of LEDs. Please note that the control register setting and the light intensity reading is based on I2C time sequence in the communication one by one.

5.2.2 I/O Simulating Port

In our design sensor node, some sensors connect to the main board with same connection pins, but with with different interface technology. For example, the temperature/humidity sensor uses I/O communication technology, but the luminance sensor and accelerometer use I2C bus communication technology and they use the same pins of the connector module. Therefore, after the sensor node uses I2C bus technology to collect some sensory data, it needs to release I2C bus and restore the corresponding pins to I/O port for other type sensory data delivery. As above description, we need to simulate I/O port for the temperature/humidity information collecting. It needs to initialize the I/O port, sets them to three status "high impedance", "input" and "output", and uses the delay technology to simulate the communication time sequences of the sensors. In our sensor node, the data of temperature/humidity sensor SHT11 is communicated with the I/O simulating port. After it is powered, the data port and clock port are set to "output" status. The microcontoller sends the "start working" signal to the sensor SHT11. Then the microcontroller sends a "start temperature measuring" command to the sensor, and after some delay it starts reading "temperature data". Similarly, the sensor node can implement the humidity information collecting.

5.2.3 ADC Interface

For some analog sensors, the analog signals need to change to digital signals. In our sensor node, the input analog voltage in the ADC interface is changed to a 10-bit digital value. Its voltage reference (VREF) determines the ADC conversion range. The minimal value of VREF represents "GND", and the maximal one represents the value that the pin voltage subtracts 1LSB.

5.2.4 Serial Port

The serial port includes time clock generator, transmitter, receiver, and has three lines including "XCK1, TXD1" and "RXD1". "XCK1" is used for the synchronous transmitting mode. "TXD1" and "RXD1" are used for data sending and receiving respectively. After turning on the global interruption, it enables the serial port, enables the data receiving and saves the received data to the data register. Then, it collects the sensory data, writes them to the sending

register, enables the transmission function and sends out them. Finally, close the serial port and global interruption.

5.2.5 SPI interface

The microcontroller ATmega128L connects to radio communication CC2420 with SPI. In order to make sure the correct communication, master and slave devices have to operate in the same mode. In our sensor node, the microcontroller ATmega128L is the master device and the radio transceiver chip CC2420 is the slave device. Because the SPI communication time sequences of CC2420 has been fixed, we need to set the related register of ATmega128 accordingly. The SPI communication between ATmegal128L and CC2420 mainly involves writing and reading the related registers including control and data registers. Among them, it is most important to complete the writing operation to the sending register TxFIFO and the reading operation to the receiving register RxFIFO.

5.3 Adaptive Data Processing

The microcontroller software module is the core of sensor node, which is responsible for controlling, coordinating other modules. It controls the communication with other nodes, and deals with the sensory data locally. The on-site processing of sensory data, which mainly includes two sub-functions, makes adjustment to the data sensing according to environmental situations. One is that it deals with the sensory data considering the variations of measuring values. The second one is that it is able to deal with different sensory data with different priorities specially under some emergency situations.

5.3.1 Detection of Sensory Data Changing

Because normally the luminance and temperature' variation are smooth, it is unnecessary to collect these information frequently and we set the threshold values to decide the information collecting. It can also help save the hardware resource and the sensor node's energy because data delivery through wireless communication module consumes more energy than other modules in the sensor node. With the temperature/humidity sensor as an example, after the sensor nodes starts up, it will collect the sensory data in every one second. Each time, it will compares the current collected data with the last one, if the change is larger than the predefined threshold, it will transmit to the server and save the current one in the sensor node for the next comparison. However, if the sensor node does not send the sensory data up to two minutes, then it immediately sends and saves it without considering the sensory data's change.

5.3.2 Priority Setting for Different Sensors

In our designed node, there are multiple sensors that may collect the environment information at the same time. In this situation, the sensor with the highest priority will get the opportunity to delivery the sensory data to the microcontroller through the connector module. And other sensors will be delayed for some time. However, if one sensor detects an emergency incident, it will immediately get the resource to complete the data delivery in spite of its priority. For example, in our designed node, the accelerometer has the highest priority. But if the temperature/humdiity sensor detects that the temperature is changed significantly, then it will be set to the highest priority because this temperature significant change may indicate that there occurs emergency incidents. After the data is delivered, the priority can be reset to the initial ones.

6. Experiment Results and Analysis

In this chapter, we set up a test-bed to conduct the experiments with our developed sensor nodes. In the following, we firstly illustrate the multiple paths building procedure. Then, we set up a network scenario with an unavailable area simulating the fire disaster and give up the multiple paths building procedure under this situation.

6.1 Multi-Path Building Procedure

The position of sensor nodes is shown in Fig. 17. The communication range of each sensor node is illustrated in the circle with different color. From the figure, the sensor node "1945" can communicate directly with the sink node, the sensor nodes "1946" and "1949". The sensor node "1946" can communicate directly with the sink node, the sensor nodes "1945" and "1949". And the sensor node "1949" can communicate directly with the sensor nodes "1945" and "1946". In the following, we give out the detailed multi-route building procedure of the sensor node "1946" as an example.

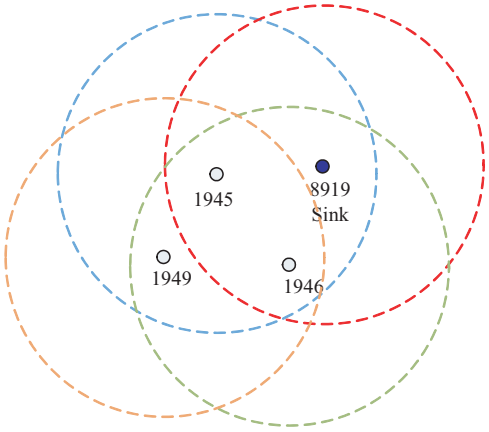


Fig. 17. Experiment scenario of multi-route building procedure

Before the sensor node "1946" wants to send data to the sink node, it needs to build the route to the sink node. Firstly, it broadcasts a RREQ message.

Time (us)	Length	Frame control field								Sequence number	Dest. PAN	Dest. Address	Source Address
		Type	Sec	Pnd	Ack req		Intra PAN						
+3200		DATA	0	0	0		1			0x32	0x2420	0xFFFF	0x0000004792631946
=35240265	41												

MAC payload												LQI	FCS
0C	00	00	20	24	00	19	89	95	22	47	00		
00	00	46	19	63	92	47	00	00	00	FF	01	52	OK

Fig. 18. Flow chart of intermediate nodes receiving RREP

Because the sink node is located within the communication range of the sensor node "1946", it receives the RREQ message and replies a RREP message to it. Fig. 19 shows our the RREP message sent from the sink node to the sensor node "'1946'. After the sensor node receives the RREP message. It will send back an ACK frame to the sink node and the direct communication path between them is built up.

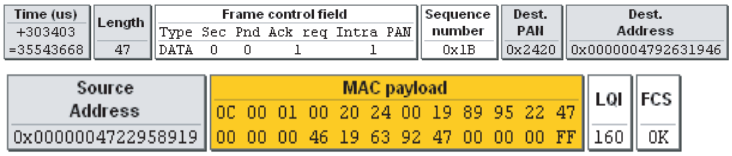


Fig. 19. The RREP message sent from the sink node to the sensor node "1946"

Similarly, because the sensor node "1945" is also located in the communication range of the sensor node "1946", it can receive the RREQ message. In this situation, the sensor node "1945" will forward the received RREQ to the sink node. The forwarded RREQ message is shown in Fig. 20. As shown in the figure, the fields "Type" and "Source address" are still "00" and "1946", but the field "Number of hops" has been changed to "01".

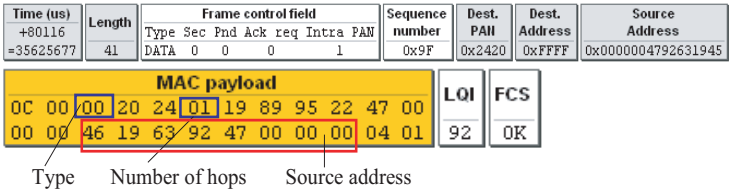


Fig. 20. The RREQ message forwarded by the sensor node "1945"

After the sink node receives the RREQ message forwarded by the sensor node "1945", it will reply to the sensor node "1945" a RREP message, which is shown in Fig. 21. In the figure, the fields "Type", "Number of hops", "Source address" are "01", "00" and "1946" respectively.

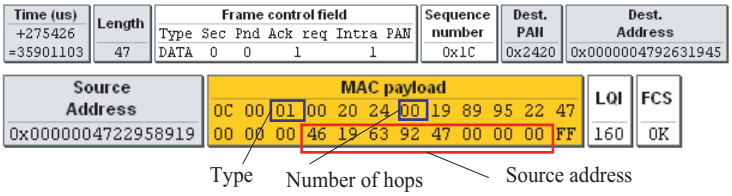


Fig. 21. The RREP message sent from the sink node to the sensor node "1945"

After the sensor node "1945" receives the RREP message, it will send back an ACK to the sink node and check the field "Source address". It finds that it is not the RREP message's final destination and will search the route to the sensor node "1946" in its routing table, which is built up during the RREQ forwarding procedure. After it finds out the corresponding route entry, it will continue forwarding out the RREP message, which is shown in Fig. 22. In the figure, the field "Number of hops" has become "01", and the field "Source address" is still "1946". Also, please note that the field "Address of forwarding node" is "1945".
After the sensor node "1945" receives the forwarded RREP message, it will send back an ACK to the sensor node "1945" and finally build up the second route via the sensor node "1945" between the sensor node "1946" and the sink node. In fig. 23, we give out the final result of building up multiples routes for the sensor nodes "1946" and "1949". The sensor node "1945" firstly starts up and builds a direct communication route to the sink node, which is not shown

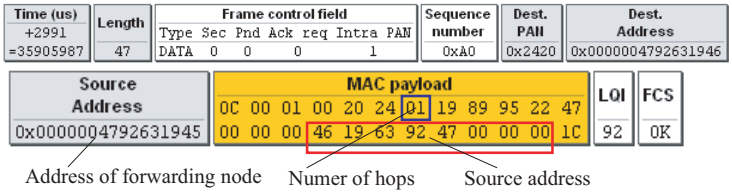


Fig. 22. Flow chart for stage of routing reply

in the figure. Then, the sensor node "1946" starts up and builds up two routes to the sink node, which are shown with the blue solid and dashed lines. Finally, the sensor node "1949" starts up and also builds up two routes to the sink node, which are shown with the red solid and dashed lines.

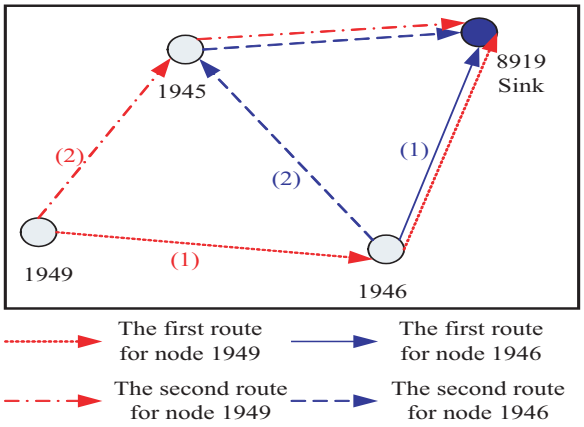


Fig. 23. The experiment result of building up multiple routes

6.2 Multi-Path Building Around the Unavailable Area

The experiment scenario of wireless sensor network is shown in Fig. 24. In the figure, the sensor node "8919" is the sink node. In order to simulate the fire disaster, we set the temperature threshold of some sensor nodes to a lower value. In our experiments, we change the temperature threshold of sensor nodes "1944, 1945, 1946" to 10°C, which is lower than current environmental temperature and the other sensor nodes' temperature thresholds are still 60°C. We start up the sensor nodes "8919, 1943, 1944, 1945, 1946, 1947" one bye one and among them the sensor nodes "1943"∼"1947" and the sink node can communicate with each other directly. After the sensor nodes "1944, 1945, 1946" start up and measure the environmental temperature, they will set them as unavailable sensor nodes because the measured temperature is larger than the predefine threshold value. Following that, they will notify their corresponding neighbor nodes "1943" and "1947". We set that the sensor nodes immediately send the data to the sink node after they start up, thus each sensor node will build their routes one bye one.

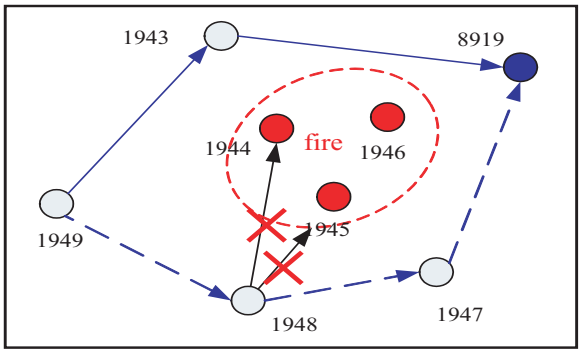


Fig. 24. Experiment scenario of wireless sensor network

The routing table and neighbor node table of sensor nodes “1943” and “1947” are shown in Tabs. 7, 8, 9 and 10. When the sensor node “1943” starts up, there is only the sink node that is working so that it only builds up one route in its routing table. And when the sensor node “1947” starts up, the sensor nodes “1943, 1944, 1945, 1946” and the sink node are working. But among them, the sensor nodes “1944, 1945, 1946” have been set to unavailable nodes so that they can not help forward the data from the sensor node “1947”. Thus, the sensor node “1947” only can build up the second route along the sensor nodes “1947⟶1943⟶8919” besides the direct communication route between it and the sink node.

Node’s ID	Available?
1944	No
1945	No
1946	No
1947	Yes

Table 7. Neighbor nodes table of the sensor node “1943”

Node’s ID	Available?
1943	Yes
1944	No
1945	No
1946	No

Table 8. Neighbor nodes table of the sensor node “1947”

In the following, we change the transmission power of the sensor nodes “1948” and “1949”. The sensor node “1948” can only communicate with the sensor nodes “1944, 1945, 1947”, and the sensor node “1949” can only communicate with the sensor nodes “1943, 1944, 1945, 1948”. And we start up the sensor nodes “1948” and “1949” one by one, their routing tables are shown in Tabs. 11 and 12. From their routing tables, the sensor nodes “1948” and “1949” successfully build the routes avoiding the unavailable areas.

Destination node's ID	Successor node's ID	Number of hops
8919	8919	1

Table 9. Routing table of the sensor node "1943"

Destination node's ID	Successor node's ID	Number of hops
8919	8919	1
8919	1943	2

Table 10. Routing table of the sensor node "1947"

Destination node's ID	Successor nodes's ID	Number of hops
8919	1947	2

Table 11. Routing table of the sensor node "1948"

Destination node's ID	Successor node's ID	Number of hops
8919	1943	2
8919	1948	3

Table 12. Routing table of the sensor node "1949"

7. Conclusions

In this chapter, we designed a new multi-path routing protocol, MSMRP, to cross around the unavailable areas based on our previously proposed MSRP routing protocol. In particularly, we design a neighbor node table exchanging mechanism that can help build an alternate route around the unavailable areas and try to avoid the multiple paths intersect. When a RREQ is arriving at some unavailable sensor nodes, they will not forward it so that these sensor nodes will not be included into the inverse routes from the sink to the source node. When a sensor node becomes unavailable during the RREP forwarding procedure, its precursor node will try to find the alternate route to forward the RREP to the destination. Finally, we implement the proposed protocol in the real sensor nodes and set up a testbed to conduct detail experiments. The experimental results show that MSMRP can perform well as we expect.

8. Acknowledgments

The authors gratefully acknowledge the support by "the Fundamental Research Funds for the Central Universities" under grant No. 2009JBM007, the support of the project-sponsored by SRF for ROCS, SEM under grant No. [2008]890, and the support of the National Natural Science Foundation of China (NSFC) under Grant No. 60802016, 60972010 and 60833002.

9. References

Chen, Y. & Nasser, N. (2008). Enabling QoS multipath routing protocol for wireless sensor networks, *Proc. of IEEE International Conference on Communications (ICC'08)*, Beijing, China, pp. 2421–2425.

Deering, S. & Hinden, R. (1998). Internet protocol, version 6 (IPv6), specification, RFC 2460.

Fang, Q., Gao, J. & Guibas, L. J. (2006). Locating and bypassing holes in sensor networks, *Springer Mobile Networks and Applications* **11**(2): 187–200.

Ganesan, D., Govindan, R., Shenker, S. & Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks, *ACM SIGMOBILE Mobile Computing and Communications Review* **5**(4): 11–25.

Gao, D., Niu, Y. & Zhang, H. (2009). Micro sensor routing protocol in IPv6 wireless sensor network, *Proc. of IEEE International Conference on Networking, Sensing and Control (ICNSC'09)*, Okayama, Japan, pp. 55–59.

Huang, X. & Fang, Y. (2008). Multiconstrained QoS multipath routing in wireless sensor networks, *Wireless Network* **14**(4): 465–478.

Huo, H., Zhang, H., Niu, Y., Gao, S., Li, Z. & Zhang, S. (2006). MSRLab6: An IPv6 wireless sensor networks testbed, *Proc. of the 8th International Conference on Signal Processing*, Vol. 4, Guilin, China.

Jafarian, M. & Jaseemuddin, M. (2008). Routing of emergency data in a wireless sensor network for mines, *Proc. of IEEE International Conference on Communications (ICC'08)*, Beijing, China, pp. 2813–2818.

Jennifer Yick, Biswanath Mukherjee, D. G. (2008). Wireless sensor network survey, *Computer Networks* **52**: 2292–2330.

Khan, I. & Javed, M. (2008). A survey on routing protocols and challenge of holes in wireless sensor networks, *Proc. of International Conference on Advanced Computer Theory and Engineering (ICACTE'08)*, Pukhet, Thailand, pp. 161–165.

KIM, M., JEONG, E., BANG, Y.-C., HWANG, S., SHIN, C., JIN, G.-J. & KIM, B. (2008). An energy-aware multipath routing algorithm in wireless sensor networks, *IEICE Transactions on Information and Systems* **E91-D**(10): 2419–2427.

Lee, S.-J. & Gerla, M. (2001). Split multipath routing with maximally disjoint paths in ad hoc networks, *Proc. of IEEE International Conference on Communications (ICC'01)*, Helsinki, Finland, pp. 3201–3205.

Marina, M. & Das, S. (2001). On-demand multipath distance vector routing in ad hoc networks, *Proc. of the 9th International Conference on Network Protocols (ICNP'01)*, California, USA, pp. 14–23.

Nasser, N. & Chen, Y. (2007). Secure multipath routing protocol for wireless sensor networks, *Proc. of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, pp. 12–12.

Teo, J.-Y., Ha, Y. & Tham, C.-K. (2008). Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming, *IEEE Transactions on Mobile Computing* **7**(9): 1124–1137.

Toledo, A. & Wang, X. (2006). Efficient multipath in sensor networks using diffusion and network coding, *Proc. of the 40th Annual Conference on Information Sciences and Systems (CISS'06)*, Princeton, NJ, pp. 87–92.

7

Relation-based Message Routing in Wireless Sensor Networks

Jan Nikodem, Maciej Nikodem, Marek Woda and Ryszard Klempous
Wroclaw University of Technology
Poland

Zenon Chaczko
University of Technology Sydney
Australia

1. Introduction

Sensor networks and their related topics represent some of the greatest and challenging possibilities in the research field that have come about in recent years. Emerging technologies like wireless sensor network (WSN), standards enabled legacy sensors, ubiquitous and cloud computing, middleware, communication systems, internet protocols (IP) and next generation networks are leading to a set of new paradigms where wireless sensors can be treated as vital components of common infrastructure and a shared resource with an ability to serve multiple and concurrently executing applications run by various users in distributed environment. This is in strong contrast to traditional concepts where dedicated sensor devices are being physically and logically hard-wired to communication and computing infrastructure serving very specific and dedicated data/information processing applications. Wireless sensor networks consist of a number of small electronic devices (nodes) distributed in an area that by far exceeds the communication range of a single sensor. Message routing is one of the most important issues in such networks. This is mainly due to the large number of nodes, variety of possible communication paths, restricted power source and variability (in time and space) of environmental conditions in which the WSN operates. A shared communication channel and restricted communication ranges require that nodes of the WSN cooperate and/or coordinate their actions while messages are routed from nodes to the base station (BS). It is a well-known idea Descartes & Lafleur (1960) to solve large and complex problems by dividing them into smaller and possibly simpler tasks. The most crucial element of such an attempt is to decide how to divide the problem in order to get a problem that can be solved efficiently and, what is more important, can be used to find a solution to the original problem.
A distributed system, such as a WSN, is traditionally seen as a set of spatially distributed nodes that communicate, coordinate their actions and inform other nodes about their status using special messages sent over the communication channel Dollimore et al. (2005). Such a system is usually assumed to be isolated from the outside word – even if it measures its parameters and/or listens to the status messages from other nodes, it is still not affected by the environmental conditions and its changes. We are going to look at such a system as it

consists of independent elements that adjust their actions according to the actual situation in their neighbourhood and WSN in order to achieve globally defined goals. Surroundings of each node is composed of two elements:

- neighbourhood – a set of WSN elements (i.e. nodes) that are in the surroundings of the node,
- environment – a set of elements that are in the surroundings of a node, influence its behaviour, but are not elements of the WSN.

The above approach enables to describe any system as an open system in which communication activity adapts to stimulus that originates both from system elements as well as from environment. This enables the system to respond to harsh and unpredictable situations that may be beneficial in many applications.
When analysing WSN it is important to capture four related components:

- independence of WSN elements,
- cooperation, and
- communication between WSN elements,
- interaction of elements with the environment.

This chapter focuses on communication in a distributed system such as a WSN. The main purpose of communication in WSNs is to retransmit messages and route them to the base station. Our interest in distributed WSNs is not only due to spatial distribution of nodes of the network, but also due to the fact that decisions on message routing and communication paths are taken in distributed manner as an effect of cooperation between nodes.

2. The relational model of communication in WSN

Investigation of activities in wireless sensor network can be based on relational model of cooperation between nodes of the network Nikodem (2009). Relational model captures dependencies (relations) between nodes of the network and defines actions that nodes may take in different situations. Actions are taken by every node individually, so the relational model can be used to describe independent elements that cooperate within the network in order to achieve globally defined goals. since the relational model reflects the nature of real WSNs, therefore, it includes all previous proposals to efficient network organisation, communication and routing. Moreover, it enables to construct new algorithms that will achieve globally defined goals through local actions taken be each node of the network.

2.1 Actions and relations between nodes

Communication activity in WSN can be described using three binary relations defined over set of actions - Act. Set of actions contain all activities that can be carried out by every node of the network individually but with respect to other nodes and environment (i.e. situation in the neighbourhood). Ability to execute specific action depends on the state of the node (e.g. network establishment actions, network management, and normal operation) and its execution cause the state of the node to be changed. Therefore, all actions that can be taken by nodes of the WSN are defined over a Cartesian product of set of nodes *Nodes* and set of all possible states *States*:

$$\text{Act} : \mathit{Nodes} \times \mathit{States} \rightarrow \mathit{States}. \tag{1}$$

Measurement of environment parameters, data aggregation, sending messages to a single (or group) node and message receiving are examples of actions that can be taken by nodes. Since nodes are autonomous, therefore each node can execute actions independently from other nodes. Undoubtedly, this is an advantage since this enables nodes in a different part of the WSN to perform various (possible related) actions simultaneously. On the other hand a number of actions become essential only when two or more nodes cooperate. In such situation cooperation requires that nodes execute actions that are related which is formally denoted as:

$$a_i^{(1)} \mathscr{R} a_j^{(2)}, \tag{2}$$

where $a_i^{(1)}, a_j^{(2)} \in \text{Act}$ and $a_i^{(k)}$ denotes i-th action that is executed by k-th node. Eq. (2) should be read as: *action* $a_i^{(1)}$ *is in relation with* $a_j^{(2)}$. Relations are used to determine actions that are related to each other and are either executed together (but not necessarily in the same time instant) or cannot be executed together. Since nodes can execute a vast number of actions that can be part of different relations, therefore relations have their names and symbols. Since this chapter focuses on communication, therefore, we will only consider communication related relations and simplify the notation. From now on sending and receiving a message will be denoted as x where x is an ID of node that is either sending or receiving the packet. Whether node x sends or receives the message will arise from the context or will be explained in the text.

To describe a variety of possible dependencies between different elements of real world WSN it is enough to define three elementary relations Jaron (1978); Nikodem (2008):

- subordination - π,
- tolerance - ϑ,
- collision - κ.

When message sending and receiving actions are considered then subordination

$$x_{\mathrm{R}} \pi y_{\mathrm{S}}, \tag{3}$$

means that node y receives data whenever node y send it. Subordination is transitive which means that if x is subordinated to y and z is subordinated to x then z is also subordinated to y:

$$x_{\mathrm{R}} \pi y_{\mathrm{S}} \text{ and } z_{\mathrm{R}} \pi x_{\mathrm{S}} \Rightarrow z_{\mathrm{R}} \pi y_{\mathrm{S}}. \tag{4}$$

Subordination is antisymmetric which means that if x is subordinated to y then y is not subordinated to x:

$$x_{\mathrm{R}} \pi y_{\mathrm{S}} \Rightarrow \neg \left(y_{\mathrm{R}} \pi x_{\mathrm{S}} \right). \tag{5}$$

We can define a set Π of pairs of nodes of the WSN that are in subordination relation. This set consist of ordered pairs of nodes such that:

$$\Pi = \{ \langle x, y \rangle \mid x_{\mathrm{R}}, y_{\mathrm{S}} \in \text{Act and } x_{\mathrm{R}} \pi y_{\mathrm{S}} \} \tag{6}$$

When dealing with communication tolerance relation between nodes x and y

$$x_{\mathrm{R}} \vartheta y_{\mathrm{S}} \tag{7}$$

means that x may receive messages send by node y. When x tolerates y then it is less likely that node y sends messages to x - y prefers subordinated nodes. Nevertheless, y may send message

to x and whenever this happens node x will receive the message and route it towards the base station. In contrast to previous relation tolerance is symmetrical

$$x_R \vartheta y_S \Rightarrow y_R \vartheta x_S, \tag{8}$$

but is not transitive. Subordination and tolerance relation can be composed - if x is subordinated to y and z tolerates x then z also tolerates y:

$$x_R \pi y_S \text{ and } z_R \vartheta x_S \Rightarrow z_R \pi y_S. \tag{9}$$

Set of all nodes that tolerate each other is a set of pairs $< x, y >$ such that

$$\Theta = \{\langle x, y\rangle \mid x_R, y_S \in \text{Act and } x_R \vartheta y_S\}. \tag{10}$$

It follows From the definition of relations π, ϑ for corresponding sets of subordinated and tolerated nodes (Π and Θ) that

$$\Pi \subseteq \Theta. \tag{11}$$

The final relation that we need to consider is the relation of collision which for data transmission and reception activities in WSN specifies all those sensor nodes that don't exchange messages among themselves. The relation of collision between node x and node y is denoted as:

$$x_R \kappa y_S, \tag{12}$$

The above relation takes place when a node x does not receive any messages transmitted by the node y, including both broadcasted and explicitly addressed messages from the node y to the node x. Additionally, if a node x has a collision relation with node y and the node z is subordinated to the node x, then the node z has also a collision relation with the node y. This can be expressed as:

$$x_R \kappa y_S \wedge z_R \, \pi \, x_S \Rightarrow z_R \kappa y_S. \tag{13}$$

The relations of tolerance and collision are mutually exclusive therefore only those WSN nodes that are not in a relation of tolerance can remain to stay in a relation of collision. Hence, if we denote a set of nodes that remains in relation of collision as:

$$\mathrm{K} = \{< x, y > \mid \; x_R, y_S \in Act \wedge x_R \kappa y_S\}, \tag{14}$$

then the sets of nodes that remain in relations of tolerance and collision meet the following criterion:

$$\Theta \cap \mathrm{K} = \varnothing. \tag{15}$$

Since, $\Pi \subseteq \Theta$ also applies therefore sensor nodes that are both in relation of tolerance and subordination cannot remain in relation of collision

$$\Pi \cap \mathrm{K} = \varnothing. \tag{16}$$

2.2 Neighbourhood, Neighbouring and Environment

When studying multi-hop communication in sensor networks it is not possible to omit such specific aspects of WSN as sensor node cooperation. Due to limited radio range, the majority of nodes in the WSN are not able to transmit data directly to the base station, therefore the nodes have to rely on the mechanism of retransmission offered by other nodes in their surroundings. Wireless sensor networks are truly distributed systems where sensor nodes characterised by limited communication resources cooperating among each other in order to

support the network infrastructure activity as a whole. Events and entities that are in the perception range of a node belong to its surroundings. In the set that we call the surroundings we can identify two distinctive subsets: *neighbourhood* and *environment*. Considering that we are interested in communication aspects of WSN, the concept of *neighbourhood* is particularly significant; hence we dedicate this concept in our further investigation.
In WSN related literature the definition of neighbourhood is frequently used, often becoming a basis for the definition of several routing algorithms Braginsky & Estrin (2002); Burmester et al. (2007); Manjeshwar & Agrawal (2001); Younis & Fahmy (2004). Let us begin from explaining the meaning of $\mathrm{Map}(\mathrm{X}, \mathrm{Y})$ expression that can be defined as a collection of mappings of set X onto set Y (surjection). Let us define $\mathrm{Sub}(\mathrm{X})$ as a family of subsets X and the neighbourhood as

$$\mathrm{N} \in \mathrm{Map}(Nodes, \mathrm{Sub}(Nodes)). \tag{17}$$

Furthermore, if $\mathrm{N}(x)$ belongs to neighbourhood of the node x and $\mathrm{N}(S)$ is a neighbourhood of the set S of nodes then, using the neighbourhood relation (here denoted as η) we can define a collection of nodes which are neighbours of the given node x as:

$$\mathrm{N}(x) = \{y \mid y \in Nodes \wedge x \eta y\}, \tag{18}$$

and denote the set of neighbours of all nodes that belong to the set S as:

$$\mathrm{N}(\mathrm{S}) = \{y \mid y \in Nodes \wedge (\exists x \in \mathrm{S} \mid x \eta y)\}. \tag{19}$$

In this discussion, we assume that the neighbourhood relation is a symmetric:

$$x \, \eta \, y \Rightarrow y \, \eta \, x. \tag{20}$$

This implies, that if a node x remains in a neighbourhood relation with y (i.e. x is able to communicate with y) then the node y is also in a neighbourhood relation with x.
In WSN literature several various locality models were proposed Nikodem et al. (2009). Various benefits and drawbacks of sensor node clasterization or unique transmission paths in context of the applied definition of neighbourhood are also discussed. However, the most accepted approach for defining the locality is the one based on the concept of the neighbourhood that is derived from the technological limitation of radio communication. In some specific situations the partitioning of network into clusters can be very beneficial, to a degree this can be seen as an oversimplification that makes our computation much easier. However, the trade-off is a reduction of the solution space. In the case of a singular retransmission path the solution space consists only of one element.
Let us consider the neighbourhood family of $N = \{\mathrm{N}_i \mid i \in \mathrm{I}\}$ for which the following conditions are met:

$$(\forall i \in \mathrm{I} \mid \mathrm{N}_i \neq \varnothing)(\cup_i \mathrm{N}_i = Nodes), \tag{21}$$

$$(\exists^{\succ} i, y \in \mathrm{I} \mid i \neq j)(\mathrm{N}_i \cap \mathrm{N}_j \neq \varnothing). \tag{22}$$

This translates onto a local mode (for each node) and takes the form of:

$$(\forall y \in Nodes)(\exists^{\succ} i \in \mathrm{I} \mid y \in \cap \mathrm{N}_i \neq \varnothing). \tag{23}$$

The expression $\exists^{\succ}$ can be translated as: "there are as many instances as the structure of the network allows for". The neighbourhood obtained when taking this approach can be interpreted as the most natural of all possible instances that can also guarantee the maximum retransmission capabilities for all allowable solutions.

Referring back to concepts of surroundings (S), neighbourhood (N) and environment we could observe that:

$$((N \subset WSN) \wedge (E \nsubseteq WSN)) \wedge ((N \cup E \subset S) \wedge (N \cap E = \varnothing)). \tag{24}$$

Neighbourhood is a collection of all of the neighbouring surrounding that belong to the WSN, while the environment (E) consists of all of the elements of surrounding that do not belong to the wireless sensor network but that do have an effect on its behaviour.

2.3 Forming Actions - Chains

Relational dependencies of the chain functions of WSN in most cases describe connections between the neighbouring nodes and adapt general principles of neighbourhood. To deal with it, let us focus on the relation of subordination π. Out of all four relations only this one is transitive, which allows us to model the retransmission paths. The π relation which is both transitive and reflective, forms a preorder in the set of actions Act (1). Further investigation requires a stronger order of the set of actions Act. Introducing a partial order does not appear difficult. In real time applications, nodes are distributed more or less randomly over a given area (i.e. they may be dispersed out of a aeroplane). In the case where two network, nodes are found very close to each other, one of them becomes tacit (mute) and in reserve. In this way, a singular communication node substantially greater robustness and survivability is formed. In mathematical terms, such "binding" of two elements can be expressed as:

$$(\forall x, y \in \mathrm{Act})(x \,\pi\, y \wedge y \,\pi\, x) \Rightarrow (y = x). \tag{25}$$

The above expression shows that subordination happens asymmetrically which in turn may lead to a partial set order (asymmetric preorder). Therefore, the set of actions Act is partially ordered (poset). In the discussed formal apparatus we have a stronger relationship than the one indicated in expression (25). As shown in (5), the subordination relation is of an antisymmetric nature; hence this is equivalent to irreflexivity as every relation that is antisymmetric is both asymmetric and irreflexive. Indeed in WSN, a situation when a sensor node transmits to itself does not belong to a category of logical behaviours. Irreflexivity put together with transitivity provides a strict partial order. The set of actions Act being finite and partially ordered can be represented in many ways as any two argument relation can be represented in a form of a directed graph or a diagram. For such a graphic representation we can use the Hasse diagrams which can help us to show the subordination relation between pairs of elements and the whole structure of partial classification of the set of actions. Although, the Hasse diagrams are simple and very intuitive tools for dealing with finite posets, it turns out to be a difficult task to draw "clear" diagrams for more complex situations when we try to represent all possible communication links in the structure of WSN. In most cases when we apply the Hasse technique by first drawing a graph with the minimal elements of an order and then incrementally adding other missing elements we may end-up producing rather poor and unreadable diagrams where internal structure and symmetries of the order are no longer present due to a large number of connections. Therefore, we need to search for a better solution. Our approach using the relations may in the future lead to more viable solution for the representation of connectivity in WSN.

In multi-hop sensor networks, the subordination relation that reflects communication aspects of WSN, is not a relation that is cohesive or finite. This means that:

$$(\exists x, y \in \mathrm{Act})(\neg(x \,\pi\, y \vee y \,\pi\, x)), \tag{26}$$

hence, there are elements for which such a relation does not take place, thus the subordination relation can be described as a set of a partial order (poset). It is possible to select subsets of such a set that are linearly ordered so that the partial order will additionally meet the condition of cohesion such as:

$$(\forall x, y \in \mathrm{Act})(x \,\pi\, y \vee y \,\pi\, x). \tag{27}$$

It needs to be noted that this expression contradicts the previous one. In multiplicity and partial order theories, ordered subsets for which the order relation is found to be cohesive are called chains. To form the chain we shall define the subordination relation setting by the following induction:

$$\pi^n = \pi^{n-1} \circ \pi, \ldots \pi^2 = \pi^1 \circ \pi, \pi^1 = \pi. \tag{28}$$

Hence

$$\pi^n = \{< x, y > | < x, y > \in \mathrm{Act}^{n-1} \times \mathrm{Act}\}, \tag{29}$$

where x shall be called the direct successor of y, while y will be called direct predecessor of x. Forming the communication activities in multi-hop WSNs is the fundamental problem because there is a question whether messages from the network area can be passed onto the base station. On a global scale (this involves the whole WSN), to build suitable structure that allows us to find the answer for this question we could draw on a concept from the theory of multiplicity - transitive closure of 2-argument relation of subordination π on the set Act. However, in this work, the problem of solving the communication activity is perceived from the local level (node neighbourhood). Therefore, we shall consider a case when a packet is transmitted from the node y and after certain number of retransmissions should reach the base station (BS). Applying the setting of subordination relation π, for each sensor node y we define sets of its ascenders Asc and descenders Des using the following expressions:

$$\mathrm{Asc}_\pi(y) = \{z \in \mathrm{Act} \mid (\exists n \in \mathrm{N})(y\pi^n z)\}, \tag{30}$$

$$\mathrm{Des}_\pi(y) = \{z \in \mathrm{Act} \mid (\exists n \in \mathrm{N})(z\pi^n y)\}. \tag{31}$$

Expressions (30),(31) define sets with full communication space of the node y. One of our main aims, however, is to find an answer to the question "to whom send a packet in open space?" hence we need to pay more attention to the set Des_π. It is worth to notice, that for a packet to arrive from the sensor node y to its destination at the node BS it is necessary for the base station to be one of the elements of the set Des_π. Additionally, we could form many subsets of the Des_π set and some of these subsets may help us to determine communication activity in WSN. Among the subsets of Des_π, we can distinguish two types of subsets:

- four sets that are partially ordered, and
- family of well-ordered chains (linearly ordered sets)

The selected ordered chains can be defined as:

$$\mathrm{Des}_\pi^{min}(y) = \{x \in \mathrm{Des}_\pi(y) \mid \mathrm{BS}\,\pi\, x)\}. \tag{32}$$

The subset (32) contains the selected nodes that are the direct ascenders of the base station (BS). Hence, only the retransmission that involves these nodes allows the packets sent from the node y reaching the BS. The power of this set determines the maximum number of packets that can be delivered from the node y to the BS. Second subset

$$\mathrm{Des}_\pi^{max}(y) = \{x \in \mathrm{Des}_\pi(y) \mid x\,\pi\, y)\}, \tag{33}$$

contains the nodes that are direct followers of the y node, in other words, these are the nodes that are required to execute the retransmission of the packet issued from the node y. The power of this set determines the maximum number of packets that could be sent from the node y to the BS. Third subset

$$\mathrm{Des}_{\pi}^{mis}(y) = \{x \in \mathrm{Des}_{\pi}(y) \mid \neg(\exists n \in \mathrm{N})(\mathrm{BS}\pi^{n}x)\}, \tag{34}$$

contains nodes that become the dead end on the paths to the base station. A packet that arrives at such a node does not have even a chance to reach the BS. The last subset

$$\mathrm{Des}_{\pi}^{pfex}(y) = \{x \in \mathrm{Des}_{\pi}(y) \mid \mathrm{Card}(\mathrm{Asc}_{\pi}(x)) > 1\}, \tag{35}$$

is made up of nodes called pontifixes that are located at intersections of the packet routes. These nodes become the bottlenecks on the routing path from node y to the BS. Skilful shaping of the communication activity allows for the best utilisation of these elements. The power of the set of pontifixes defines the capability of packet to escape from one routing path onto another during the retransmission to the base station (BS).
From the perspective of shaping the communication activity in WSN, the second most interesting subset group $\mathrm{Des}_{\pi}(y)$ represents a family of chains $\mathrm{Chn}_{\pi}(y)$ that constitutes linearly ordered subsets. For each iteration of the $\mathrm{Chn}_{\pi}^{i}(y)$ chain the following condition applies:

$$(\forall \mathrm{Chn}_{\pi}^{i}(y) \subset \mathrm{Des}_{\pi}(y) \mid i \in \mathrm{I})(\mathrm{BS} = \perp \wedge\ y = \top), \tag{36}$$

where the symbol $\perp$ denotes the smallest element BS and the symbol $\top$ denotes the biggest element (y).

3. Communication towards base station

In a wireless sensor network nodes are responsible for the collection of information (individual action) and forwarding them to the base station (collective action). As it was described previously, such action may be described using the three relations - subordination, tolerance and collision. Subordination relation is particularly important, because of its transitivity and asymmetry, and it was used in the developed simulator.
At first, let's consider subordination relation only and suppose that the node x is a source of information. Then, the set $\Pi(x)$ contains all nodes, to which x can send messages directly. Using the subordination relation, a node that receives the information is able to forward it to its neighbors that are in the subordination relation with it. Therefore, we can define the set of the node descendants that contains all nodes to which the message may be sent to:

$$\mathrm{Des}_{\pi}(x) = \{y \mid (\exists n \in \mathrm{N})(y\pi^{n}x)\}, \tag{37}$$

where $y\pi^{n}x$ indicates that there are n intermediate nodes $y^{(i)}$ such that one can build a chain of relationships

$$y\pi y_n, y_n\pi y_{n-1}, y_{n-1}\pi y_{n-2}, \ldots, y_2\pi y_1, y_1\pi x. \tag{38}$$

When subordination reflects direction towards the base station then it is ensured that the base station belongs to the set $\mathrm{Des}_{\pi}(x)$ for each node x. Therefore, each message generated by x will eventually reach the base station. Moreover, it follows from the properties of subordination relation and the fact BS belongs to $\mathrm{Des}_{\pi}(x)$ that a message sent from the node x and retransmitted to subordinated nodes, always reaches the BS (assuming that all nodes on the

communication path have enough energy). This is due to transitivity property of the subordination relation and the fact that in chain of relationships (38) we have $y_i \pi x$ for every $i = 1, 2, \ldots, n$ and $y_i \pi y_j$ for any $i > j$. Since the subordination relation is asymmetric, so in the chain of relationships each node occurs only once - otherwise, if

$$y_{i+1} = y_j \tag{39}$$

for some $i > j$ then from the fact that $y_{i+1} \pi y_i$ follows that $y_j \pi y_i$. However, since $i > j$ therefore $y_i \pi y_j$ and so the relation becomes symmetric which contradicts the assumption (5). This means that in sequence (38) each node can occur only once. Therefore, and due to the fact that $\mathrm{Des}_\pi(x)$ is finite and contains the BS follows that for every x there exists a finite subordination relationship chain that leads to the BS, i.e.:

$$\mathrm{BS} \pi y_n, y_n \pi y_{n-1}, y_{n-1} \pi y_{n-2}, \ldots, y_1 \pi x. \tag{40}$$

The above property results directly from the definition of set $\mathrm{Des}_\pi(x)$ that includes only these nodes that are closer to the BS than node x. As a consequence each node y_i in relationship chain (40) is closer to the BS then x and y_j for any $i > j$. If x is located in the communication range of the BS then set $\mathrm{Des}_\pi(x)$ is a singleton that consists only of the BS. Set $\mathrm{Des}_\pi(x)$ consists of a number of nodes y that are subordinated to x and, in connected networks (i.e. networks in which each node can communicate directly or using retransmission with BS), constitute one or more relationship chains. These chains may differ in number of elements but always lead to the BS.
Similar properties do not hold for tolerance relation since BS does not necessarily belong to

$$\mathrm{Des}_\vartheta(x) = \{y \min(\exists n \in \mathrm{N})(y \vartheta^n x)\}. \tag{41}$$

Moreover, there is no guarantee that $y_i \neq y_j$ in the tolerance relationship chain

$$y \vartheta y_n, y_n \vartheta y_{n-1}, \ldots, y_2 \vartheta v_1, y_1 \vartheta x \tag{42}$$

for any combination of $i \neq j$. This is a direct consequence of symmetry property that may lead to loops in chain where part of the chain begins and ends with the same node, e.g.

$$x \vartheta y_n, y_n \vartheta y_{n-1}, \ldots, y_2 \vartheta y_1, y_1 \vartheta x. \tag{43}$$

As a result tolerance relationship chain may be infinite even if $\mathrm{Des}_\vartheta(x)$ is always finite (since number of nodes in the network is finite).
Above considerations present that tolerance relation itself is not sufficient to guarantee that cooperation within the sensor network will lead to proper routing of messages (i.e. that messages will reach the base station). However, tolerance has features that make it very useful as an auxiliary to the subordination. In real life application of WSN it may be particularly useful to cope with locality effects - this corresponds to situations when divided problems cannot be solved or does not improve the overall result. Similarly in WSN, tolerance will enable routing paths variation in order to prevent message loss (e.g. due to dead ends). In an extreme case tolerance relation (that is symmetric) may force a node to send a packet back to its ascender in order to find an alternative routing path. The combination of subordination and tolerance relations allows drawing on advantages of both relations. Subordination ensures that messages always reach the base station while tolerance increases by far the number of available communication paths.

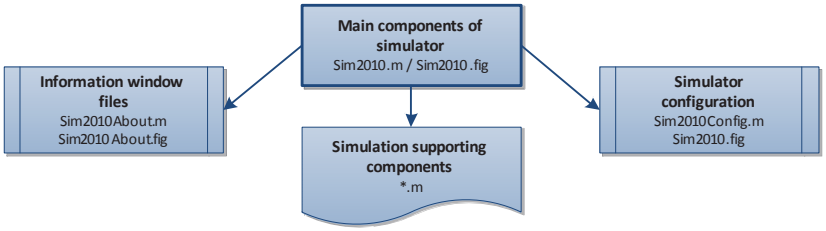


Fig. 1. Simulator architecture - key components

4. Simulation of WSN communication behaviour

In order to present the relational approach that can model behaviour and operation of WSN we have developed a network simulator. Our simulator models behaviour of every single sensor that operates independently in order to meet globally defined criteria and with respect to situation in its environment.

4.1 Simulator

The MATLAB environment is required for set up and proper operation of the simulator. The simulator was written and had been tested in MATLAB version R2009b. Only the basic features of the MATLAB environment were used, so no additional tool kits (Toolboxes) are required. The architecture of the simulator is presented in Fig. 1. The entry point of the simulator is `Sim2010.fig` file which starts the simulator GUI.

Work with the simulator Fig. 2 starts from parameters being setup (Phase I). This includes such parameters as network size, number of sensors, etc. This stage is surmounted by the deployment of sensors in the defined working area, visible in the visualisation area of the main simulator.

The first action undertaken in Phase II, is the selection of one of the seven algorithms available in the simulator. Then, depending on the choice made, one can change the default parameters of the algorithm. At this stage, one can also decide how to present the results of simulation and its detail by setting additional parameters in the configuration window.

Approval of the configuration changes made in this step allows for the transition to Phase III. Simulation begins when the RUN SIM button is pressed. From that moment, the simulation runs, with time as well as simulation results/parameters being visualised on the screen in the form of graphs and numerical results. Simulation can be also saved to an AVI file. During the simulation, a user can control it (stop and resume it), using the simulation control panel or configuration window to change the appearance of visualisation window. Completion of the simulation process ends PHASE III. Pressing the RESET DATA button, allows the user to jump back to the first stage and resuming simulation from the beginning (possibly with new parameters). Fig. 3 shows the main window of the simulator in which the basic parameters of WSN are defined, the simulation is visualised and information about the current state of the simulation (number of loops, number of messages etc.) and values of network parameters

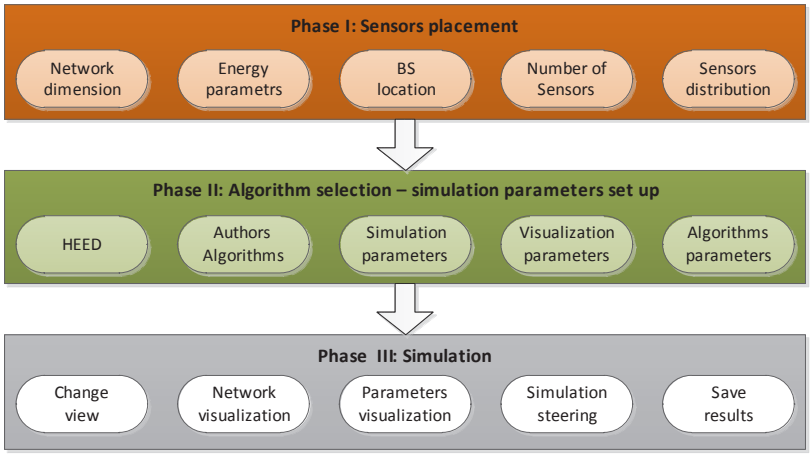


Fig. 2. Main simulation phases

(the average cost of energy, lifespan, etc.) are presented. The basic network parameters that can be entered by the user are:

- size of the network and its area - it is determined by defining a rectangular area in which WSN nodes will be deployed. Because one of the vertices of the area is permanently located at the point (0,0) this area is defined by specifying the length of two sides of the rectangle along the X and Y axis ("Network Size" field). It should be noted that currently the simulator operates for a two-dimensional network, which means that it is not possible to determine the size of the nets along the Z axis.
- position of the base station - we have assumed that there is only one base station in the simulated WSN that can be located at any point of the network area. It is common to place the base station in a corner of the area which is the worst possible position
- number of parameters and sensors - simulator allows to control such network parameters as the number of WSN nodes deployed ("Sensor - Number"), the maximal communication range of a single node ("Sensor - Range") and initial energy of each node ("Sensor - Energy"). In determining the number of nodes and their maximum range, one has to remember that these parameters are related to the size of the network. Setting too few nodes, or too short communication range can cause the network to be disconnected (some nodes of the network will not be able to communicate with the base station). Given network area (P) and the maximum communication range of a node (R_t), the number of nodes required to ensure network is connected, can be estimated. Note that if in each circular area of the diameter $R_t/2$ at least one node is located then any two nodes located in two adjacent areas will be always able to communicate directly. This will be ensured regardless of their position within this area. Since the entire network area is rectangular, we assume that the area of $R_t/2$ diameter can be approximated by a

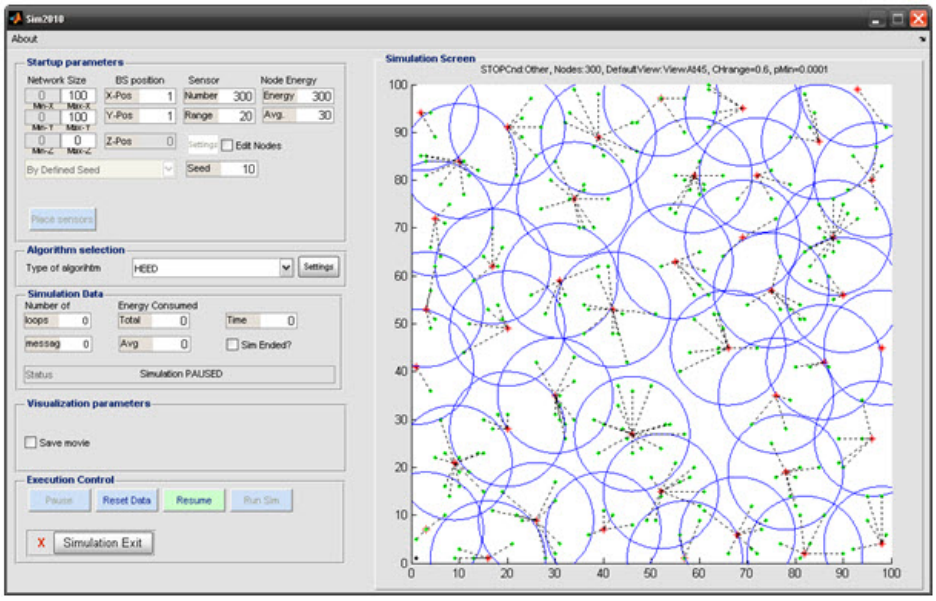


Fig. 3. Main simulator window

square of diagonal $R_t/2$, inscribed in the circle. If so then the number of areas that will fit on the entire network is equal to

$$N = \frac{R_t^2}{2\sqrt{2}P}. \tag{44}$$

Once the number of areas is known, one can estimate the number of nodes to be scattered in the network that ensures each of N areas is covered with at least one node. This problem is equivalent to the ball-and-bins problem in which balls are thrown randomly to bins, which is the well-known in mathematics. It was presented that when

$$n = 2N \log N = \frac{R_t^2}{\sqrt{2}P} \log\left(\frac{R_t^2}{2\sqrt{2}P}\right), \tag{45}$$

nodes (balls) are used then the probability that there is at least one node (ball) in each area (bin) is close 1.0. It should also be noted that this estimate is inflated due to the assumption that the area covered by communication range of a single node is square rather than circle.

In addition to these parameters, the user can also influence the arrangement of nodes in the network. The simulator assumes that nodes are distributed evenly throughout the network (which is the assumption commonly adopted in the literature), however, one can control this distribution by identifying the seed used to generate sequences of random numbers. Using the drop-down list one can specify if the distribution of nodes should be completely random, or random with a seed that is entered by a user - in that case one must select "By Defined Seed" and enter the value of seed in the "Seed" window. Because of this, the same distribution

of nodes in the network can be generated repeatedly, and thus one will be able to compare the actions on the same network with various parameters of the simulation and relations settings. The same window enables to determine which routing algorithm will be used for communication ("Type of algorithm" field). At this moment, the simulator implements three groups of algorithms in seven different variants. The groups are:

- shift register,
- energy balanced,
- HEED,

and differ in the idea of operation, criteria for selecting communication paths (consecutive retransmissions) and the principles of relations ordering. The main difference between the first two groups and HEED is that HEED is a standard hierarchical protocol Younis & Fahmy (2004), which does not use the relationship mechanism. The remaining two groups differ in rules that are used to order nodes within relations. For group of 'Shift register' algorithms ordering takes place only once - after the deployment of nodes, during the initialisation of the network. This distinguishes these algorithms from 'Energy balanced' where ordering takes place after every message sent by a node (sort is made by nodes that have sent, received or heard the message exchanged between neighbouring nodes). For both groups, the ordering concerns part of all WSN nodes. This is determined by setting a percentage of nodes in 'Sorted nodes [%]' window. The value determines what portion of nodes will sort their neighbouring nodes according to their proximity to the growing distance from the base station (for groups 'Shift register') or decreasing amount of remaining energy (for the group 'Energy balanced'). Remaining nodes do not sort their neighbouring nodes, which means that the order neighbours in the relation depends on the order in which node learnt of their existence. Relation for each node is represented in simulator as a vector (Register) of neighbouring nodes. Order of nodes within the vector corresponds to the relation ordering between nodes.
Seven routing algorithms available in the current version of the simulator consist of:

- Shift register - this is the algorithm in which each node neighbourhood (represented as a vector) behaves like a cyclic shift register, the shift occur only within a subordination relation, and messages are always sent to the first node from the register. The parameter of this algorithm is the intensity of the other subordination relation that determines the number of neighbours who are subordinated to the node. This parameter determines how many neighbours (counting from the beginning of the vector) are taken into consideration when node is about to send the message.
- Shift register [%] - an algorithm is similar to the previous one but the intensity of the subordination relation is expressed by specifying the percentage of neighbours that are in a subordination relation rather than the number of nodes.
- Shift register [Card$(\Pi) = k$] - in this algorithm the subordination relation includes only neighbouring nodes that are closer to the base station than the current node. Compared with the 'Shift register' algorithm, the difference is that in 'Shift register' subordination relation may consist of nodes that are more distant from the base station than the current node. In the current algorithm, this situation will never take place, although there is no certainty that the best neighbours (the closest to the base station) will be in a subordination relation. For example, this may happen if the registry (that represents the relation) is not sorted.

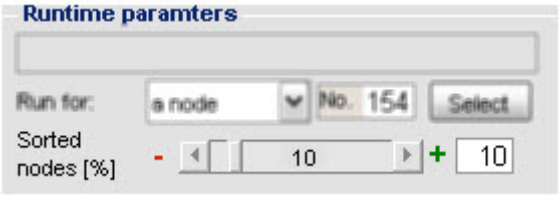


Fig. 4. Parameter Sorted Nodes [%] in the configuration window

- Energy balanced - this is an algorithm in which the subordination relation is composed of a number of neighbours in the left part of the vector (either sorted or not) and the number of nodes in relation is an algorithm parameter. The message is sent to the first node from the vector. After each messages sent, the node sorts this vector according to the amount of residual energy in neighbouring nodes - see description of sorting parameter 'Sorted nodes [%] earlier in this section.
- Energy balanced [%] - this algorithm is similar to the previous one but the difference is that the intensity of the subordination relation is determined by indicating the percentage of the neighbouring nodes that are in the relation.
- Energy balanced [Card$(\Pi) = k$] - similar to 'Shift register [Card$(\Pi) = k$]' the algorithm also restricts the subordination relation to only these neighbours that are closer to the base station than the current node.
- HEED - this is one of the most popular hierarchical algorithm, which defines how to group neighbouring nodes into clusters and transmit messages in the WSN. This algorithm has been implemented in order to compare with our proposal of relational based routing and communication.

4.2 Neighbourhood organisation and network communication efficiency

In the self-organisation phase executed prior to the proper operation of the network, each node collects information about its neighbourhood. Then, using the globally defined metric (expressed in number of retransmissions or the Euclidean distance from the Base Station), each node organises (i.e. sorts according to the residual energy in neighbouring nodes) its neighbours. Number of nodes in the network, which make such an arrangement, is determined by one of the parameters and defines the degree of the neighbourhood ordering. We have evaluated the impact of this parameter on the size of the communication area (that is area covered by nodes that take part in message routing), the number of intermediate nodes and energy efficiency of the algorithms used. The 'Sorted Nodes [%]' parameter specifies the percentage of nodes that sort their neighbouring nodes according to their growing distance from the base station. Other nodes do not sort the neighbourhood, which means that the order of neighbours depends on the order in which the node "learnt" of their existence. In the rest of the chapter, results of simulations and conclusions are presented. All simulations were carried out with fixed values of parameters. These are presented in table 1. Changing the number of organised neighbourhoods has a significant impact on the efficiency of all tested algorithms. And so, when the parameter 'Sorted Nodes [%]' had value 10% for both algorithms 'Shift register [Card$(\Pi) = k$]' and 'Energy balanced [Card$(\Pi) = k$]' then communication area is either very large Fig. 5 or large Fig. 6. It is worth noting that the algorithms from the group of 'Energy balanced', when working with the same parameters, are characterised by a lower

WSN parameters	
Number of sensors	300
WSN area	100×100
Position of the BS	x=1, y=1
Sensor communication range	20
Initial node energy	300
Energy cost of message sent	5
Simulation parameters	
Number of messages to send	300
Communication to the BS	from one selected node
Number of iterations	300
Deployment of nodes	random with fixed seed equal 10

Table 1. WSN and simulation parameters

average number of intermediate nodes required to route messages to the base station. When value of the parameter 'Sorted Nodes [%]' changes from 10% to a maximum value of 100% then there is a diametrical improvement for both families of algorithms. Both paths have a less complicated shape - similar to the line, and thus lead to a base station with a smaller number of hops, which in turn results in improved energy efficiency.

4.3 Principles of retransmitters selection and area of the communication size and energy efficiency

Algorithms from the 'Shift register' group can be divided due to the selection of successors (the following nodes in the routing path of a message that is transmitted to the base station):

- numerical - the value of the parameter 'Reg. capacity' defines the number of neighbouring nodes, from which the successive node is drawn when messages are about to be send,
- percentage - similar to previous but the value of the parameter 'Reg. capacity' defines the percentage of neighbours that will constitute the set from which the successive node will be drawn,
- directional - the value of the parameter 'Reg. capacity' defines the percentage of neighbours that constitute a set $\mathrm{Des}_{\pi}^{\max}(x)$ - set of nodes subordinated to the actual node (x).

4.3.1 Numeric vs. percentage selection

Numerical selection is the least effective method because it allows for the selection of retransmitters without any restrictions; even those nodes can be selected that are outside the desired direction toward the base station. This type of selection of retransmitters does not take into consideration the number of nodes in the neighbourhood that is a property of each node of the network, and may differ significantly throughout the network. Fig. 7 presents how selection of the number of potential retransmitters, appropriate to the number of nodes in the neighbourhood improves the communication efficiency. The 'Reg. capacity'= 10 allows sending the same number of packages, but without reaching the state of energy depletion in some nodes. For example, it follows from Fig. 7 that Card $(\mathrm{Des}_{\pi}^{\max})$=10 is the best value. However, this may not be true for the other nodes. Our tests show that it is the more favourable approach to use percentage selection, where Card $(\mathrm{Des}_{\pi}^{\max})$ corresponds to the number of nodes

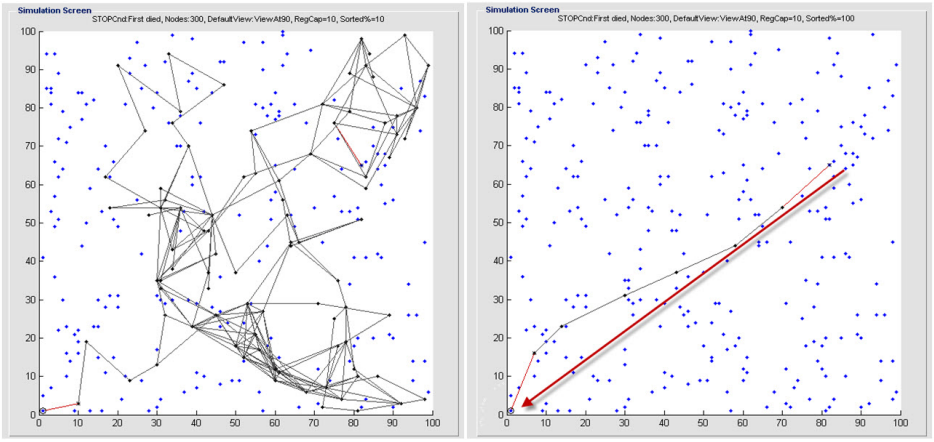


Fig. 5. Algorithm 'Shift register [Card$(\Pi) = k$]' with 'Sorted Nodes [%]' parameter equal 10% (left) and 100% (right) - retransmission path view

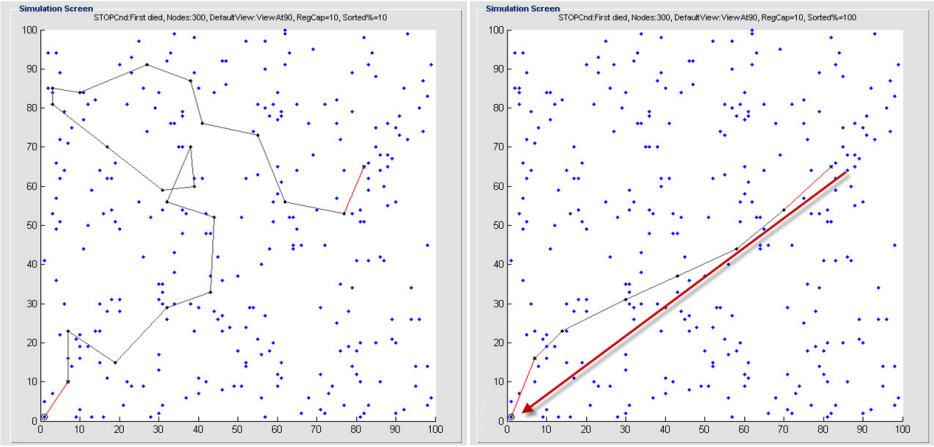


Fig. 6. Algorithm 'Energy balanced [Card$(\Pi) = k$]' with 'Sorted Nodes [%]' parameter equal 10% (left) and 100% (right) - retransmission path view

in the neighbours. Therefore, for each node of the network the number of nodes in $\mathrm{Des}_{\pi}^{\max}$ may differ but when expressed as a percentage, then it is invariant and is adjusted to the local situation of a particular node. This enables us to shape both energy efficiency and the size of the communication area.

4.3.2 Directional and even energy consumption strategy

Directional selection takes into account the neighbours of the transmitter, but only these that are in subordinate relation with it. This enables to shape WSN communication activity, by setting Card $(\mathrm{Des}_{\pi}^{\max})$ as a percentage of neighbouring nodes. Hence, it is not possible, regardless of the value of the parameter 'Reg. capacity', to send a message in a different direction, than towards the base station. When energy costs are considered then this is the best approach,

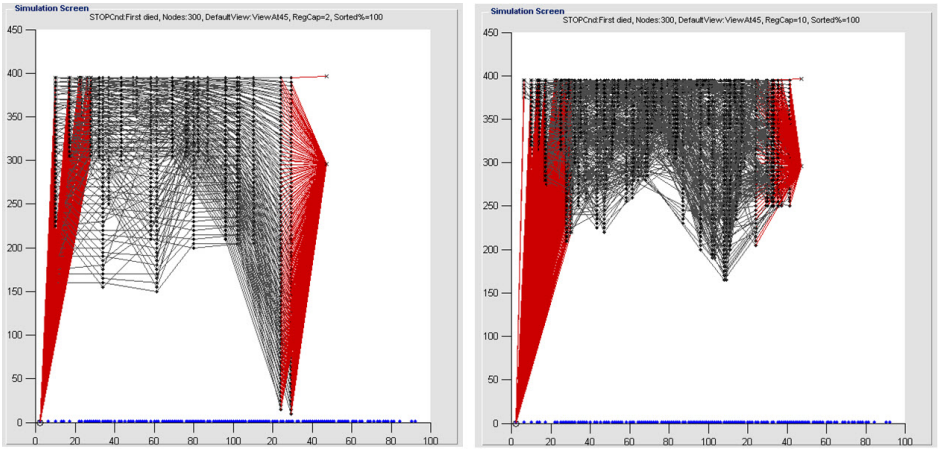


Fig. 7. Energy loses in the network operating according to 'Shift register' algorithm with 'Reg. capacity' parameter set to 2 (left) and 10 (right)

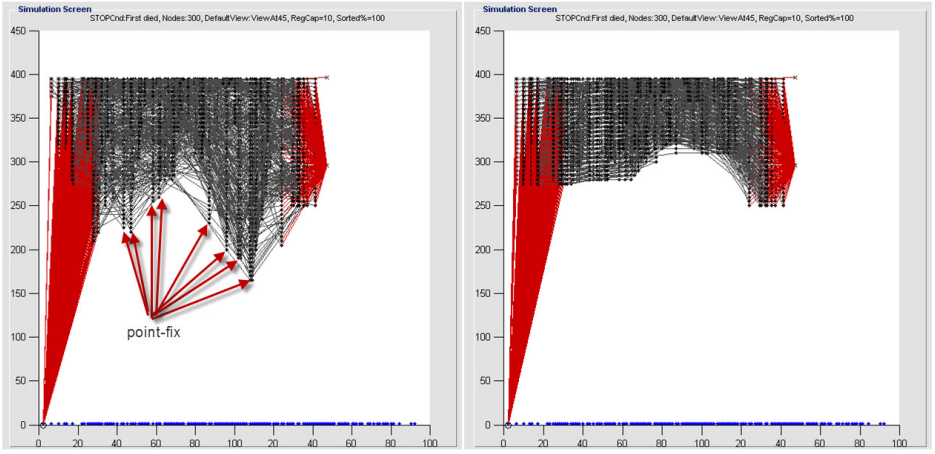


Fig. 8. Energy loses in the network operating according to 'Shift register [Card$(\Pi) = k$]' (left) and 'Energy balanced' (right) with 'Reg. capacity' parameter set to 10

however, as it can be noticed from Fig. 8, in the so-formed communication space, pontifixes (i.e. points that collect messages from a number of nodes) become a problem. As nodes that receive messages from a number of nodes they are overloaded (Fig. 8 left). The solution is in such a situation is to draw on even energy cost strategy that provides uniform, depending only on the network structure, balanced energy consumption (Fig. 8 right).

The main difference of these algorithms when compared to the 'Shift register' group is the focus on uniform energy consumption throughout the whole network. This is a very important aspect of real life systems, where energy depletion in one sensor may affect the operation of the whole network. Algorithms in 'Energy balanced' group strive for a balanced load of nodes that route messages, that in turn increases the average energy consumption required

to transmit a message to the base station. Simplifying the theory we may say that in these algorithms, each node retransmits messages to all its neighbours in turn. During transmission between the nodes neighborhood, only these neighbors are chosen that have the greatest residual energy.
The operation of these algorithms allows for excellent energy saving for nodes that otherwise die quickly. These are the 'pontifixes', in which different communication paths converge. Equivalent energy algorithms cope very well with such a situation. Increased consumption of energy for these nodes can be seen very well on left part of Fig. 8. On the other hand there is almost perfectly balanced energy consumption when all nodes are involved in the transmission (Fig. 8 right).

5. Conclusions

This article presents a relational approach to model the behaviour of wireless sensor networks. The model draws on relations that enable us to represent general, globally defined goals of the network, as well as describe the operation of a single node that has limited information about the network. Three relations (subordination, tolerance and collision) can be used to model communication activities and to control routing paths that are used to transmit messages from sources to the base station. Although, the best setup of relations parameters is not known yet, simulations present that adjusting the intensity of relations enables to control power consumption and extend network lifetime. This improvement results from the fact that every node of the network can adjust its operation according to the current situation in its neighbourhood, rather than strictly following some predefined routing algorithm. The relational approach is also more general than routing algorithms presented in literature so far. Moreover, it encapsulates all previous proposals, so they can be used when needed.

Acknowledgement

This paper has been written as a result of realisation of the project entitled "Detectors and sensors for measuring factors hazardous to environment - modeling and monitoring of threats". The project is financed by the European Union via the European Regional Development Fund and the Polish state budget, within the framework of the Operational Programme Innovative Economy 2007-2013. The contract for refinancing No. POIG.01.03.01-02-002/08-00.

6. References

Braginsky, D. & Estrin, D. (2002). Rumor routing algorthim for sensor networks, *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ACM, New York, NY, USA, pp. 22–31.

Burmester, M., Le, T. V. & Yasinsac, A. (2007). Adaptive gossip protocols: Managing security and redundancy in dense ad hoc networks, *Ad Hoc Netw.* **5**(3): 313–323.

Descartes, R. & Lafleur, L. J. (1960). *Discourse on Method and Meditations*, New York: The Liberal Arts Press.

Dollimore, J., Kindberg, T. & Coulouris, G. (2005). *Distributed Systems: Concepts and Design*, Addison-Wesley.

Jaron, J. (1978). Systemic prolegomena to theoretical cybernetics, *Technical report*, Inst. of Techn. Cybernetics.

Manjeshwar, A. & Agrawal, D. P. (2001). Teen: A routing protocol for enhanced efficiency in wireless sensor networks, *Parallel and Distributed Processing Symposium, International* **3**: 30189a.

Nikodem, J. (2008). Autonomy and cooperation as factors of dependability in wireless sensor network, *Dependability of Computer Systems, International Conference on* pp. 406–413.

Nikodem, J. (2009). Relational approach towards feasibility performance for routing algorithms in wireless sensor network, *Dependability of Computer Systems, International Conference on* pp. 176–183.

Nikodem, J., Klempous, R., Nikodem, M., Woda, M. & Chaczko, Z. (2009). Multihop communication in wireless sensors network based on directed cooperation, *Selected papers on Broadband Communication, Information Technology & Biomedical Application, BroadBandCom '09*, pp. 239–241.

Younis, O. & Fahmy, S. (2004). Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on Mobile Computing* **3**: 366–379.

8

MIPv6 Soft Hand-off for Multi-Sink Wireless Sensor Networks

Ricardo Silva, Jorge Sa Silva and Fernando Boavida
University of Coimbra
Portugal

1. Introduction

Although Wireless Sensor Networks (WSNs) are one of the most promising technologies of the 21st century - with potential applications in virtually all areas of activity, ranging from the personal area to the global environment - a considerable number of challenges has still to be addressed in order to make WSNs a day-to-day reality. First of all, reachability issues (including IP connectivity, addressing and routing) must be solved. Then, other problems such as self-configuration, quality of service, and security must also be tackled. A crucial aspect, however, is mobility. Many applications require sensor mobility, and either network mobility, to be effective. Some examples include the use of WSNs for vehicle monitoring and control, or health parameters monitoring of ambulatory patients. Without efficient mobility mechanisms, the application areas of WSNs will be highly restricted.

In terms of WSN reachability, there is clear movement towards the adoption of IPv6. The use of IP in sensor nodes has considerable benefits in terms of connectivity, and IPv6 has several advantages when compared to IPv4, the most prominent being the much larger address space. There are, nonetheless, other important advantages of IPv6, such as native support for mobility, anycast addressing, security and self-configuration.

Recently, the IETF created the 6LowPAN group Mulligan (2008) to study the integration of IPv6 in simple IEEE 802.15.4 wireless devices. 6LowPAN proposes a middleware layer to integrate IPv6 in WSNs. Concerning packet headers, although the IPv6 header is simpler when compared to the IPv4 header, it is larger because of the use of 128-bit addresses, as opposed to the 32-bit addresses in IPv4. To circumvent this, 6LowPAN proposes the use of compressed headers.

There are already some implementations of 6LowPAN modules for the TinyOS and Contiki operating systems. However, mobility is not yet supported in these IPv6-over-WSNs environments.

Although mobility of WSNs has been addressed in the recent past, most of the existing work assumes mobility of the whole WSN (i.e., of sink nodes) Dantu (2005) Labrindis (2005) Raviraj (2005), leaving out the issue of sensor node mobility. There are, nevertheless, some models Ekici (2006) Heidemann (2002) that propose the use of MAC-layer protocols to support mobile sensor nodes registration. However, to the best of our knowledge, they do not address the integration of WSNs in the IP world.

In this paper we propose a framework for an effective support of mobility in WSNs. The innovative aspects of the framework consist of the use of mobile IPv6 (MIPv6) in wireless sensor

networks, the use of Neighbor Discovery for discovery of sink nodes and subsequent node registration and, last but not least, the use of a soft hand-off approach which prevents connectivity breaks while the sensor nodes are moving. Section 2 presents the proposed framework, including the sink node discovery and soft hand-off mechanisms. The framework has been evaluated through implementation, and the obtained results are presented in section 3. Section 4 provides the conclusions and guidelines for further research.

2. Proposed Framework

The proposed framework has the objective of efficiently dealing with the main requirements of wireless sensor networks, with the aim of overcoming some of the most important obstacles that prevent real world WSN deployments. The distinguishing features of the framework are the following:

- Multi-sink approach, in order to simplify routing; this precludes the need for complex and unrealistic multi-hop routing protocols and drastically reduces node energy constraints;
- Use of Mobile IPv6, thus leading to the availability of generalised IP connectivity and of native mobility;
- Soft hand-off approach, thus maximising the connectivity of mobile sensor nodes;
- Link quality prediction, allowing sensor nodes to decide if hand-off to other sink node is beneficial and/or feasible.

In the following sub-sections, these features and their underlying mechanisms will be addressed and explained in detail.

2.1 Sink Discovery and Node Registration

Two basic types of topologies can be used in WSNs: Single-sink multi-hop topology, also known as mesh topology, and multi-sink single-hop topology, also known as star topology.
In mesh topologies, all sensor nodes perform not only sensing tasks but also routing tasks, forwarding data towards the sink node through neighbouring nodes. At first glance, multi-hop communication appears to be more energy-efficient when compared to long-range single-hop communication, due to the fact that mesh topologies lead to shorter distances between transmitter and receiver. However, the apparent energy optimization of mesh topologies comes with too high a price, which is at the basis of the failure of real world WSN deployment: extreme complexity at various levels. In fact, mesh topologies require aggregation methods, signaling messages, increased memory, broadcast procedures, substantial overhead, complex routing protocols and/or large routing tables. This complexity is more critical in mobile environments. The dynamics of these environments causes changes in the network topology and, therefore, in routing, which leads to additional complexity and overhead.
Naturally, a mesh topology can be transformed into a star topology if several sink nodes are deployed, each covering a relatively small cell comprising several sensor nodes. In this case, energy-efficiency of sensor nodes can still be achieved Ð distances to a sink node can be kept small Ð and, in fact, sensor nodes can be simpler, as they do not need to forward packets or to perform complex routing tasks. The price to pay is the deployment of more sink nodes, but clearly in many cases it is easier to deploy more sink nodes than to use forbiddingly complex routing protocols.
However challenging and interesting might be the routing problem in mesh-based WSNs, the hard fact is that most (if not all) real applications of WSNs use a star topology. The reason

is that with a star topology, the routing complexity disappears, and simple routing solutions can be adopted. This is, in fact, the rationale for using a multi-sink single-hop approach in the proposed framework, depicted in the scenario presented in Figure 1.

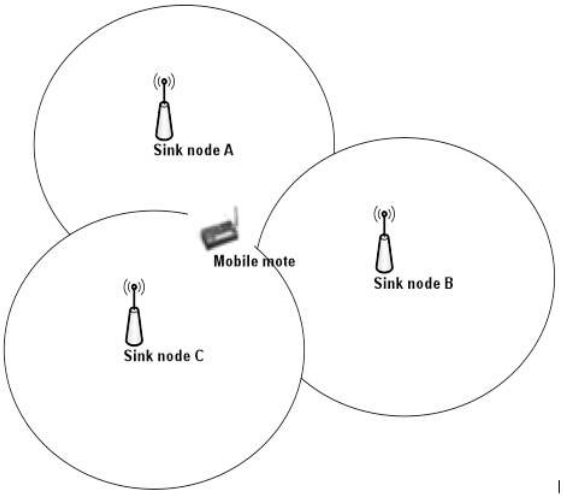


Fig. 1. Multi-Sink WSN mobility scenario

The use of multiple sink nodes must be accompanied by sink node discovery mechanisms which allow mobile sensor nodes to dynamically detect them and perform the necessary registration. The mechanism developed by the authors Ð based on preliminary work presented in Silva (2008) Ð is initiated by mobile sensor nodes, in order to avoid energy-expensive broadcasts from sink nodes. The underlying protocol is clearly an extension of the Neighbor Discovery protocol, and was implemented with the help of ICMPv6 extension messages. After choosing a sink node, mobile sensor nodes perform a registration operation, depicted in Figure 2a).
The registration operation consists of the following steps (see Fig. 2a):

1. Upon deployment, the node broadcasts a Router Solicitation (RS) message.
2. Sink nodes in range send back Router Advertisement (RA) messages.
3. The node collects the received RA messages and chooses the best sink node, based on the Received Signal Strength Indicator (RSSI) of each of the received message.
4. The node sends an acceptance message (ACCEPT) to the selected sink node.
5. The selected sink node receives the ACCEPT and responds with the TTL value to be used by the sensor node.
6. The node receives the TTL and self-configures its global address, based on the address prefix of the sink node.
7. The node sends an Acknowledgment message (ACK) to the sink node.
8. The sink node inserts the new sensor node in its Binding Table.

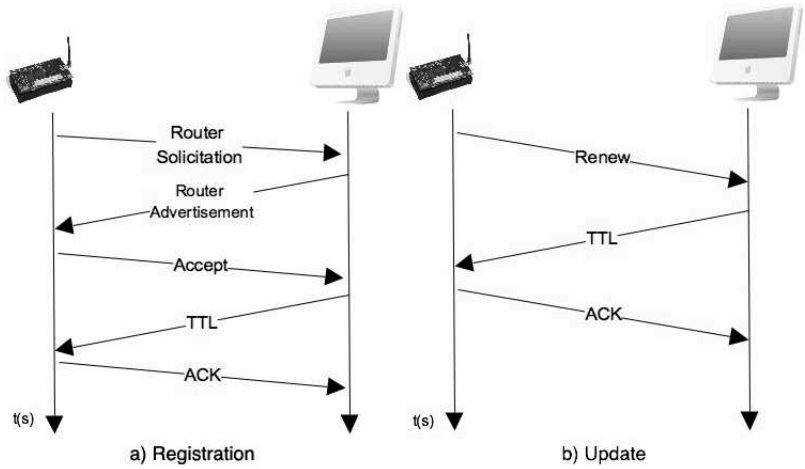


Fig. 2. Sink node discovery, registration and update

In the registration procedure the node uses the IPv6 stateless configuration mechanism to build its own address, using as prefix the one of the chosen network, and as suffix its Interface Identifier.
After registration, each node maintains a Time-To-Live (TTL) value. When this value becomes zero, the mobile node evaluates the signal strength and the Link Quality Indicator of all the sink nodes in the area to choose the best one. If the elected sink node is the one already in use by the mobile node, it is only necessary to start the update procedure (Figure 2b). If a new sink node is chosen, the registration procedure must be performed. The update procedure is simpler than the registration procedure, as the mobile node requests, using a unicast message, the revalidation of the registration.

2.2 Soft Hand-Off

In order to support node mobility, sink nodes maintain a binding table (see Table 1) with all their registered nodes, TTLs, supported services and nodesÕ Care-of-Address (CoA). Table 1 presents the various fields of the binding table.

Home Address	TTL	List of Services	Care-of-Address
Obtained during the node discovery procedure			<Null> or <New prefix + > Old sufix

Table 1. Binding Table

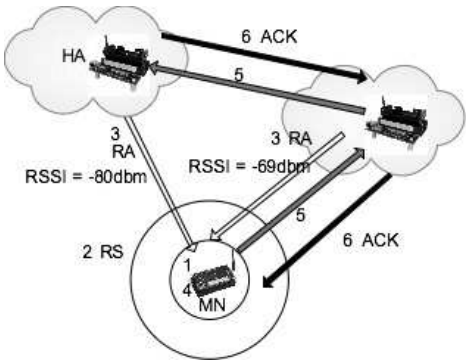
The first three fields of this table are filled in during the initial registration procedure. The CoA is initialised as null, being updated each time the node moves to a new foreign sub-network. The node, in turn, internally registers its Home Agent (Sink Node) Address, which remains the same while the current registration is valid.

If a node detects that the connection to its current sink node is in the critical zone Silva (2009), it initiates the sink node discovery/registration procedure described in section 2.1, by sending an RS message. Note that the new sink node discovery is performed before the connection to the current sink node is broken, in order to achieve a soft hand-off. This soft hand-off procedure is illustrated in Figure 3, below, and consists of the following steps:

1. The mobile sensor node (MN) detects a bad connection to the current sink node.
2. The MN broadcasts a Router Solicitation message (RS).
3. The MN receives (in the example) two Router Advertisements (RA).
4. The MN selects the sink node with the best received signal strength and re-configures its global address, changing the prefix to the one of the new sink node.
5. The MN sends a Binding Update message notifying the HA of its new COA, through the new link, guaranteeing that the message arrives there.
6. Upon reception of the Binding Update, the HA sends an Acknowledgement message to the MN and updates the COA in its Binding Table.

The choice of a new sink node should take into account not only the received RSSI, but also the nodeÕs velocity, the existing noise level and the mean time taken by hand-off operations. If a mobile node moves away from its current sink node with constant velocity $V(m/s)$, in an environment with noise level $N(dBm/m)$, and takes M seconds to perform the soft handoff, the link quality to its current sink node at the end of the hand-off can be estimated by:

$$Q_M = RSSI - (M \times V \times N) \tag{1}$$



Equation (1) can be used to predict the link quality at the end of the hand-off process and, thus, it can assist the decision on if and when to choose another sink node. For example, considering an RSSI of $-60dBm$, a 2 seconds mean hand-off time, a velocity of $2m/s$ and a noise level of $5dBm/m$, at the end of the handoff process the link quality would be:

$$Q_M = -60 - (2 \times 2 \times 5) \Leftrightarrow$$
$$Q_M = -80dBm$$

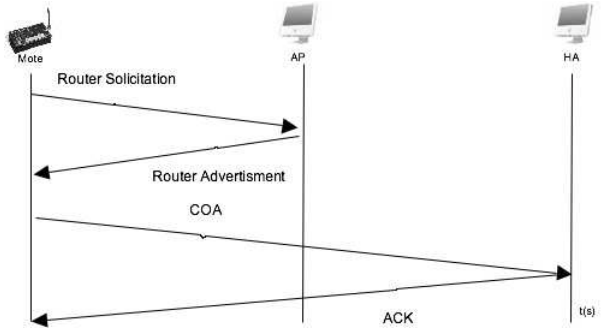


Fig. 3. Soft Handoff

The same formula can be applied not only to predict the link quality at the end of the hand-off, but also to predict the link quality within the home network, after M units of time. Such deductions are extremely useful to optimize the behaviour of sensor nodes in dynamic environments. Based on mobility and environment characteristics, nodes will be able to self adapt to a variety of situations.
If communication between a Correspondent Node (CN) and the Mobile Sensor Node (MN) is taking place during the hand-off, a transparent CoA update procedure is performed by the MN during the soft hand-off, as described above, and this leads to no message losses. This is complemented by a Binding Update sent by the Home Agent to the CN, in order to optimize subsequent communication instances. Figure 4 illustrates the process, which is comprises the following steps:

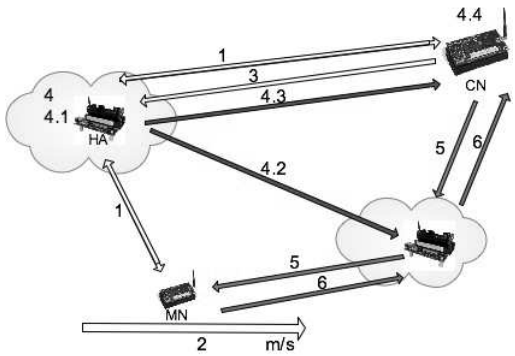


Fig. 4. Communication path update

1. The MN is communicating with CN.
2. The MN moves to a new attachment point.
3. The CN sends a message towards the HA:
4. The HA checks the CoA of the MN in the binding table.

4.1. The HA uses the CoA as the new destination address.

4.2. The HA tunnels the packet to the CoA.

4.3. The HA notifies the CN about the new CoA.

4.4. The CN Updates an internal Binding Cache.

5. The next time, the CN sends messages directly to the CoA.
6. The MN uses always its current attachment point to relay its messages.

3. Evaluation

To test and evaluate the performance of the proposed framework we implemented it in a real platform. We used MicaZ motes programmed with a 6lowPAN implementation Harvan (2007) modified according to our architecture. The sink nodes were Mib520 attached to ubuntu-based machines and running a special daemon, that we developed in C to support our framework. We used ICMPv6 message types 150 to 160 in order to implement the proposed framework supporting protocol. Additionally, we re-used the RA and RS messages from the Neighbor Discovery protocol.
The main purpose of the carried out test was the determination of the average duration of the soft handoff procedure. To measure this, we configured a network with two sink nodes and a mobile sensor node. Each sink node had two interfaces, one to the WSN and another to a local IPv6 network. Figure 5 illustrated the test-bed scenario. Wireshark was installed and used in order to monitor all packets and to control time, rates and delays. The test suites comprised three steps:

1. The initial registration of the MN in the HA, using the proposed procedure;
2. The movement of the MN;
3. The soft hand-off process.

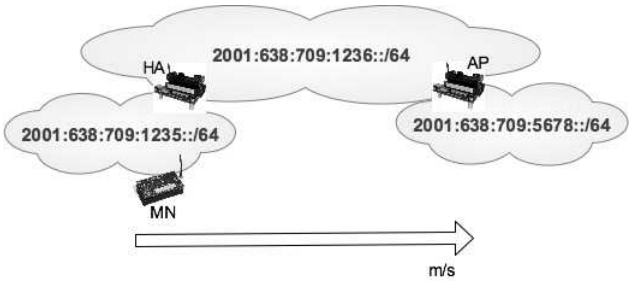


Fig. 5. Test-bed scenario

We measured the time elapsed since the node detects a quality degradation of the link connection to the HA, until it finishes the soft handoff process to the new attachment point. We performed 300 hand-off operations and corresponding measurements. The results are presented in table 2.

Minimum	Maximum	Mean	Std. deviation
2.081761	2.124737	2.10470933	.009944052

Table 2. Total soft-hand-off time (seconds), including the initial detection of signal quality degradation

The determined mean soft hand-off time can be used in conjunction with Equation (1) to estimate the quality of the sink connection under a variety of situations. For instance, as determined in Silva (2009) the minimum quality level guaranteeing connectivity (also known as rupture point) is $-88dBm$. Below this level, a hard hand-off must take place, that is, there will be and interruption of the connectivity. Using this value, the mean hand-off time determined in the tests and equation (1), it is possible to determine the maximum value for the product of velocity and noise (which we will represent by ΔC). Hence:

$$\begin{aligned} -88 = -60 - (2.10470933 \times \Delta c) \Leftrightarrow \\ -28 = -2.10470933 \times \Delta c \Leftrightarrow \\ \Delta c = \sim 13.305 dBm/s \end{aligned}$$

In addition to obtaining the mean value for soft hand-off operations, the tests allowed us to verify the feasibility of the proposed framework, namely the use of the multi-sink approach, mobile IPv6, soft hand-off and link quality prediction.

4. Conclusion

Although considerable work has been and is being done in the area of wireless sensor networks, relatively few deployments exist. This is mainly due to the complexity inherent to multi-hop routing and to the lack of efficient mobility solutions.
In an attempt to circumvent these problems, we have proposed a framework that eliminates the need for multi-hop communication, uses mobile IPv6 as the basis for node mobility, explores the use of Neighbor Discovery for the discovery of sink nodes and subsequent node registration and, last but not least, allows soft hand-off. The proposed approach has been implemented in a laboratorial environment in order to assess its feasibility and to identify potential problems. In addition to proving the feasibility of the proposal, the tests that were carried out also allowed us to obtain mean hand-off values, which can be used by sensor nodes to estimate the link quality while moving from one sink node to another.
Future work will address three important aspects: further exploration and refinement of the soft hand-off technique; study of the impact of and solutions for movement to successive foreign networks; and study and implementation of route optimization techniques.

5. References

G. Mulligan et al. 'The 6lowpan website'. Available: www.ietf.org/html.charters/6lowpan-charter.html.

K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, "Robomote: enabling mobility in sensor networks," April 2005, pp. 404Ð409.

A. Labrinidis and A. Stefanidis, "Panel on mobility in sensor networks," in MDM Õ05: Proceedings of the 6th international conference on Mobile data management. New York, NY, USA: ACM, 2005, pp. 333-334.

P. Raviraj, H. Sharif, M. Hempel, H. H. Ali, and J. Youn, "A new mac approach for mobile wireless sensor networks," in Proceedings of the 14th IST Mobile and Wireless Communication Summit, 2005.

E. Ekici, Y. Gu, and D. Bozdag, "Mobility-based communication in wireless sensor networks," Communications Magazine, IEEE, vol. 44, no. 7, pp. 56-62, July 2006.

W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," vol. 3, 2002, pp. 1567-1576 vol.3.

R. Silva, J. S. Silva, C. Geyer, L. da Silva, and F. Boavida, "Wireless sensor networks - service discovery and mobility," in 7th International Information and Telecommunication Technologies Symposium, Foz do Iguau, BRAZIL, 2008.

R. Silva, J. S. Silva, M. Simek, and F. Boavida, "A new approach for multi-sink environments in wsns," 11th IFIP/IEEE International Symposium on Integrated Network Management, Jun. 2009.

M. Harvan, "Connecting wireless sensor networks to the internet a 6lowpan implementation for tinyos 2.0," presented at the Jacobs University Bremen, Germany, 2007.

9

Cooperative Clustering Algorithms for Wireless Sensor Networks

Hui Jing and Hitoshi Aida
The University of Tokyo
Japan

1. Introduction

1.1 Wireless sensor networks

Wireless sensor networks have been made viable by the convergence of micro-electro-mechanical systems technology, wireless communications and digital electronics (Akyildiz et al., 2002). They are expected to consist of a large number of inexpensive sensor nodes, each having sensing, data processing and communicating components with limited computational and communication power. To provide various measurements such as light, temperature, pressure and activity, these low-cost, low-power, multifunctional sensor nodes have been widely deployed in a vast variety of environments for commercial, civil, and military applications such as surveillance, vehicle tracking, climate, etc.. However, a single sensor's view of the environment is restricted both in range and in accuracy, due to it only covers a limited physical area and may produce noisy data by the quality of the hardware. Accordingly, aggregation of the individual surveillance allows users to accurately and reliably monitor an environment.

Once sensor nodes are deployed throughout an area, they collect data from the environment and automatically establish dedicated networks to transmit their data to a base station. The nodes collaborate to gather data and extend the operating lifetime of the entire system. Wireless sensor networks offer a longevity, robustness, and ease of deployment that is ideal for environments where maintenance or battery replacement may be inconvenient or impossible (Hac, 2003). In recent years, with the rapid development of embedded systems including energy efficient devices, hardware/software co-design and networking support, sensor nodes have been smaller in size and more efficient in data processing and transmission. However, they are still limited in power, memory and computational capacities. As a result, the key challenge is to maximize the lifetime of sensor nodes due to the fact that it is not feasible to replace the batteries of thousands of nodes.

1.2 Clustering algorithms for wireless sensor networks

As one of the most widely investigated topology control mechanisms for wireless sensor networks, the clustering algorithm provides network scalability and energy efficient communications by reducing transmission overhead and enhancing transmission reliability. It can localize the route set up within the cluster and thus reduce the size of the routing table stored at the individual sensor node. Clustering can also conserve communication bandwidth since it limits the scope of inter-cluster interactions to cluster heads and avoids redundant exchange

of messages among sensor nodes (Younis et al., 2003). Moreover, clustering can stabilize the network topology at the level of sensor nodes and thus cuts on topology maintenance overhead (Abbasi & Younis, 2007).
The clustering protocols have been extensively proposed for achieving scalability through hierarchical approaches specifically for wireless sensor networks. In our research, we divide these clustering algorithms into self-configuring cluster formation and centralized cluster formation. In centralized cluster formation, the base station elects cluster heads each round to afford guarantee about the placement and number of cluster heads by a centralized clustering scenario. Hence, these protocols often need sensor nodes to be equipped with high-sensitivity global positioning system receivers for gathering position information of sensor nodes. In self-configuring cluster formation, each sensor node makes autonomous decisions itself using a distributed algorithm. The advantages of this approach are that no long-distance communication to the base station is required and distributed cluster formation can be done even without the exact location information of the sensor nodes in the network. In addition, no global communication is needed to set up the clusters and nothing is assumed about the current state of any other sensor node during cluster formation (Heinzelman, 2000).
In this chapter, we mainly concentrate on self-configuring cluster formation. In a clustering scheme, the network is partitioned into several clusters. Every cluster would have a leader, referred to as the cluster head. A cluster head is elected by the sensor nodes in a cluster for self-configuring cluster formation. A cluster head may be just one of the nodes or a node that is richer in resources. The cluster membership should be fixed or variable. After election, each cluster head broadcasts an advertisement message using carrier-sense multiple access for media access control protocol. Other nodes determine their cluster by the received signal strength of the advertisement messages, which is used as a measure of the required transmit power. Each non cluster head node determines which cluster it belongs to by choosing the cluster that requires the minimum communication energy. In a cluster, a cluster head gathers sensing data from all sensor nodes in the same cluster through a preset time division multiple access schedule and produces a condensed summary which is forwarded to the base station in each frame. A sensor node is associated with, at most, one cluster head and all communications are relayed through the cluster head.
The rest of this chapter is organized as follows. First of all, we introduce clustering algorithms for wireless sensor networks in Section 2. Then in Section 3, a cooperative game model for clustering in wireless sensor networks is presented for the nature of strategic interaction. Afterwards, we develop conditions to form cluster head coalitions and describe the cooperative game theoretic clustering algorithm in Section 4. Furthermore, as the results of simulation, we quantitatively analyze network lifetime, data transmission capacity and energy efficiency in Section 5. Finally, we draw conclusions in Section 6.

2. Previous Works

During recent years, a number of algorithms on self-configuring clustering had been presented for achieving energy efficiency. Low-Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman, 2000; Heinzelman et al., 2002) is an application-specific protocol architecture that forms clusters by a distributed algorithm. Cluster heads are burdened with a long-distance transmission to base station. Clustering explicitly encourages data aggregation to reduce the transmission burden in the network. This way, depending on the network configuration an increase of network lifetime can be accomplished (Hac, 2003). Afterwards, the low energy adaptive clustering hierarchy with deterministic cluster head selection (DCHS)

(Handy et al., 2002) extends LEACH's stochastic cluster head selection algorithm by a deterministic component and solves the problem of which the network is stuck after a certain number of rounds by a low cluster head selection threshold. Hybrid energy-efficient distributed clustering (HEED) (Younis & Fahmy, 2004) is a distributed scheme in which cluster heads are periodically selected according to a hybrid of the sensor node residual energy and communication cost. Recently, energy-efficient distance based clustering routing scheme (EEDBC) (Han et al., 2007) considers a distance from the base station to a cluster head and the residual energy as the criterion of the cluster head election for balance energy consumption among cluster heads. Therefore, this approach provides fully distributed manner and energy efficiency. In this section, we explain clustering algorithms which are widely investigated in the past few years.

2.1 Low-energy adaptive clustering hierarchy (LEACH)

LEACH is a protocol architecture for sensor networks that combines the ideas of energy-efficient cluster-based routing and media access together with application-specific data aggregation to achieve good performance in terms of system lifetime, latency and application-perceived quality (Heinzelman et al., 2002).

The operation of LEACH is divided into rounds. Each sensor node elects itself to be a cluster head at the beginning of round $r+1$ (which starts at time t) with probability $P_i(t)$. $P_i(t)$ is chosen such that the expected number of cluster heads for this round is k. Thus, if there are N sensor nodes in the network, the expected number of cluster heads is:

$$E[number\ of\ cluster\ heads] = \sum_{i=1}^{N} P_i(t) = k. \tag{1}$$

Each sensor nodes to be a cluster head once in N/k rounds on average. $C_i(t)$ is denoted as the indicator function determining whether or not sensor node i has been a cluster head in the most recent ($r \bmod \frac{N}{k}$) rounds, then each sensor node should choose to become a cluster head at round r with probability:

$$P_i(t) = \begin{cases} \dfrac{k}{N - k(r \bmod \frac{N}{k})} & : C_i(t) = 1\,, \\ 0 & : C_i(t) = 0. \end{cases} \tag{2}$$

Therefore, only sensor nodes that have not already been cluster heads recently, and which presumably have more energy available than other sensor nodes that have recently performed this energy intensive function, may become cluster heads at round $r+1$.

As shown in the flowchart of Fig. 1, LEACH processes as follows: once the sensor nodes have elected themselves to be cluster heads using the probabilities in (2), the cluster head should let all the other nodes in the network know that they have chosen this role for the current round. Therefore, each cluster head broadcasts an advertisement message. This message is a short message containing the node's ID and a header that distinguishes this message as an announcement message. Other nodes determine their clusters for this round by choosing the cluster heads that require the minimum communication energy, based on the received signal strength of the advertisement from each cluster head. Assuming symmetric propagation channels for pure signal strength, the cluster head advertisement heard with the largest signal strength is the cluster head that requires the minimum amount of transmit energy to communicate with. Note that typically this will be the cluster head closest to the sensor, unless

there is an obstacle impeding communication. In the case of ties, a random cluster head is chosen. After each sensor node has decided to which cluster it belongs, it informs the cluster head that it will be a member of the cluster. Each node transmits a join message back to the chosen cluster head. This message is again a short message, consisting of the node's ID and the cluster head's ID. The cluster heads in LEACH act as local control centers to coordinate the data transmissions in their cluster. The cluster head sets up a time division multiple access schedule and transmits this schedule to the sensor nodes in the cluster. This ensures that there are no collisions among data messages and also allows the radio components of each non cluster head to be turned off at all times except during their transmit time, thus reducing the energy consumed by the individual sensors. After the time division multiple access schedule is known by all sensor nodes in the cluster, the data transmission can begin. Fig. 2 shows an example of clusters formed in one round of LEACH. In this figure, each cluster has taken on a different color. In the cluster, the cluster head is denoted by a triangle. The position of base station is $(50, 175)$.

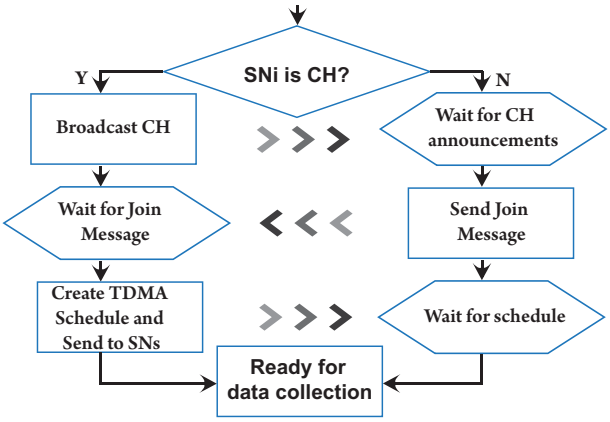


Fig. 1. Flowchart of LEACH procedure. (SN: sensor node; CH: cluster head)

2.2 Low energy adaptive clustering hierarchy with deterministic cluster head selection (DCHS)

DCHS is an energy-efficient clustering hierarchy protocol which is a modified version of the LEACH. Due to the inclusion of the residual energy level available in each sensor node, the approach increases the lifetime of a LEACH network. It can be achieved by (3), relative to the sensor node's residual energy. And this mechanism is expanded by a factor that increases the probability for any sensor node that has not been cluster head for the last k/N rounds.

$$P_i(t) = \frac{k}{N - k(r \bmod \frac{N}{k})}[\frac{E_{i\_res}}{E_{i\_ini}} + (r_s \mathrm{div} \frac{k}{N})(1 - \frac{E_{i\_res}}{E_{i\_ini}})]. \quad (3)$$

with r_s as the number of consecutive rounds in which a sensor node has not been a cluster head. E_{i_res} and E_{i_ini} denote the residual and initial energy for sensor node i, respectively. Additionally, r_s is reset to 0 when a sensor node becomes a cluster head. For the deterministic selection of cluster heads only local and no global information is necessary. The nodes

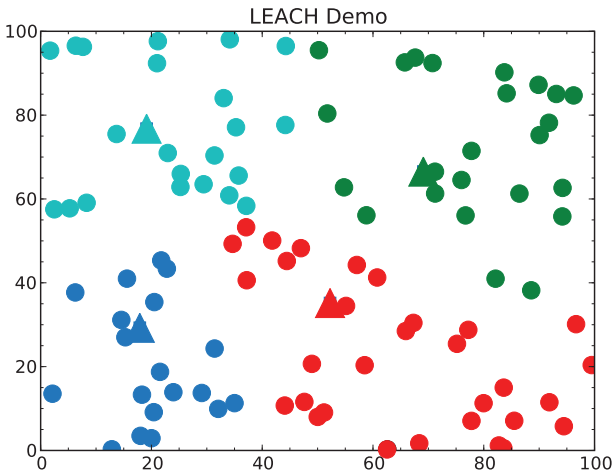


Fig. 2. The example: Cluster formation of LEACH in one round

determine themselves whether they become cluster heads. A transmission between the base station and a cluster head is not necessary.

2.3 Hybrid energy-efficient distributed clustering (HEED)

HEED considers a hybrid of energy and communication cost when selecting cluster heads. Unlike LEACH, it does not select cluster heads randomly. Only sensor nodes that have a high residual energy can become cluster heads (Abbasi & Younis, 2007). HEED has three main characteristics:

- To achieve well distribution of cluster heads in the network, the probability that two sensor nodes within each other's transmission range becoming cluster heads is small.
- Energy consumption is assumed to be multiform for all the sensor nodes.
- Within a given node's transmission range, the probability of cluster head selection can be adjusted to ensure inter cluster head connectivity.

In HEED, each sensor node is mapped to exactly one cluster and can directly communicate with its cluster head. The algorithm is divided into three phases:

1. Initialization phase: The algorithm first sets an initial percentage of cluster heads among all nodes. This percentage value, C_p, is used to limit the initial cluster head announcements to the other sensor nodes. Each sensor node sets its probability of becoming a cluster head, CH_p, as follows: $CH_p = C_p \times E_{res}/E_{ini}$, where E_{res} is the current energy in the node, and E_{ini} is the initial energy, which corresponds to a fully charged battery. CH_p is not allowed to fall below a certain threshold p_{min}, which is selected to be inversely proportional to E_{ini}.
2. Repetition phase: During this phase, every sensor node goes through several iterations until it finds the cluster head that it can transmit to with the least transmission power (cost). If it hears from no cluster head, the sensor node elects itself to be a cluster head and sends an announcement message to its neighbors informing them about the change of status. Finally, each sensor node doubles its CH_p value and goes to the next iteration

of this phase. It stops executing this phase when its CH_p reaches 1. Therefore, there are 2 types of cluster head status that a sensor node could announce to its neighbors:

- Tentative status: The sensor node becomes a tentative cluster head if its CH_p is less than 1. It can change its status to a regular sensor node at a later iteration if it finds a lower cost cluster head.
- Final status: The node permanently becomes a cluster head if its CH_p has reached 1.

3. Finalization phase: During this phase, each sensor node makes a final decision on its status. It either picks the least cost cluster head or pronounces itself as cluster head.

2.4 Energy-efficient distance based clustering (EEDBC)

EEDBC considers the uneven energy consumption of cluster heads which is resulted from uneven transmission cost between inter-cluster and intra-cluster communication due to the difference of distance to the base station. In other words, the basic ideal is that the closer to the base station, the larger cluster area. Therefore, each sensor node has the probability of becoming a cluster head which is determined by the distance to the base station and its residual energy.

$$P_i(t) = c \times \frac{d(S_i, BS) - d_{min}}{d_{max} - d_{min}} \times \frac{E_{i\_res}}{E_{i\_ini}}. \tag{4}$$

where c is a constant coefficient between 0 and 1, $d(S_i, BS)$ represents the distance between sensor node i and the base station, d_{max} represents the distance of the farthest sensor node from the base station and d_{min} represents the distance of the closest sensor node. E_{i_res} and E_{i_ini} denote the residual and initial energy for sensor node i, respectively. Fig. 3 shows an example of clusters formed in one round of EEDBC. In this figure, the denotation is same as the example of LEACH. We can find that the farther sensor nodes have higher probability to become cluster heads.

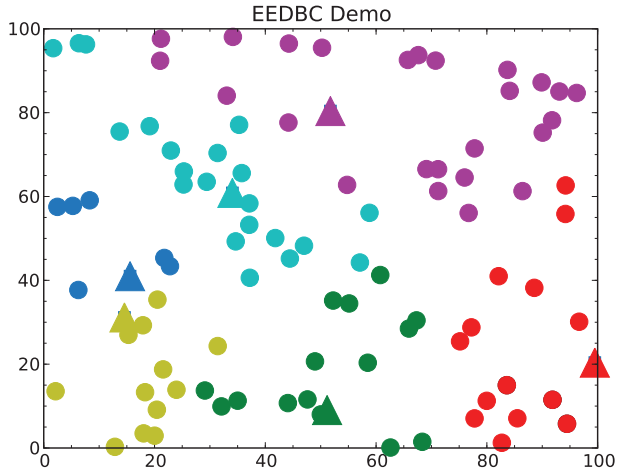


Fig. 3. The example: Cluster formation of EEDBC in one round

However, in the previous research, most of the game formulations for wireless sensor networks are non-cooperative games (Felegyhazi et al., 2006; Zheng et al., 2004), where sensor nodes act selfishly, to minimize their individual utility in a distributed decision-making environment (Machado & Tekinaya, 2008). Even if residual energy is utilized in the clustering algorithms, the behavior of sensor node is individual. Consequently, the network partition is expedited, and uneven residual energy is distributed across sensor nodes. In order to obtain global optimization, a cooperative game theoretic model is provided for balancing energy consumption of sensor nodes and increasing network lifetime and stability in this paper. Then, through the solution of the model, feasible cost allocations, we propose and analyze the cooperative clustering approach.

3. Cooperative Game Theoretic Model of Clustering Algorithms for Wireless Sensor Networks

3.1 Game and solution

Game theory is a mathematical basis for capturing behavior in interactive decision situation. It provides a framework and analytical approach for predicting the results of complex and dynamic interactions between rational agents who try to maximize personal payoff (or minimize private cost) according to strategies of other agents. The theory is generally divided into the non-cooperative game theory and the cooperative game theory. In non-cooperative games, the agents have distinct interests that interact by predefined mechanisms and deviate alone from a proposed solution, if it is in their interest, and do not themselves coordinate their moves in groups. In other words, for individually rational behaviors, they cannot reach an agreement or negotiate for cooperation. Contrarily, a cooperative game allows agents to communicate for allocating resources before making decisions by an unspecified mechanism. It is concerned with coalitions which are composed of group of agents for coordinating actions and feasible allocations. Cooperative game theory is concerned with situations when groups of agents coordinate their actions. Consequently, Cooperative games focus how to assign the total benefits (or cost) among coalitions, taking into account individual and group incentives, as well as various fairness properties (Nisan et al., 2007).
In this chapter, we mainly consider a cost sharing game which is a cooperative game concentrating on cost but not benefits. It is composed of a set $\mathcal{A}$ of n agents and a cost function c. Let $\mathbb{R}^+$ denote a set of nonnegative real numbers and $2^{\mathcal{A}}$ denote the set of all subsets of $\mathcal{A}$. We define the notion of a cost sharing game as follows:

Definition 3.1. (Cost Sharing Game) *A cost sharing game consists of a finite set $\mathcal{A}$ of n agents and a cost function c: $2^{\mathcal{A}} \longrightarrow \mathbb{R}^+$ to denote the nonnegative cost from the set of coalition.*

As a widely applicable concept, the Shapley value is a solution that assigns a single cost allocation to cost sharing games. We choose this solution to a cooperative game since the computational complexity is small and the Shapley value provides relatively anonymous solution by a random ordering of the agents. It had been proved that the Shapley value is the unique value on the set of games satisfying anonymity, dummy and additivity. Let $S \subseteq \mathcal{A}\backslash\{i\}$ denote all coalitions S of $\mathcal{A}$ not containing agent i. For any agent $i \in \mathcal{A}$ and any set $S \subseteq \mathcal{A}\backslash\{i\}$, the probability that the set of agents that come before i in a random ordering is precisely S is $s!(n-1-s)!/n!$, where $s = |S|$ is cardinality of S. Then the Shapley value ϕ on the cost

function c is represented by the following equation (5): For each agent i,

$$\phi_i(c) = \sum_{S \subseteq \mathcal{A} \setminus \{i\}} \frac{s!(n-1-s)!}{n!} (c(S \cup \{i\}) - c(S)) \tag{5}$$

where ϕ indicates the cost allocation in the cost sharing game $(\mathcal{A}, c)$.
Shapley value has three properties defined as follows:

- Anonymity: Even the agents change names, their cost shares do not change. Therefore, ϕ satisfies anonymity.
- Dummy: An agent who does not add to the cost should not be charged anything. Formally, if for every set $S \subseteq \mathcal{A} \setminus \{i\}$, $c(S) = c(S \cup \{i\})$, then $phi_i(c) = 0$.
- Additivity: For every two cost functions c_1 and c_2, $phi(c_1 + c_2) = phi(c_1) + phi(c_2)$, where $c_1 + c_2$ is the cost function defined by $(c_1 + c_2)(S) = c_1(S) + c_2(S)$.

3.2 Energy consumption model for wireless sensor networks

In various wireless sensor networks, to achieve maximum network lifetime, each sensor node should minimize the system energy dissipation through cooperation in our research. Therefore, for quantitative analysis of performance, we use a similar model applied in (Han et al., 2007; Handy et al., 2002; Heinzelman, 2000; Heinzelman et al., 2002) for the radio energy consumption where the transmitter consumes energy for radio electronics and power amplifier, and the receiver consumes energy for radio electronics in Fig.4.

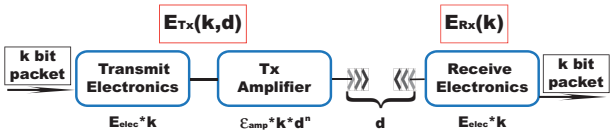


Fig. 4. Radio energy model

In radio propagation models, the free space propagation model (d^2 propagation loss) and the 2-ray ground reflection model (d^4 propagation loss) are used, according to the distance between the transmitter and receiver. The free space propagation model is used to predict received signal strength when the transmitter and receiver have a clear, unobstructed line-of-sight path between them. And the 2-ray ground reflection model is a useful propagation model that is based on geometric optics, and considers both the direct path and a ground reflected propagation path between transmitter and receiver. The cross-over distance between two propagation models is denoted by d_{co}. Power control can be used to invert the loss by setting the power amplifier to ensure a certain power at the receiver. Hence, the expressions for transmitting a message with l-bit over a distance d are:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d); \tag{6}$$

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & : d < d_{co}, \\ lE_{elec} + l\varepsilon_{tr}d^4 & : d \geq d_{co}. \end{cases} \tag{7}$$

And the formula for receiving an l-bit message can be determined by:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec}. \tag{8}$$

For this model, the energy for data aggregation per bit is denoted by E_{DA}. For quantitative analysis, we assume that there are N sensor nodes distributed uniformly in a M×M region with k clusters, the length of each transmission data is l bits. Accordingly, the energy consumption of a cluster head in one frame can be expressed as:

$$E_{CH}(n) = l[n(E_{elec} + E_{\mathrm{DA}}) + \varepsilon_{tr} d_{toBS}^4], \tag{9}$$

where d_{toBS} is the distance between the cluster head and base station, n is the sensor node number in each cluster. Moreover, each sensor node as non cluster head should send its sensing data to the cluster head. The energy dissipation of a non cluster head is presumed to follow the free space model. We assume that d_{toCH} is the distance between the sensor node and the cluster head in the same cluster. Thus, the energy consumption of a non cluster head is:

$$E_{non-CH}(d_{toCH}) = l[E_{elec} + \varepsilon_{fs} d_{toCH}^2]. \tag{10}$$

Then if we assume the area of a cluster is a circle with radius $R = M/\sqrt{k\pi}$ and the cluster head is at the center of the cluster, the expected value of d_{toCH}^2 is derived from (Heinzelman et al., 2002) as follows:

$$E[d_{toCH}^2(k)] = \frac{M^2}{2k\pi}. \tag{11}$$

3.3 Cooperative game theoretic model of clustering

To understand the effect of energy and transmission cost on the clustering, in this paper, we consider the cost sharing game with 3-agents. In the case shown in Fig. 5, the CCH is assumed as the candidate cluster header. We consider the CCH_E and the CCH_D with the redundant energy and the distance from the CCH, respectively. We define this cost sharing game as follows:

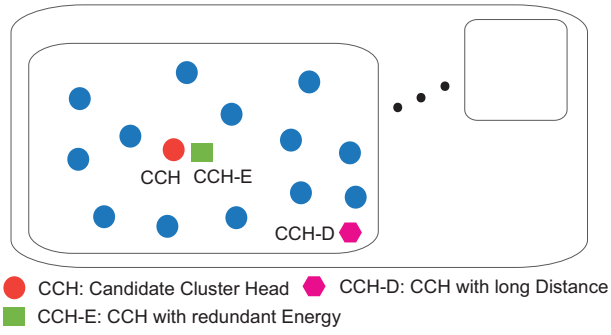


Fig. 5. Cluster architecture for cooperation

Definition 3.2. (Cost Sharing Game for Clustering) *Let $(\mathcal{A}, c)$ be a cost sharing game for clustering in wireless sensor networks. The set of $\mathcal{A} = \{CCH, CCH\_E, CCH\_D\}$ of 3-agents is the candidate cluster headers set. For a coalition set $S \subseteq \mathcal{A}$, the cost function of this coalition is defined as the total energy consumption of all sensor nodes for data collection in one round involving β frames while each agent in S is as a cluster header. Moreover, when chosen as a cluster header, the CCH_E consume the redundant energy firstly. Correspondingly, the total cost should subtract the redundant energy of CCH_E.*

As one of the properties of the Shapley value, anonymity represents that changing the names of agents does not change their cost shares. In order to concentrate on impact on system-wide optimization, we assume that the CCH_E with redundant energy (E_{red}) is close to the CCH and the CCH_D is the farthest sensor node from CCH. Therefore, if the CCH_E is elected as a cluster header, the distance from sensor nodes to cluster header d_k is the same as the value d_{toCH} deduced from k clusters in (11). Contrarily, if the CCH_D is as one of cluster headers, at this time, the distance from sensor nodes to cluster heads d_{2k} should be d_{toCH} derived from $2k$ clusters in the whole region. We denote a coalition of candidate cluster heads as S. Wherefore, the cost function defined by this instance is the following:

$$c(S) = \beta c_{CH}(S) + \beta c_{non-CH}(S) + c_{red}(S); \tag{12}$$

and we assume that $c(\varnothing) = 0$. $c_{CH}(S)$ represents the energy consumption of all cluster heads in S. It can be written as $c_{CH}(S) = sE_{CH}(n/s)$. $c_{non-CH}(S)$ is the energy consumption of all non cluster heads when agents in S are as cluster heads. We can obtain $c_{non-CH}(S)$ as:

$$\begin{cases} (n-s)E_{non-CH}(d_{2k}) & : s > 1 \text{ and } CCH\_D \in S, \\ (n-s-1)E_{non-CH}(d_k) + E_{Tx}(l,d) & : \text{otherwise}, \end{cases} \tag{13}$$

where $s = |S|$ and $E_{Tx}(l,d)$ is transmission energy consumption over the distance between the CCH and the CCH_D. $c_{red}(S)$ represents the redundant energy of the CCH_E when $CCH\_E \in S$. Therefore, we have:

$$c_{red}(S) = \begin{cases} -E_{red} & : CCH\_E \in S, \quad (14a) \\ 0 & : \text{otherwise}. \quad (14b) \end{cases}$$

We consider the cost sharing game for clustering expressed in Definition 3.2. The solution of this game ($\phi_{CCH}, \phi_{CCH\_E}, \phi_{CCH\_D}$) is figured out by the Shapley value from (5). The objective of the model is to achieve global optimization of energy consumption from coalitions of cluster heads. In other words, the solution describes an approach to the fair allocation of cost obtained by cooperation among agents of candidate cluster heads in clustering. Therefore, the fair way to allocate system cost is to allocate energy consumption from each agent considering the capacity of redundant energy and transmission energy. For example, since $\phi_{CCH} + \phi_{CCH\_E} + \phi_{CCH\_D} = c(\{CCH, CCH\_E, CCH\_D\})$, ϕ_{CCH_D} can be described as the fair energy cost allocation of all nodes in the cluster while the CCH_D is elected as a cluster head considering its transmission cost.

4. A Novel Cooperative Clustering Algorithm

4.1 Basic idea

According to the cost allocations from the cost sharing game for clustering, we present the cooperative game theoretic clustering algorithm (CGC) in this section. Different from previous non-cooperative clustering algorithms, our basic idea is that sensor nodes should trade off individual cost with network-wide cost. Consequently, a CCH should cooperate with other capable sensor nodes to form a coalition as cluster heads considering number of sensor nodes in a cluster, the redundant energy and the transmission energy.

4.2 Conditions of cooperation

All sensor nodes participate in the cluster head selection process through our scheme. In the end, competent sensor nodes are elected as cluster heads. If there are no partners, the candidate cluster head is decided to accomplish data collection in the round by itself. At this time, the system energy consumption is $c(\{CCH\})$. Therefore, we can derive conditions of coalitions as follows:

- Cooperate with a sensor node with redundant energy:
  $\phi_{CCH} + \phi_{CCH\_E} < c(\{CCH\})$;
- Cooperate with a sensor node with long distance:
  $\phi_{CCH} + \phi_{CCH\_D} < c(\{CCH\})$.

4.3 Cooperative game theoretic clustering algorithm (CGC)

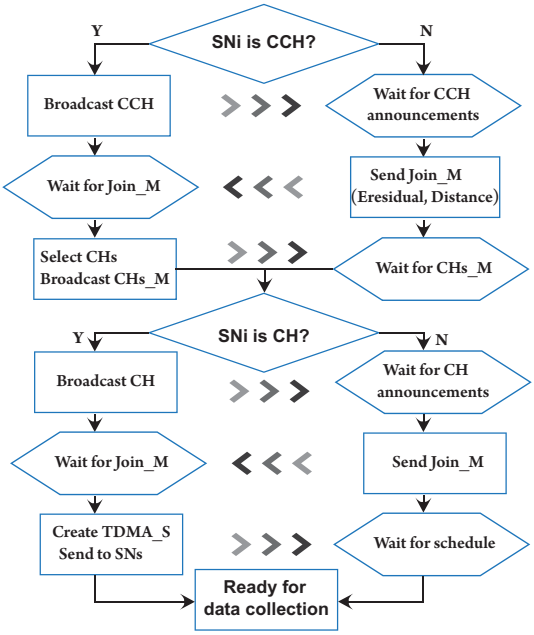


Fig. 6. Flowchart of CGC procedure

As shown in the flowchart of Fig. 6, the CGC processes as follows: at the beginning of round r, each sensor node elects itself to be a candidate cluster head with probability $P_i = \frac{k}{N-k*(r \bmod \frac{N}{k})} \frac{E_{residual}}{E_{initial}}$, which is the similar with DCHS (Handy et al., 2002). Then each CCH broadcasts an advertisement message by carrier-sense multiple access protocol to let other sensor nodes choose the optimum cluster due to received signal strength. Thus, these announcements must be broadcast to reach all of sensor nodes in the area. Afterwards, each non-CCH node sends the join message including sensor node's ID, the residual energy and the distance from the CCH to be concerned with cluster head election. After receiving all join messages of non-CCHs in a cluster, a CCH could adjust the final coalition of cluster heads according to conditions of cooperation mentioned in Section 4.2, where for sensor node i,

$E_{red\_i} = E_{residual\_i} - E_{residual\_CCH}$. Then a CCH broadcasts the set ID of cluster heads, and other sensor nodes listen and wait for the reception of cluster head coalition message. If selected as a cluster head, a sensor node would broadcast an advertisement message to inform other nodes in the network of its decision. Otherwise, non-CHs wait for cluster head announcements and choose the optimum cluster. With that, each non cluster head node sends the join message to the cluster head which is chosen through received signal strength. After receiving all join messages in a cluster, a cluster head creates a time division multiple access schedule according to number of sensor nodes in the current cluster. Finally, it transmits this schedule to ensure that there are no collisions among data transmission and non cluster heads could decrease energy consumption during idle time. After receiving time division multiple access schedules, all sensor nodes get sensing data and transmit it to cluster heads during their allocated time slots. For data collection, cluster heads aggregate individual data from each non cluster head and send condensed summaries to the base station.

5. Simulation and Analysis

In this section, we describe the simulation environment and the analysis of results. Our simulation is based on ns2 and LEACH (Heinzelman, 2000; Heinzelman et al., 2002). The simulation scenarios consist of simplex energy distribution with different position distribution. In the simplex scenarios, the position of each sensor node is random, lattice, semi-lattice and normal distribution, respectively. In the semi-lattice distribution, half of sensor nodes are distributed with lattice method; the others are randomly distributed in the area. Moreover, Fig. 7 and 8 provide a detailed analysis of the simplex scenario with random distribution in the best case. We also present a statistical analysis of other results with the 0.975 confidence in Fig. 9 and 10.

Table 1. Simulation parameter values

Parameter	Value
N	100
M	100m
k	5
d_{co}	86.4m
ε_{fs}	3×10^{-12}J/bit/m^2
ε_{tr}	4×10^{-16}J/bit/m^4
R_b	1Mbps
E_{elec}	0.5nJ/bit
E_{DA}	0.1nJ/bit

5.1 Simulation set-up

In (Daly & Chandrakasan, 2007), a 1Mbps 916.5MHz on-off keying (OOK) transceiver for wireless sensor networks had been designed in a 0.18-μm CMOS process. The minimal receiver power consumption is 0.5mW. Moreover, the noise figure of the Radio Frequency front-end including the 3.5dB loss of the surface acoustic wave (SAW) filter is between 14dB and 15dB for all gain settings, indicating that the tuned low noise amplifier (LNA) dominates the noise figure. Therefore, in our simulation, we set E_{elec} is 0.5nJ/bit for a bit rate (R_b) 1Mbps transceiver,

the thermal noise floor is 99dBm, the receiver noise figure is 14dB and a signal-to-noise ratio(SNR) is at least 28dB to receive the signal with no errors. Thus, the minimum receive power $P_{r-thresh}$ for successful reception is $P_{r-thresh} \leq -57$dBm. With that, the cross-over distance d_{co} is 86.4m. And in (7), ε_{fs} and ε_{tr} are 3×10^{-12}J/bit/m^2 and 4×10^{-16}J/bit/m^4, respectively. Furthermore, the ARM (Advanced RISC Machine) architecture is widely used in embedded designs. For power saving features, ARM CPUs are dominant in wireless sensor networks, where low power consumption is a critical design goal. In recent years, the new version of ARM has been successfully used for many years in a wide range of wireless device application. Building on the Cortex foundation, the processor achieves performance of 2.0DMIPS/MHz, low power of 0.5mW/MHz and speed up to 1GHz. Thus, we assume that the energy consumption of per bit data aggregation (E_{DA}) is 0.1nJ/bit. For our simulation, we assume that 100 sensor nodes are dispersed into the 100m$\times$100m area with 5 clusters and the simulation is finished when the rate of sensor nodes alive is less than 0.1.

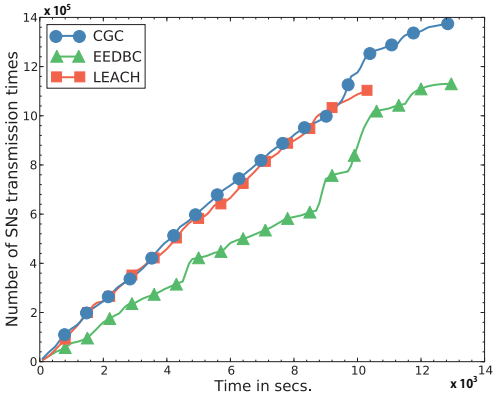


Fig. 7. Lifetime and data capacity

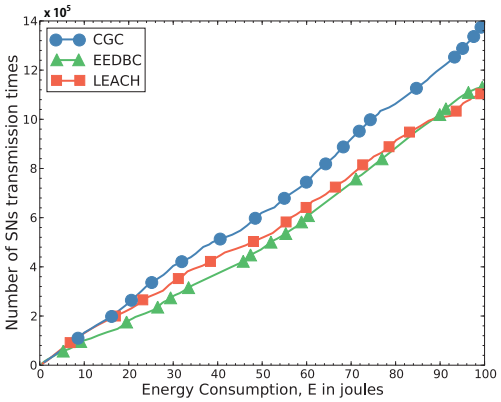


Fig. 8. Energy efficiency

5.2 Analysis of simulation results

In this section, we introduce the results of simplex scenario while the initial energy of a sensor node is 1J and the position of base station is $(50, 175)$. In our simulation, we use the number of sensor nodes transmission times defined as the sum of transmission times for each sensor node to represent the data transmission capacity. The effect of capacity of data transmission on the time is shown in Fig. 7. As illustrated in this figure, both in CGC and EEDBC, the network lifetimes are greatly prolonged more than that of LEACH about 25%. Typically, however, the final number of sensor nodes transmission times is increasing up to 24.5% and 21.6% compared with LEACH and EEDBC, respectively. Accordingly, at the same time, our scheme provides more amount of transmission data to base station. In other words, CGC also reduces the data transmission latency. Fig. 8 compares the three algorithms in terms ofA@energy efficiency defined as the number of sensor nodes transmission times per unit energy. The result shows that CGC is the most efficient scheme and the transmission data per unit energy is delivered up to approximate 22% in the end.

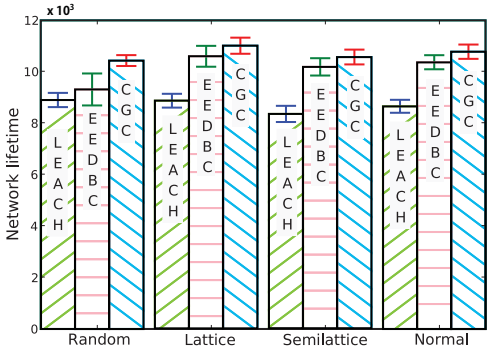


Fig. 9. Statistical analysis of lifetime

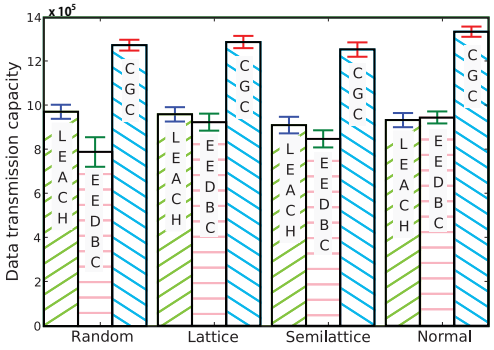


Fig. 10. Statistical analysis of data capacity

From the statistical analysis of network lifetime in Fig. 9 and data transmission capacity in Fig. 10, comparing with other approaches, our scheme can guarantee to prolong network lifetime and improve data transmission capacity up to 5.8% and 35.9%, respectively.

The results of simulation show that CGC outperforms other algorithms on network lifetime, data transmission capacity and energy efficiency with concern of position distributions. Therefore, our scheme can surely guarantee to prolong network lifetime, reduce data transmission latency and improve the utilization of energy.

6. Conclusion

In this chapter, we presented a cooperative game theoretic model for clustering algorithms in wireless sensor networks, which is provided for balancing energy consumption of sensor nodes and increasing network lifetime and stability. Moreover, from feasible allocations of energy cost as the results of this model, we proposed and analyzed the cooperative clustering algorithm to obtain system-wide optimization from conditions of cooperation, considering the redundant energy, communication costs and number of sensor nodes in a cluster adapting to various wireless sensor networks. The basic idea is that each sensor node should trade off individual cost with network-wide cost. Consequently, each capable sensor node should cooperate with others in cluster formation for collective decision-making. Furthermore, we presented performance evaluation and comparison of the existing clustering algorithms with our approach quantitatively with respect to network lifetime, data transmission capacity and energy efficiency. We provided a detailed analysis of the simplex scenario with random position distribution in the best case and a statistical analysis of the scenarios with different position distributions including random, lattice, semi-lattice and normal distributions. Comparing with other approaches through simulations, our protocol can surely guarantee to prolong network lifetime and improve data transmission capacity up to 5.8% and 35.9%, respectively.

7. References

Abbasi, A. A. & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks, *Computer Communications* **Vol. 30**(No. 14-15): 2826–2841.

Akyildiz, I., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey, *Computer Networks: The International Journal of Computer and Telecommunications Networking* **Vol. 38**(No. 4): 393–422.

Daly, D. & Chandrakasan, A. (2007). An energy-efficient ook transceiver for wireless sensor networks, *IEEE Journal Solid-State Circuits* **Vol. 42**(No. 5): 1003–1011.

Felegyhazi, M., Hubaux, J.-P. & Buttyan, L. (2006). Nash equilibria of packet forwarding strategies in wireless ad hoc networks, *IEEE Transactions on Mobile Computing* **Vol. 5**(No. 5): 463–476.

Hac, A. (2003). *Wireless Sensor Network Designs*, John Wiley and Sons.

Han, Y., Park, S., Eom, J. & Chung, T. (2007). Energy-efficient distance based clustering routing scheme for wireless sensor networks, *Lecture Notes in Computer Science, Computational Science and Its Applications* **Vol. 4706/2007**: 195–206.

Handy, M. J., Haase, M. & Timmermann, D. (2002). Low energy adaptive clustering hierarchy with deterministic cluster-head selection, *Proceedings of 4th IEEE Conference on mobile and wireless communications network*, pp. 368–372.

Heinzelman, W. (2000). Application-specific protocol architectures for wireless networks, *Ph.D. thesis, Massachusetts Institute of Technology* .

Heinzelman, W., Chandrakasan, A. & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications* **Vol. 1**(No. 14): 660–670.

Machado, R. & Tekinaya, S. (2008). A survey of game-theoretic approaches in wireless sensor networks, *Computer Networks: The International Journal of Computer and Telecommunications Networking* **Vol. 52**(No. 16): 3047–3061.

Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V. V. (2007). *Algorithmic Game Theory*, Cambridge University Press.

Younis, M., Youssef, M. & Arisha, K. (2003). Energy-aware management for cluster-based sensor networks, *Computer Networks* **Vol. 43**(No. 5): 649–668.

Younis, O. & Fahmy, S. (2004). Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on Mobile Computing* **Vol. 2**(No. 4): 366–379.

Zheng, Z., Wu, Z. & Lin, H. (2004). Clustering routing algorithm using game-theoretic techniques for wsns, *Proceedings of the 2004 international symposium on circuits and systems*, pp. IV–904–7.

10

A Cluster Head Election Method for Equal Cluster Size in Wireless Sensor Network

Choon-Sung Nam[1], Kyung-Soo Jang[2] and Dong-Ryeol Shin[1]
Sungkyunkwan University[1] and
Kyungin women's college[2]

1. Introduction

Wireless sensor networks (WSNs) are composed of many homogeneous or heterogeneous sensor nodes with limited resources. A sensor node is comprised of three components: a sensor, a processor and a wireless communication device. A sensor of nodes detect a change in surroundings, a processor processes sensing data collected from neighbour nodes or own environmental information, and a wireless communication device is capable to send and receive sensing data.

Sensor networks consist of a great number of sensor nodes and one or several sink nodes. The role of a sensor node is to detect and process own environmental information, to convert it to sensing data, to send it to neighbour nodes or sink nodes, and to collect it from neighbour nodes. On the other hands, the role of a sink node is to collect sensing data from sensor nodes and to be gateway that interconnects different network and transmits data to it.

Generally, sensor nodes of WSNs are randomly scattered on specific area for satisfying user's requirements (detecting, observing and monitoring environment) and have to self-organized network. It is difficult to exchange and charge node battery as the area where sensor nodes are located in is inaccessible location. So, it is important issue to design power-efficient protocol method for low-power operation and prolonging the network lifetime (Akyildiz et al, 2002).

A sensor node needs wireless ad-hoc network capability to collect sensing data of wireless sensor network without a communication infrastructure. Sensor networks are, however, not suitable for the existing ad-hoc routing method (Tubaishat & Madria, 2003) because of sensor nodes with limited capability. Thus sensor networks require wireless ad-hoc routing method considering self-organization, restrictive power, and data-based communication(Sohrabi et al, 2000) and need multi-hop routing mechanism because of the limited transmission radius of a sensor nodes.

WSNs should design for routing algorithm considering low-power operation because it has limited features and is a traditional wireless networks completely different from 'the network(Al-Karaki & A.E. Kamal, 2004). In WSNs, routing methods can divide into two routing mechanisms: 'flat-routing' and 'hierarchical-routing'. The 'flat-routing' technique regards the whole network as one region, enabling all nodes to participate in one region. On

the other hands, the 'hierarchical-routing' technique is to execute local cluster routing scheme based on clustering.
The feature of sensing data is that adjacent sensor nodes have similar or same sensing data(Ameer Ahmed Abbasi and Mohamed Younis, 2007). That is, the duplicate sensing data exist in sensor networks. To prevent duplicate sensing data, the 'hierarchical-routing' technique uses the clustering scheme. The Cluster region is a local area assigned by user's requirement. It is composed of a cluster head node and member nodes. A cluster head is for aggregating sensing data from member nodes. The number of sensing data in the 'hierarchical-routing' is lower as cluster head works. Thus, the 'hierarchical-routing' is more energy-efficient routing technique than the 'flat-routing'.
A process of clustering is as follows. First, a sink node elects cluster heads among all scattered sensor nodes. Each cluster head makes a local cluster by using advertisement message. Member nodes send sensing data to own cluster head. A cluster head collects sensing data from member nodes for 'data-aggregation' that prevents duplicate data. When a sink node requests user-demand, in response to user-demand, a cluster head prevents unnecessary query flooding. To communicate with sensor nodes which are outside sensing range, a sensor node is suitable for multi-hop networking(Toumpis & Goldsmith, 2003). It is important to measure the number of cluster member nodes in local cluster based on multi-hop clustering. If there are many member nodes in local cluster, the energy consumption in a local cluster is increased. The energy drain of a cluster head is also increased. On the other hand, if there are little member nodes in a local cluster, the energy consumption is low. The energy drain of a cluster head is also low. Thus, it is important how many member nodes are needed to set up a local cluster for energy-efficient sensor networks.
This chapter shows energy-efficient cluster formation method. To achieve this, a local cluster should know the number of optimal member nodes and adjusts the position of a cluster head considering the distance between cluster heads and member nodes. That is to build balance among local clusters. Thus, this method can find low-power mechanism of sensor networks for clustering.
The organization of this chapter is as followings: in section 2, we shows an overview of previous clustering methods and describe problems of them. In section 3, we present the cluster head election method for equal size. In section 4, we compare previous methods with the proposed method, and analyze them. Finally, in section 5, we present conclusion and future works.

2. Clustering mechanism for sensor networks

2.1 Cluster head selection with random costs

The typical clustering method is LEACH(Heinzelman et al, 2000). LEACH is a routing method based on clustering for distribution energy consumption of wireless sensor networks. The feature of LEACH is a clustering method to distribute energy consumption to all sensor nodes in sensor networks. To achieve this, LEACH elects randomly a cluster head which aggregates sensing data from member nodes in local cluster and processes them for managing a local cluster workload. LEACH consists of two stages: 'set-up' stage and 'steady-state'. The 'set-up' stage is to form a cluster and the 'steady-state' stage is to comprise of several TDMA frames. In 'set-up' stage, all sensor nodes select a cluster head by threshold T(n) in equation 1. Each node selects random number between 0(zero) and 1(one).

If the selected number is a smaller number than threshold T(n), the node that has a smaller number is a cluster head in the current round.

$$T(i) = \begin{cases} \dfrac{p}{1 - p * (r \bmod \frac{1}{p})}, & i \in G \\ 0 \qquad\qquad\qquad\qquad , otherwise \end{cases} \tag{1}$$

In equation (1), p is the ration of a cluster head, r is the current round, and G is a set of nodes that were not a cluster head in 1/p round. By equation (1), all nodes only become a cluster head among 1/p round once. The more round is increased, the more probability which a node becomes a cluster head is increased. After 1/p round, a node can become a cluster head with same probability, again. The energy drain of cluster head is so bigger than a member node because of aggregating, processing and sending sensing data from member nodes. To prolong sensor network lifetime, a cluster head have to be circulated. Through this mechanism, LEACH can circulate equally a cluster head. A fair distribution of cluster head selection might make equal energy consumption of cluster heads and be probable for fair energy consumption of all sensor nodes in sensor networks.

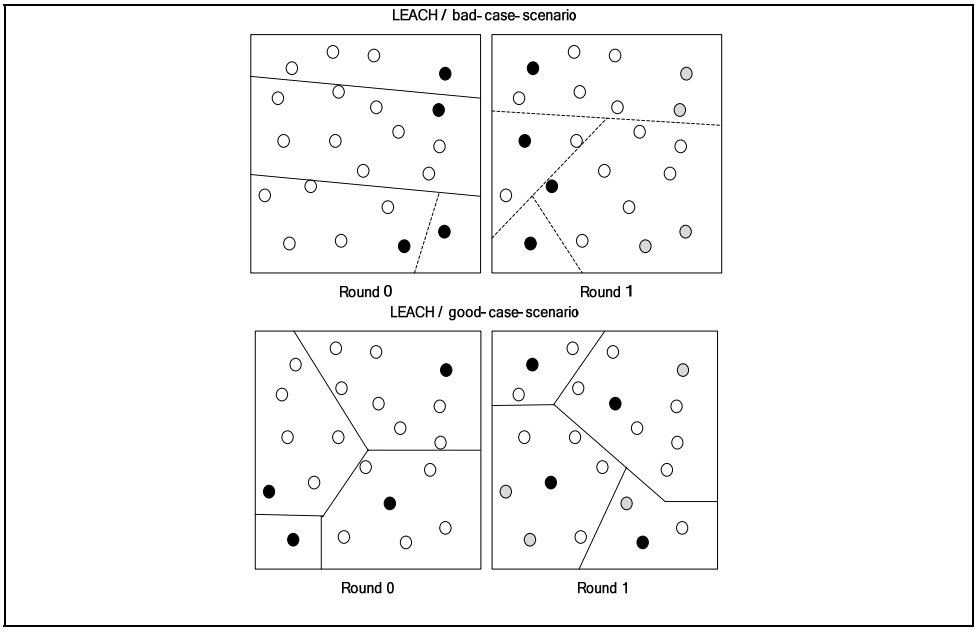


Fig. 1. Cluster formation in LEACH

When LEACH organizes a cluster, it can form equally a cluster (good-case-scenario) or not (bad-case-scenario). In LEACH, as a local cluster is organized by the selected cluster head, location of cluster heads affects the number of member nodes in a local cluster. If there are many member nodes in local cluster, the energy spending of a cluster head is increased. On the other hand, if there are little member nodes in local cluster, the energy consumption of a

cluster head is decreased. That is, that the energy consumption of cluster head is affected by the number of member nodes. As a result, in LEACH, it is difficult to keep up the balance of node energy of whole sensor networks.

In LEACH, all member nodes delivery sensing data directly to a cluster head or the sink node because LEACH assumes transmit power control. However, a sensor node is suitable for communicating the node with outside sensing range based on multi-hop routing method because of node's communication limited(Gutierrez et al, 2001, Noseong Park et al, 2005). That is, in case of outside the range of a cluster head or the sink node, sensor networks should organize clustering using multi-hop routing mechanism.

LEACH-C(LEACH-Centralized)(Heinzelman et al, 2002) is similar to LEACH. That means that two algorithms are same to data transmission processes between the BS and the sensor nodes. On the other hand, the process of cluster head selection in LEACH-C is different with LEACH. LEACH-C uses a central control algorithm to form the clusters that may produce better clusters by dispersing the cluster head nodes throughout the network. During the set-up phase of LEACH-C, each node sends information about its current location (possibly determined using a GPS receiver) and energy level to a sink node. A sink computes the average energy level of all nodes by received message, and then give the right which is not possible for the cluster heads if the sensor node have lower energy than the average energy level. Using the remaining nodes as possible cluster heads, the BS finds clusters using the simulated annealing algorithm(Murata & Ishibuchi, 1994) to solve the NP-hard problem of finding optimal clusters(Agarwal & Procopiuc, 1999). This algorithm attempts to minimize the amount of energy for the non-cluster head nodes to transmit their data to the cluster head, by minimizing the total sum of squared distance between all the non-cluster head nodes and the closest cluster head. After the cluster heads are elected, member nodesf can select the cluster head which they can communicate with minimum energy consumption. A cluster is organized by the node transmitting the message as a determined cluster head node. After clustering, The cluster heads perform TDMA scheduling, transmit the schedule to member nodes in local clusters, and then start the data transmission time. The strong point of LEACH-C is that it can equally distribute waste to energy between sensor nodes by positioning cluster heads into the center of cluster. A sensor node, however, should be loaded with GPS receiver set. And it has not still guaranteed balance of energy consumption of whole sensor networks. This technique makes the price of sensor nodes increase high. Because of a number of sensor nodes to be needed for the network ranges from hundreds to hundred-thousands, this technique is not appropriate(Handy et al, 2005).

Above two methods increase the energy consumption because of additional overhead for knowing the energy level. To achieve this problem, HEED(Younis & Fahm, 2004) proposes the cluster head selection method using by distributed processing. HEED can select the cluster heads only considering the parameters of nodes. In HEED, the cluster head election should use only local data, have low amount of data for clustering and be completed in a certain period of time. Thus the advantages of HEED are that algorithm time terminate in a certain period of time regardless of cluster size and do not consider the location of nodes. HEED do not also guarantee the equal distribution of the cluster heads in networks like LEACH and LEACH-C.

2.2 Cluster head selection with equal member nodes

ACHS(Adaptive Cluster Head Selection)(Choon-Sung Nam, 2008) is the method to divide unequal cluster size into equal cluster size for balance of energy consumption in a local cluster. In case the number of member nodes per a local cluster is more or less than average number of member nodes, this cluster could be an unequal cluster. To solve unfairness among local clusters, ACHS re-selects cluster heads using by distance between cluster heads and between member nodes and a cluster head. This method is as follows. First, the sink node elects a cluster head randomly like LEACH equation (1). The selected cluster head informs neighbor nodes for an advertisement message. In response to the message, each member node registers with own cluster head. A cluster head sets up and stores the farthest member node (FMN) with cache memory among member nodes. In the same way, it keeps the shortest cluster head (SCH) with cache. If the difference of FMN and SCH is same, this means that local clusters are divided into equal cluster size.

In Fig. 2-(a), if the gap of FMN is longer than SCH, in case of cluster head 'A', the cluster size is bigger than neighboring cluster size as the cluster which has cluster head 'A' invades a domain of neighboring cluster which has cluster head 'B'. In other words, that cluster size is bigger means that the number of member nodes is so more. Thus the cluster head 'A' should be moved to FMN as difference between FMN and SCN, and is reselected a cluster head among near nodes. If the gap of FMN is shorter than SCH, in case of cluster head 'B', the neighboring cluster size is bigger than the cluster size of 'B' as the neighboring cluster 'A' invades own domain. Thus, the cluster head 'B' moves to SCH as difference between FMN and SCH, and is reselected a cluster head among near nodes. After these processes, a local cluster would be divided equally like Fig.2-(b).

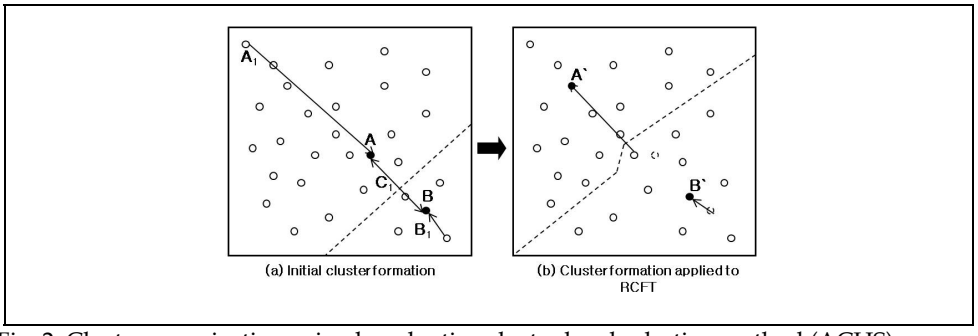


Fig. 2. Cluster organization using by adaptive cluster head selection method (ACHS)

ACHS used direct data transmission method that computed the distance between cluster heads and member nodes. ACHS has the same problem on communication range like LEACH. In case of outside transmission range, it cannot communicate with outside nodes. As a result, it is difficult to establish scalable network. Thus ACHS also need to multi-hop routing method for clustering. Another problem has to be to reorganizes the equal cluster unnecessarily for equal clusters although previous established local cluster is equal.

3. Cluster Head Election Method for Equal Cluster Size

3.1 Cluster head capacity

This method is for energy distribution as all sensor nodes would be selected as a cluster head after 1/p round. And it helps efficient-energy saving of nodes since the nodes which has high remaining energy are elected as a cluster head. However, it does not consider unequal energy consumption of nodes by unequal clusters. The elected cluster head is not again selected as a cluster head during 1/p rounds although the node has the most energy than others.

Above described, we knew that the energy gap between a cluster head and a member node is big during managing clustering. This reason is as following: A member nodes just detects own surrounding environment and transmit the sensing data to a cluster head. A mount of aggregated data produced by a cluster head depends on the number of own member nodes. Thus a cluster head should be selected by energy drain ratio as setting up threshold, T(i).

As shown equation (2), if r is 0, r=0, the probability of all sensor nodes, T(i)r=0, is 'p' because all sensor nodes have not been selected as a cluster head.

$$T_{i=0}(i) = \begin{cases} \dfrac{p}{1 - p * (r \bmod \dfrac{1}{p})} = p, i \in G \end{cases} \tag{2}$$

If r >0, the threshold value of a node that is selected as a cluster head is reduced by amount of energy consumption. The consumption energy ratio, $E_{ch}/E_{initial}$, added to the previous threshold value is the next threshold value. E_{ch} is amount of energy drain of a cluster head and $E_{Initial}$ is initial energy of nodes. If a node is a member node, the consumption energy ratio, E_{mem}/E_{inital}, subtracted from the previous threshold is the next threshold value. This is as following:

$$T_{i=0}(i) = \begin{cases} T(i)_{r-1} - \dfrac{E_{mem}}{E_{Initial}}, i \in G_{r-1} \\ T(i)_{r-1} - \dfrac{E_{ch}}{E_{Initial}}, otherwise \end{cases} \tag{3}$$

Except for the case that E_{ch} is same as E_{mem}, all nodes are selected as a cluster head at least once during 1/p rounds. In next rounds of cluster head selection, the nodes' threshold value that is used with cluster head selection is different as is a cluster head energy consumption in own local cluster. This difference is from the fact that the number of member nodes in local cluster varies from each other. If a cluster head has fewer member nodes than the average number of member nodes, the threshold value is also lower. This means that the cluster head is re-selected as a cluster head during 1/p rounds. This will result in energy distribution of sensor networks and increasing network life time.

3.2 Equal cluster size

In direct communication, if sensor nodes are located out of transmission range, cluster heads should be more selected for connecting nodes. To configure the scalable sensor networks,

the clustering method should use multi-hop communication. For cluster formation adapted multi-hop routing, a local cluster should be organized by the selected cluster head. First, a sink node selects a cluster head, 5% nodes among all nodes, like LEACH. The selected cluster head sends the ADV message to neighbour nodes with 1(one) hop for collecting member nodes. Nodes which received the message repeat this process until they meet the nodes of another local cluster. The nodes which received the ADV message judge what kind of a cluster head. The nodes set up a cluster head as the cluster head id (CHid) included the ADV message, increase their hop-count by one and reply the REP message to own cluster head. And then a cluster head registers own sensor id. Through this process, a cluster head can know the number of own member nodes and hop counts between own and member nodes(Choonsung Nam, 2008)
The pseudo code of clustering process based on multi-hop is as follows.

```
Procedure cluster formation
Input     selected cluster head id
Output    node Information belonging to cluster
------------------------------------------------
If received ADV from cluster head Then
  Begin
    If (Node.My_CHid != null )
      insert into Node_Info_values(CHid, Hopcnt++)
      reply REP to sender
      send ADV message to neighbor nodes
      return true
    Else
      return false
  End
------------------------------------------------
ADV              Advertisement message
REP              Respond message
CHid             Cluster head id
Hopcnt           Hop count
Node_Info_value  Node information value
```

Fig. 3. Pseudo code for clustering process based on multi-hop

To prevent unequal cluster formation, above method only proposed equal cluster formation technique using difference between the FMN and the SCH. To balance the clusters, we add above method to the method which is to balance the number of member nodes. For example, in Figure 20, 200 sensor nodes are located in 10 x 10 grid structure. The cluster head is gray circle A, B, C, D and E, 5% among 100 sensor nodes. By multi-hop clustering method based on the CH, a cluster can be organized local cluster like a dotted line. The alphabet 'A', 'B', 'C', 'D' and 'E' are the CHs. The number of member nodes each CH has is that A is 21, B is 16, C is 14, D is 21, and E is 23. Above mentioned, a cluster head can know the number of own member nodes and the adaptive number of member nodes. In this example, the adaptive number of member nodes is 19, (all sensor nodes / cluster heads). So, cluster head 'A' and 'D' is adaptive cluster distribution. The cluster head 'B', 'C' and 'E' is not adaptive. To balance the clusters, the clsuter heads are replaced with the dark circle 'A', 'D', and 'E'. Cluster head 'B' and 'E' is not replaced because the hop count of FMN and SCH

is same. The change of cluster area is black line. The number of cluster member nodes (black line) is that A is 21, B is 18, C is 10, D is 22, and E is 24. That is unequal cluster division than previous cluster formation. Cluster 'E' is changed more unequal cluster size. Specially, cluster 'C' is more unequal cluster size than before. The cases of imbalance cluster are as following:

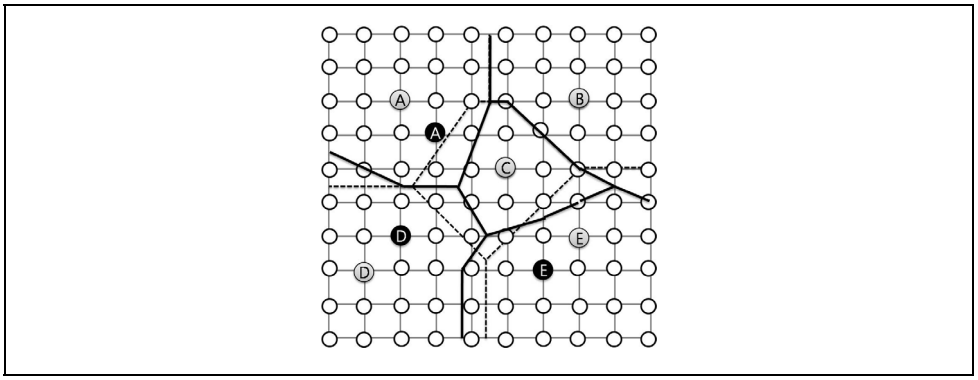


Fig. 4. Imbalance of a local cluster by changing cluster heads

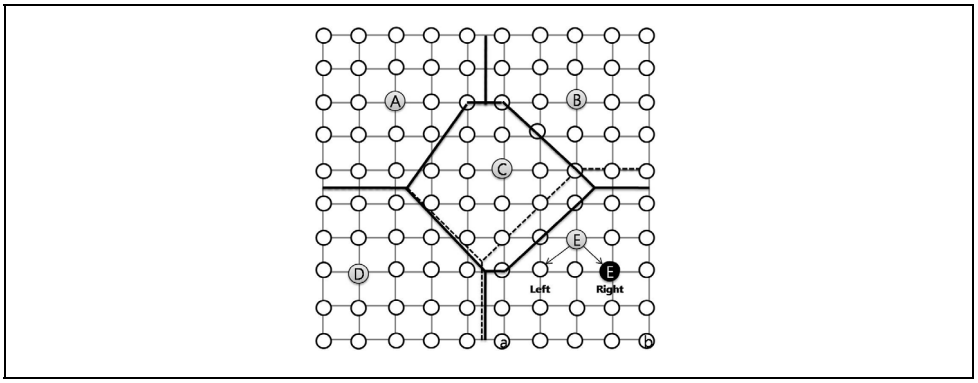


Fig. 5. Balance of a local cluster by keeping the adaptive clusters

Although a local cluster has adaptive number of member nodes(all nodes/th number of cluster heads), the replacement of cluster head is elected to only balance the size of local cluster. This method do not guarantee adaptive local cluster as the previous adaptive local clusters are changed. If local clusters are imbalance, the replacement of cluster head should be selected by the current cluster head for balancing clusters. The previous method does not have the condition which node is better as a cluster head with same distance or hop counts. To achieve this problem, we don't change the adaptive cluster and change only unequal cluster. We define the adaptive cluster that has the number of member nodes with plus or minus 10% of the adaptive number of member nodes. That is from 17 to 21. In Fig.5, the equal local cluster is 'A' and 'D'. The unequal local cluster is 'B', 'C' and 'E'. The proposed method changes them. Cluster 'B' and 'C' have same distance between the FMN and the

SCH and they don't re-select their cluster head. According this method, cluster 'E' is only replaced. The SCH of cluster 'E' is the cluster 'C' and the hop count of it is 2. The FMN of cluster 'E' is node 'a' or 'b', and hop count of it is 3. Cluster head 'E' should move to the FMN ('a' or 'b') as 1 hop as the difference between the FMN ('a' or 'b') and the SCH ('C') is 1. At this time, the cluster head 'E' should decide node 'a' or 'b' as the FMN. The 'E' selects node 'b' as the FMN because node 'b' is farther than 'a' from the SCH 'E'. The farther difference between 'C' and 'E', the more member nodes 'C' gets. The number of cluster member nodes by the proposed method is that A is 21, B is 18, C is 17, D is 21 and E is 18. Therefore, all local clusters are more equal clustering than above methods.

This result is shown Table 5. The standard deviation of adaptive cluster member nodes shows that the proposed method is the best.

Random cluster selection		ACHS		The proposed method	
A	21*	A	21*	A	21*
B	16	B	18*	B	18*
C	14	C	10	C	14
D	21*	D	22*	D	21*
E	23	E	24	E	23
stdev	3.4	stdev	4.9	stedv	3.1

Table 1. The number of member nodes in a local cluster

```
Procedure  reselecting cluster head
Input      selected cluster head id
Output     reselected cluster head id
-----------------------------------------------
If selected cluster head id Then
   Begin
     If the optimal number of cluster heads
       become EC
     Else
       check Diff=difference between SCH and FMN
       If  Diff=0
         become EC
       If  Diff>0
         select farther FMN from SCH
         move to SCH as far as Diff-hop(s)
       If  Diff<0
         select farther SCH from FMN
         move to FMN as far as Diff-hop(s)
   End
-----------------------------------------------
EC    Equal cluster
FMN   the farthest member node
SCH   the shortest cluster head
```

Fig. 6. Pseudo code for improved clustering

In pseudo code of Fig. 6, if the node are elected as a cluster head, it determine to have the adaptive member nodes. If it has the adaptive member nodes, the node, the current cluster head, is not changed. If it not, it determine to change the replacement of cluster heads considering three conditions. The three conditions are same to the direct communication conditions. However, in case the replacement of cluster heads have same distance, the proposed method always selects the node far from the current CH.

4. Performance evaluation and analysis

4.1 Energy model for sensor networks

We assumes the sensor energy model for radio hardware energy dissipation, like figure 10. This model can divide the transmitter energy to run the radio electronics and the power amplifier, and the receiver energy to run the radio electronics and have two channel model: the free space (d^2, distance, power loss) and the multipath fading(d^4 power loss) channel models. This model depends on the distance between the transmitter and receiver(Rappaport, 1996). Power control can be used to invert this loss by appropriately setting the power amplifier. if the distance is less than a threshold d0, the free space (fs) model is used; otherwise, the multipath(mp) model is used. Thus, to transmit an l-bit message a distance d, the radio expends

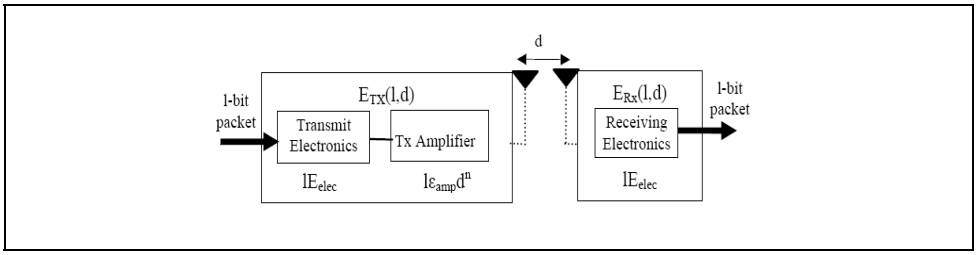


Fig. 7. Radio energy dissipation model

$$E_{Tx}(l,d) = E_{Tx-elec}(l) - E_{Tx-amp}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\varepsilon_{fs}d^4, & d >= d_0 \end{cases} \quad (4)$$

and to receive this message the radio expends:

$$E_{Rx}(l) = E_{Rx} - E_{elec}(l) = lE_{elec} \quad (5)$$

The electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $e_{fs}d^2$ or $e_{mp}d^4$, depends on the distance to the receiver and the acceptable bit-error rate. for the experiments described in this paper, the communication energy parameters are set as E_{elec}=50nJ/bit, e_{fs}=10pJ/bit/m^2 and e_{mp}=0.0013pJ/bit/m^4. Using previous experimental results(Wang et al, 1999), the energy for data aggregation is set as EDA=5nJ/bit/signal.

If the minimum distance of the multipath channel is same to the maximum distance of the free channel, we can know the minimum distance of the multipath channel by the following equation.

$$\begin{aligned} lE_{elec} + l\varepsilon_{mp}d^4 &= lE_{elec} + l\varepsilon_{fs}d^2 \\ l\varepsilon_{mp}d^4 &= l\varepsilon_{fs}d^2 \\ d &= 87.705 \end{aligned} \tag{6}$$

Above equation (6), the minimum channel of the multipath channel is about 87.7m. However, as the transmission range of regular sensor nodes is shorter than it, the channel of WSNs should be the free channel based on multi-hop routing

4.2 Network model for sensor networks

For network configuration, we assume the following network topology, as described in Table 4. We set up the size of the networks to be 100 meter x 100 meter, with a possible communication radius of a node, R, at 10 meters. To prevent an isolation node, the number of network nodes is 300. The sensor node's initial energy is 1 J (Joule) and the data packets of a node are 525 bytes between a cluster-head and member node, and a sink and a cluster-head. As described previously, a sink node is located outside of the sensor networks with the distance between a sink and the networks defined as R. It is shown in table 2.

Network size	100 m^2
The nmber of sensor nodes, N	300
Radius of sensor	10m
Length of each packet	525bytes
E_{elec}	50nJ/bit
E_{amp}	10pJ/bit/m^2
EDA	5nJ/bit

Table 2. The number of member nodes in a local cluster

4.3 Analysis for cluster head capacity

When frist round, the proposed method is almost equal to a previous method. Thus we will compare the average energy consumption of nodes when r>1. We assume that '1' round time is the time to select cluster head 20 times. In figure 12, gray dots show the nodes when using the cluster head selection method of LEACH and black dots when proposed method. When using proposed method, the average round of nodes is higher. That means that the energy re-selected nodes are lower than other node's energy and the energy distribution is good by selecting the node with the lowest remaining energy.

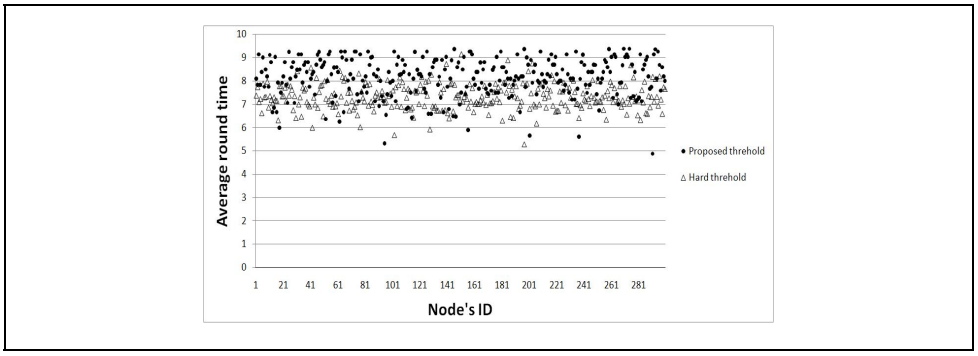


Fig. 8. Average round time of nodes

Fig. 9 shows survival rate of nodes. Node alive rounds of proposed method are longer than the method like LEACH. That means that LEACH cannot control to distribute overload of a cluster head. As the proposed method considered unequal clustering, overload of a cluster head, the nodes that used this method live longer than LEACH. As the round progresses, we can know survival rate of the proposed method is higher than LEACH. Since the percentage of alive nodes are 90%(0.9), the nodes of LEACH dramatically died than the proposed method. When the alive rate is 10%(0.1), they died slowly as the remaining nodes have few member nodes. Since 90%, the nodes of the proposed method, on the other hand, died slowly than LEACH as distributing energy consumption.

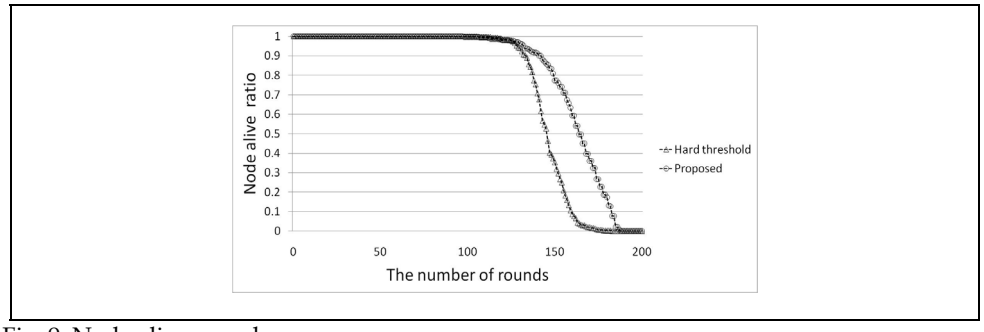


Fig. 9. Node alive round

4.4 Analysis of the number of cluster member nodes

We measured the number of member nodes and hop count in local cluster. Each node is chosen for a cluster head with equal probability. After cluster head election about 20 times, one round comes to an end. We repeated this process 10 times. We gained the result of average value and obtained the standard deviation of standard variation and clustering. The lower standard deviation, the more equal a cluster forms.

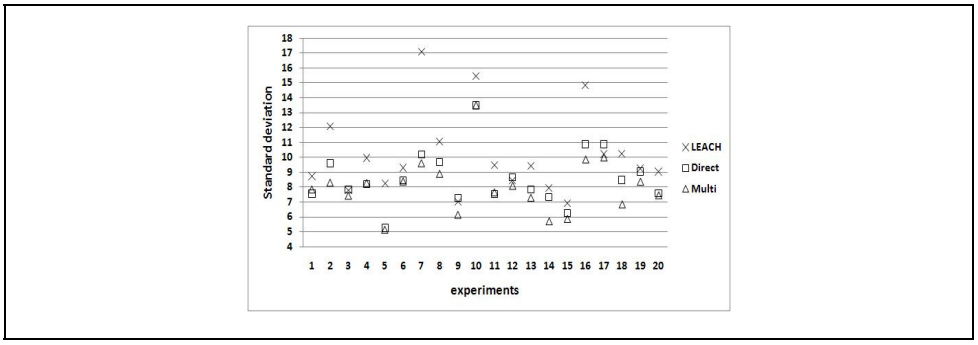


Fig. 10. The standard deviation of member nodes

Fig. 10 shows the standard deviation (STDEV) of member nodes in local cluster. Above figure, LEACH is higher than other algorithm. On the other hand, Direct(direct communication) and Multi-hop(multi-hop communication) are lower than LEACH. In case of the standard deviation of LEACH, experiments number 2, 7 and 16, a cluster is bad-case-scenario. In bad-case, Direct and Multi-hop can reduce STDEV of member nodes. In experiments number 3, 9 and 12, Direct is higher than LEACH. This means that Direct can form unequal clustering, compared with cluster formation. In case of the proposed method Multi-hop, it has little lower value than LEACH and Direct. Also, as shown in Fig. 11, Multi-hop has the lowest average standard deviation value of member nodes. So, Multi-hop can organize more equal cluster size than LEACH and Direct.

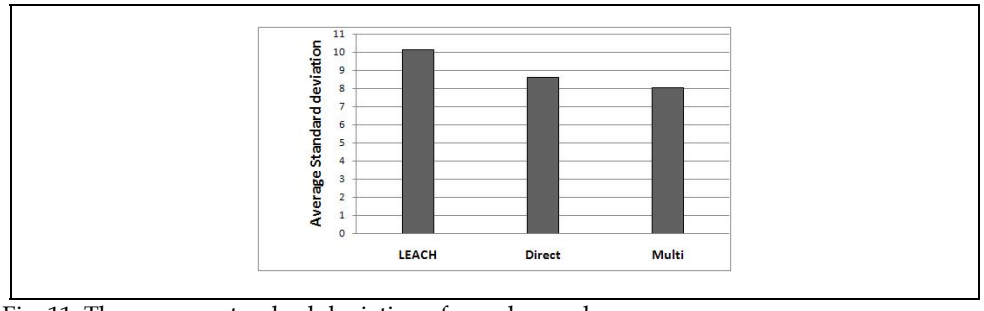


Fig. 11. The average standard deviation of member nodes

Although a cluster is formed equally, if it is long distance between a cluster head and nodes, communication cost between two nodes is increased. And we measured the average hop count of local cluster. As a result figure 24, Multi-hop has lower hop count value than LEACH and Direct. This means that Multi-hop reduces the distance between a cluster head and member nodes and communication cost of sensor nodes and a cluster head in local cluster. So, Multi-hop can form a cluster that has the adaptive member nodes and reduce energy consumption of whole sensor networks.

4.5 Finding optimal number of member nodes

We assume the number of optimal member nodes is (N/CHnum-1). We make an experiment on the standard deviation per a local cluster and the energy consumption of member nodes. In experiment, we configure the optimal member nodes as 5%~100% among member nodes and measure the energy efficiency of a local cluster.

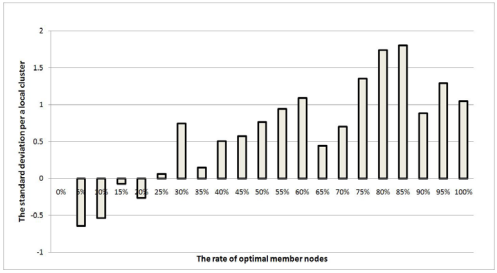


Fig. 12. Comparing with standard deviation of member nodes

Fig. 12 shows the standard deviation per a local cluster as increased the optimal number of member nodes. If the optimal number is 0%, like the direct communication method, the standard deviation value is zero because the optimal number is same. In case of the number of member nodes between 5 and 20 percent, we can show the standard deviation per a cluster is decreased. The low standard deviation value means more equal clustering and the higher value means low equal clustering. And the low value can decrease the amount of data packet.

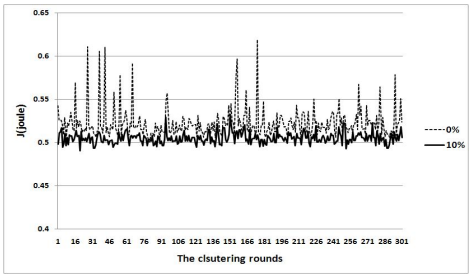


Fig. 13. Energy consumption for clustering

Fig. 13 shows comparing 0% and 10%. The 10% has lower energy consumption than 0%. The reason is as following. First reason is more permissible range. Second reason is more equal member nodes. Third reason is less data packet. Fourth reason is energy distribution.

5. Conclusion

This thesis proposed new optimized clustering algorithm through cluster head selection focused on reducing energy consumption of local clusters and overall networks. It elected the cluster head among nodes which are possible for the cluster head and proved the energy efficiency by comparing previous methods. It is performed by the network scalability and energy consumption. To achieve this, we obtained the energy consumption in Intra-cluster and Inter-cluster, and then we could find the average energy of overall network. Finally, we proposed the re-electing cluster heads method for balancing local clusters. This method uses the information which the cluster heads have. This information is the number of member nodes and distance between the member nodes and the cluster head. Thus the new cluster heads can be elected by this information.

Further works will be intended to compare and analyze the above the methods, and find the optimization clustering algorithm. To achieve this, we have to perform the experiments which are load balancing between member nodes and local clusters, and fault-tolerance in Intra-cluster and Inter-cluster. For load balancing, we would calculate the number of packets from nodes and the packet success ration of sensing data. And for fault-tolerance we would measure the data delay time of sensing data and prove the strong connectivity, which is an means of supplementing route path when the node failure. Through these experiments, we will find the optimization clustering algorithm in WSNs.

6. References

Akyildiz, I.F.; W. Su, Y. Sankarasubramaniam, & E. Cayirci. (2002)."A Survey on Sensor Networks", *IEEE Communication Magazine*, August 2002, pp. 102-114.

Ameer Ahmed Abbasi; Mohamed Younis. (2007). "a survey on clustering algorithms for wireless sensor networks," *Elsevier Journal of Computer Communications*, 30 : 2826-2841, 2007.

A. Wang; W. Heizelman, A. Chandrakasan. (1999). "Energy-scalable protocols for battery-operated microsensor networks," *Proceeding 1999 IEEE Workshop Singnal Processing Systems (SiPS '99)*, pp. 483-492, Oct. 1999.

Choon-Sung Nam; Hee-Jin Jeong , Yiseok Jeong, Dong-Ryeol Shin. (2008). "Routing Technique Based on Clustering for Data Duplication Prevention in Wireless Sensor Networks", *Proceedings of International Ubiquitous Workshop*, Jan. 2008.

Choon-Sung Nam; Hee-Jin Jeong, Dong-Ryeol Shin. (2008). "The Adaptive Cluster Head Selection in Wireless Sensor Networks", *Proceedings of IEEE International Workshiop on Semantic Computing and Applications*, pp. 147-149, 2008.

Fernandess; D. Malkhi. (2002). "K-clustering in wireless ad hoc networks," *Proceedings of the 2nd ACM international Workshop on Principles of Mobile Computing (POMC'02)*, Toulouse, France, October 2002.

J. A. Gutierrez; M. Naeve, E. Callaway, M. Bourgeois, V. Mitter and B. Heile. (2001) "IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks," *IEEE Network Magazine*, volume 15, Issue 5, September/October 2001, pp.12-19

M. J. Handy; M. Haase, D. Timmermann. (2002). "Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection", *Proceedings of IEEE*, 2002.

M. Tubaishat; S. Madria. (2003). "Sensor Networks: An Overview," *Proceedings of IEE Potentials*, April/May 2003.

Noseong Park; Daeyoung Kim, Yoonmee Doh, Sangsoo Lee, Ji-tae Kim. (2005). "An Optimal and Lightweight Routing for Minimum Energy Consumption in Wireless Sensor Networks," *Proceedings of IEEE RTSCA 2005*, August 2005.

O. Younis; S. Fahmy. (2004). "HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks," *IEEE Transactios on Mobile Computing3(4)*, pp 366–379, 2004.

P. Agarwal; C. Procopiuc. (1999). "Exact and approximation algorithms for clustering," Proceedings of 9th Annu, ACM-SIAM Symposium of Discrete Algorithms, pp. 658-667, Jan. 1999.

S. Toumpis; A.J. Goldsmith. (2003). "Capacity regions for wireless ad hoc networks", Wireless Communications, *IEEE Transactions*, Volume 2, Issue 4, Jul 2003 Page(s): 736-748

Sohrabi K.; Gao J.m Ailawadhi V., Pottie G.J. (2000). "Protocols for self-organization of a wirless sensor network," *Personal Communications IEEE*, Vol. 7 Issue 5, pp. 16-27, October 2000.

T. Murata; H.Ishibuchi. (1994) "Performance evaluation of genetic algorithms for flowshop scheduling problems", *Proceedings of 1st IEEE conference Evolutionary Computation*, vol. 2, pp. 812-817, June. 1994.

T. Rappaport; (1996). "Wireless Communiations," *Principles & Practice. Englewood Cliffs, NJ: Prentice-Hall*, 1996.

Wendi B. Heinzelman; Anantha P. Chandrakasan, Hari Balakrishnan. (2002). "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactios On Wireless Communications*, Vol. 1, No. 4, October 2002.

Wendy Rabiner Heinzelman; Anantha Chandrakasan, Hari Balakrishnan. (2000) "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proceedings of the Hawaii International Conference on System Sciences*, January 2000.

11

Optimizing Coverage in 3D Wireless Sensor Networks

Nauman Aslam
Department of Engineering Mathematics and Internetworking
Dalhousie University, Halifax, Nova Scotia
Canada, B3J-1Z1

1. Introduction

Recent advances in electronic miniaturization, software engineering and wireless communication technologies have enabled the deployment of low-power sensor nodes that are equipped with an embedded processing unit, memory, power-supply, on-board sensor, radio communication facilities (I. F. Akyildiz, W. Su et al. 2002). An important characteristic of sensor nodes is their ability to sense specific phenomena in a target field and send their data to a central node, called the Base Station/sink, possibly through multihop wireless communication links. Since most data gathering applications are concerned with collection of physical data that is generated in the target area monitored by sensor nodes, therefore coverage becomes a core meaure of performance. A fundamental issue in coverage is the quality of monitoring provided by the network. This quality is usually measured by how well deployed sensors cover a target area. In its simplest form, 1-coverage means that every point inthe target area is monitored at least one sensor. In recent years, the problem of providing sensor coverage has received extensive attention from the research community in the context of 2D sensor networks (Xing, Wang et al. 2005; Zhang and Hou 2005; Bai, Kumar et al. 2006). However, most of the real world sensor network deployments often a follow 3D model. Examples of such deployments are environmental monitoring in forests (Mainwaring, Culler et al. 2002; Szewczyk, Osterweil et al. 2004) where sensor nodes are deployed on trees of different heights in a forest, structural health monitoring of multi-storey buildings (Kim, Pakzad et al. 2006; Lynch and Loh 2006) and underwater surveillance networks (Akyildiz, Pompili et al. 2005). In most cases such deployments follow a model where sensor nodes are placed in large quantities over a target region. Excessive deployment of sensor nodes is often desirable to protect the network from individual node failures. However keeping in mind the energy and bandwidth constraints for most applications, the coverage control problem translates to choosing a set of active nodes that ensure that the target region is sufficiently monitored.

Considering the fact that sensors are deployed to interact with the physical phenomenon to gather data, coverage becomes one of the fundamental measures to gauge the service quality provided by the network to the application. Different applications may have

different requirements for coverage. Applications such as forest monitoring, or underwater sensor networks may requires every point in the deployment region to be monitored. This problem is referred to as the area coverage problem (Cardei and Wu 2006). Applications such as intrusion detection may require only coverage of specific points (hot spots) in the deployment region. Thus the solution to the coverage control problem is addressed in the context of application requirements. Another crucial aspect of WSN applications is connectivity that can be defined as the ability of sensor nodes to communicate directly or indirectly with any other active node. Typical deployments of WSNs assume sensor nodes communicate with their neighbors to forward the collected data to the sink. Without connectivity, the sensor nodes cannot forward the collected data to the base station thus hampering the quality of monitoring application.

Deployment and configuration of sensor networks to ensure the desired level of connectivity and coverage is fundamentally more challenging in 3D as compared to 2D (Poduri, Pattem et al. 2006). For the 3D case this chapter addresses the following problem:

"*Given the nodes are randomly dispersed in a target region, how to find a set of nodes such that each point in the deployment region is covered by at least one node and that the nodes are connected*".

This problem is different than finding a placement strategy in a region for full coverage, which can be solved by (Iyengar, Kar et al. 2005). It has been shown that the problem of finding a minimum set of sensors from an already deployed set is NP-hard (Yang, Dai et al. 2006). We propose an efficient algorithm that results in a connected topology in 3D while maximizing the coverage. A key feature of the algorithm is that it can be implemented in a distributed manner. Sensor nodes executing this algorithm exchange messages that are based on local information. By using the information embedded in these messages, a set of active nodes is selected such that the whole sensing region is covered. We show that the number of nodes in the active set produced by the algorithm depends on the sensing range. Considering the fact that the sensing range is an application dependent parameter, we derive a mathematical relation that is used to calculate the sensing range for the given input parameters (required coverage fraction, monitoring area and number of nodes). These calculated values provide a baseline for selecting appropriate thresholds to be used in the simulations. While the focus of this chapter remains on describing design, implementation and performance results of the proposed algorithm, we also provide insight and critical analysis of different factors effecting coverage in 3D Sensor networks. Further a detailed literature review on the related research is also provided in this chapter.

The rest of the chapter is organized as follows. Section 2 presents related work in the areas of 3D coverage schemes. Section 3 presents our system model, assumptions and preliminaries. Section 4 presents the description of our proposed distributed 3D coverage algorithm. Simulation results and analysis are presented in Section 5. Our main conclusions and directions for future research are presented in Section 6.

2. Related Work

Recently, a few researchers have investigated coverage and connectivity in 3D sensor networks. In (Poduri, Pattem et al. 2006) Poduri et al. highlight some of the challenges in designing algorithms for 3D and discussedpossible extensions of existing 2D designs for the deployment and configuration to 3D design. Research in (Alam and Haas 2006) provides a solution for the coverage and connectivity problem in a 3D underwater sensor network. The authors focused on coverage and connectivity issues of three-dimensional networks, where all the node have the same sensing range and the same transmission range. In particular, they addressed two questions. One, what is is the best way to place the nodes in three-dimension such that the number of nodes required for surveillance of a 3D space is minimized, while guaranteeing 100% coverage? Two, What should be the minimum ratio of the transmission range and the sensing range of such a placement strategy? By Using Kelvin's conjecture, they showed that the truncated octahedral tessellation of 3D space is the most plausible solution for this problem. A sphere based communication and sensing model is used to solve the node placement problem by using a truncated octahedron-based tessellation. In contrast, our work is focused on finding a solution for coverage and connectivity for a random deployment in 3D.

Andersen et. Al (Andersen and Tirthapura 2009) presesnted a scheme to optimize sensor deployemnt in presence of constraints such as senor locations and non-uniform sensing regions for the 3D WSNs. The sensor deployemnt problem orginally modeled as continous optimzation was sloved using the discrete optimization method to minimize the number of sensor deployed in the target region. The proposed technique reduces the continous optimization to a discrete optimization problem.

In another work (Cayirci, Tezcan et al. 2006) related to underwater sensor networks a distributed 3D space coverage scheme is proposed. This scheme assumes that the sensor nodes are deployed randomly and their x, and y coordinates remain fixed, however depth (z coordinate) can be manipulated. The scheme finds an appropriate depth for each sensor such that maximum coverage in 3D is maintained.

F. Chen et. al. (Chen, Jiang et al. 2008) proposed a probability based K-coverage approach for 3D WSNs. The goal is to cover the entire deployment region using at least K sensors with a certain probability 'T'. A grid distribution and a greedy heuristic are used to determine the optimal placement.

Huang et. al. (Huang, Tseng et al. 2004) investigated the coverage problem as a decision problem where the goal is to determine whether every point in the service area is covered by at least k sensors, where k is a given parameter. They proposed a polynomial time algorithm which can be executed in either a centralized or distributed manner. Each participating sensor node collects how its neighboring sensors intersect with its spherical sensing range and calculates the corresponding spherical caps which are used to determine the level of circle's coverage.

3. Network Model and Assumptions

In this section we provide description about the network model and assumptions used in our distributed coverage algorithm.

1. Communication Range: A sphere based communication ranged is assumed where each active sensor has a communication range of r_c. For reliable communication the distance between two active sensor is required to be less than or equal to r_c.
2. Sensing Region: The sphere based sensing region $\mathcal{R}_i$ of a sensor S_i located at point $\mathbb{X}$ with coordinates (x_i, y_i, z_i) is the collection of all points where a target Γ_i is reliably detected by sensor S_i.
3. Similar to (Liu and Towsley 2004), a Boolean sensing model is used. A sensor S_i is only able to detect events of interest within its sensing region $\mathcal{R}_i$. Given the sensing radius r_s from $\mathcal{R}_i$, the output of the Boolean model can be described as;

$$O[\mathbb{X}(S_i), \mathbb{X}(\Gamma_i)] = \begin{cases} \alpha & for\ \ d\big(\mathbb{X}(S_i), \mathbb{X}(\Gamma_i)\big) < r_s \\ 0 & Otherwise \end{cases} \quad (1)$$

   Where $\mathbb{X}(S_i)$ denotes the position of the sensor, $\mathbb{X}(\Gamma_i)$ denotes the location of a target and $d\big(\mathbb{X}(S_i), \mathbb{X}(\Gamma_i)\big)$ specifies the Euclidean distance between the target and the sensor. In line with the findings in (Zhang and Hou 2005), we assume that the communication range r_c is $2 * r_s$. We also assume that sensor nodes are capable of transmitting at various power level.
4. Sensor nodes are randomly dispersed over a three dimensional geographical region following a uniform distribution.
5. All sensor nodes are homogeneous in terms of energy, communication, and processing capabilities.
6. We assume that the sensor nodes are capable of switching between sleep and active modes. Most commercially available platform such as IRSI motes (MEMSIC 2011), TelosB (MEMSIC 2011), TMote Sky support features such as auto suspend, wake, and sleep mode that are used to minimize the sensor node's energy consumption.
7. All sensor nodes are location unaware i.e. they are not equipped with a GPS device.
8. The energy model presented in (Heinzelman, Chandrakasan et al. 2002) is adopted here. The amount of energy consumed for transmission E_{Tx} is of an *l*-bit message over a distance *d* is given by;

$$E_{Tx} = \begin{cases} l.E_{elect} + l.\varepsilon_{fs}.d^2 & \text{for } 0 \le d \le d_{crossover} \\ l.E_{elect} + l.\varepsilon_{mp}.d^4 & \text{for } d \ge d_{crossover} \end{cases} \quad (2)$$

Where E_{elect} is the amount of energy consumed in electronics, ε_{fs} is the energy consumed in an amplifier when transmitting at a distance shorter than $d_{crossover}$, and ε_{mp} is the amplifier energy consumed in an amplifier when transmitting at a distance greater than $d_{crossover}$.

The energy expended in receiving an *l*-bit message is given by,

$$E_{Rx} = lE_{elect} \quad (3)$$

4. Distributed Coverage Algorithm

This Section provides details of our Distributed Coverage Algorithm. The main objective of this algorithm is to select a set of sensor nodes such that each point of interest in the monitoring region is covered by at least one sensor node. Figure 1 describe the flowchart for DCA and its explanation is articulated in the following paragraph.

The algorithm consists of three main procedures. In the first procedure, when sensor nodes boot (immediately after deployment in the monitoring region) the initial network discovery process begins. The intial state of all sensor nodes in taken as '*Plain Nodes*'. At this point sensor nodes broadcast a '*Hello*' message using a tansmission radius equal to r_c. A timer '*T1*'is started locally inside each sensor node. The timer '*T1*' ensures that sensor nodes have enough time to complete the neighborhood discovery process by receiving '*Hello*' messages from other sensor nodes that are within their communication range. When timer '*T1*' expires, each node compiles a list of its one-hop neighbors. Each node then calculates a probability (referred to here as '*Active Probability*') by simply generating a random value between 0 and 1 to become an '*Active Candidate*'. In the next procedure, each node compares its '*Active Probability*' to a pre-defined value p. If the computed value of '*Active Probability*' is less than p, it changes its status to '*Active Candidate*' and broadcasts an announcement message to its neighbors within range r_s. The announcement message contains the value of its computed probability. Again the timer '*T2*' is used here to ensure that an '*Active Candidate*' is able to successfully receive announcement messages from other active candidates in its neighborhood. When the timer expires a list of active candidate messages (ACM) is build using information such as node id and '*Active Probability*'. The ACM is sorted with respect to '*Active Probability*' in decreasing order. If the entry and the head of ACM has a value lower than the node's computed probability, the sensor node changes its status to '*Final Active*' and broadcasts a notification message. Any ties are broken in favor of the sensor node with higher node id. In the final procedure, all nodes check if they received '*Final Active*' message. Any node that did not receive this message changes its status to become '*Final Active*' for the current round.

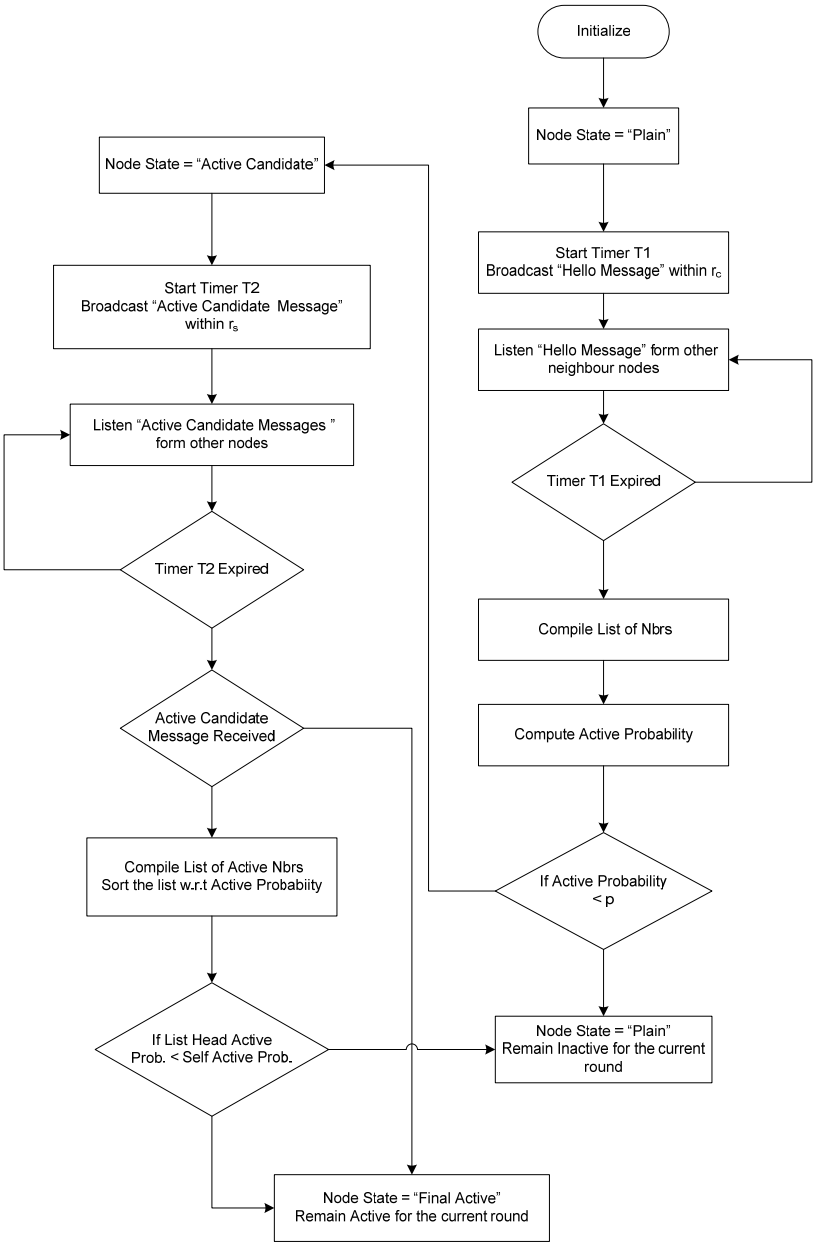


Fig. 1. Flow chart for the Distribted Coverage Algorithm

It can be noted that the sensing range plays a vital role in determining the area coverage for any given random deployemnt. In order to estimate the appropriate sensing range values for a given deployemnt region and node density we use the Poisson point process model. Let us assume that sensors are dispersed in A with intensity λ. The number of sensors located in A are given by,

$$\mathrm{N(A)} = \lambda|\mathrm{A}| \tag{4}$$

Where |A| represents the volume of three-dimensional region.
Let q be a randomly chosen point in the target region. We are interested in finding the probability that there is at least one sensor with $d(\mathbb{X}(S_i), q) \leq r_s$. Assuming a spherical sensing model, the coverage fraction η is given by the probabiliy that the point lies within at least one sensor's range:

$$\eta = \Pr[\mathrm{N(A)} \geq 1] = 1 - \Pr[\mathrm{N(A)} = 0] \tag{5}$$

The probability in (5) for a given intensity is

$$\eta = 1 - \mathrm{e}^{\frac{-\lambda.4\pi {r_s}^3}{3}} \tag{6}$$

Solving equation (6) for λ,

$$\lambda = \frac{-\log(1-\eta)}{\frac{4\pi {r_s}^3}{3}} \tag{7}$$

Using λ in equation (4) and solving for r_s,

$$r_s = \left[\frac{-\log(1-\eta)|A|}{\frac{4\pi \mathrm{N(A)}}{3}}\right]^{1/3} \tag{8}$$

The surface plot in Figure 2 describes the relationship between sensing range, coverage fraction and sensor intensity. In order to elaborate the impact of sensing range on the sensor intensity values, the plot is drawn for coverage fraction values of 0.90 to 0.999. The values sensing range takes on values between 10 am 40 m. It can be observed that the sensing range plays a significant role in determining the required coverage fraction. In order to maintain a coverage fraction of 0.99 using a sensing range of greater than 20 m the required sensor intensity is, 0.00013 $sensors/m^3$, whereas for the same coverage fraction a sensing range of 10 m results in sensor intensity of 0.0011 $sensors/m^3$.

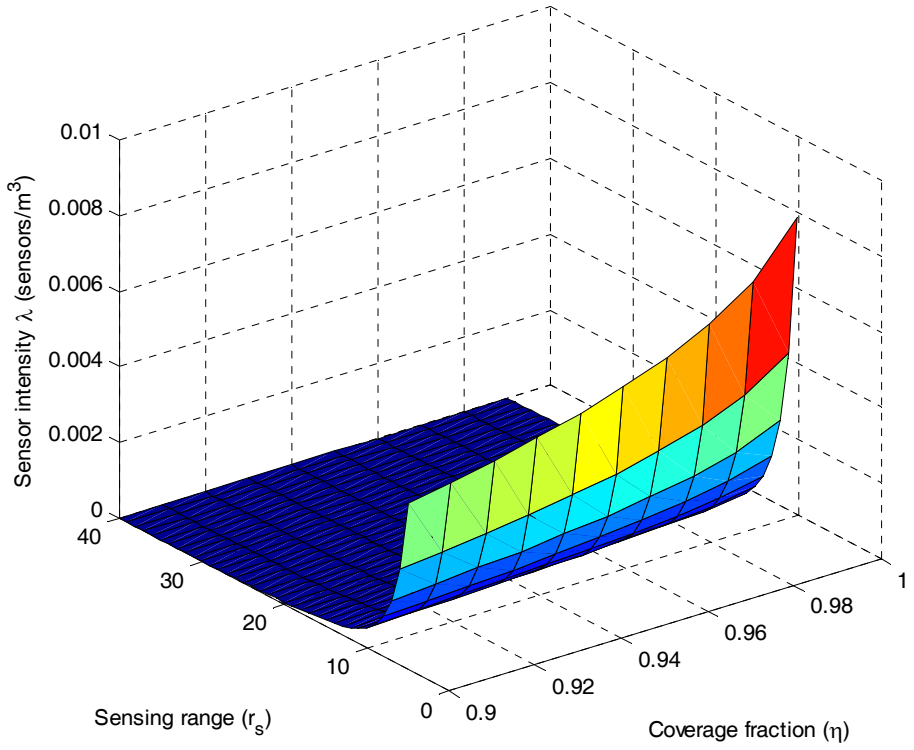


Fig. 2. Relationship between sensor intensity (λ), sensing range (r_s) and coverage fraction (η)

Figure 3 displays the results for required sensing radius vs. number of nodes and coverage fraction. The network size plotted on x-axis takes on values between 100 and 1000 nodes. Similarly, on y-axis coverage franction is plotted in the range of 0 and 1. The chosen range is necessary to demonstrate the affect of both parameters in determinig the sensing range. As an example, for a network size of 500 nodes, to guarantee a coverage fraction of 0.99 the minimum sensing radius is calculated to be approximately 13 m. Similarly, for a network size of 1000 nodes, minimum sensing radius is found to be approximately 11 m. Using topology input parameters such as deployment area information, network size and desired coverage fraction, appropriate estimates of the minimum sensing radius can be obtained. The analytical results from our model serve as a guideline to choose the optimal parameter in simulation and experimentation.

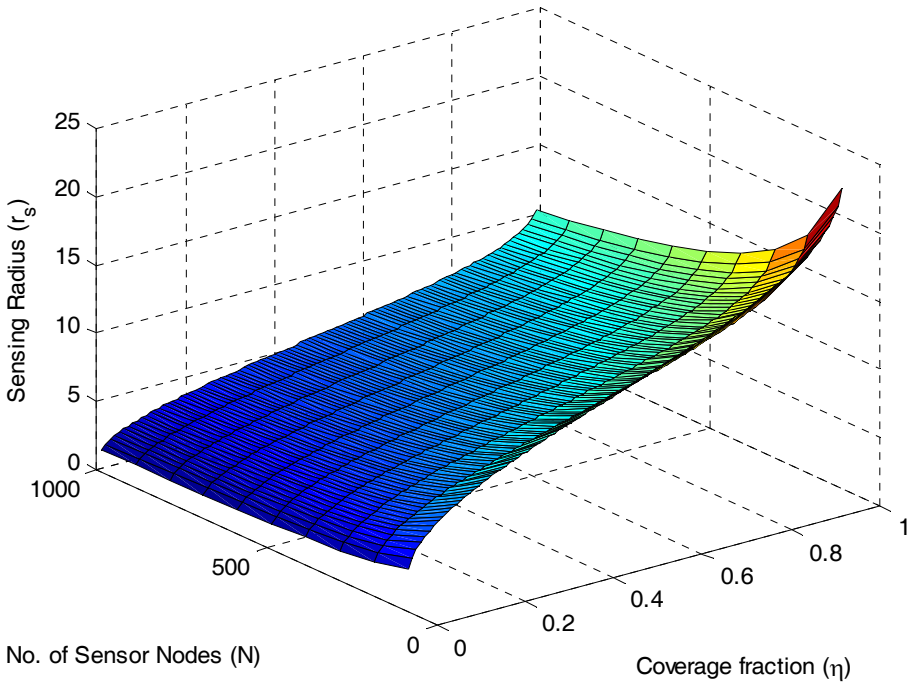


Fig. 3. Sensing range (r_s) as a function of number of nodes and coverage fraction (η)

5. Simulation Results

In this Section we present the results of a performance analysis of our DCA algorithm. The network simulation model was built using Matlab. Each simulation experiment is performed on a unique topology and consists of several rounds of network set up phase and data transmission phase. Unless otherwise stated the results presented in this section are represented as average taken over 20 independent experiments. Our aim is to find the minimum number of sensors that will achieve full coverage and connectivity in a 3D deployment region. Sensor nodes are dispersed randomly following a uniform distribution in a region of 100 x 100x 100 meters. Table 1 lists the simulation parameters used in our experiments.

We evaluated the proposed DCA algorithm with respect to the following performance metrics.

- Number of active sensor nodes: This metric provides an estimate of the solution size with respect to total number of node.
- Coverage fraction: This metric provides estimates on the percentage of points covered by k (in most cases for current work k=1) sensor nodes in the target region.
- Percentile connectivity: This metric provides a measure of the connectivity among active nodes.

Network Size	200 – 600 nodes
Area Dimensions	100 x 100 x 100 m
Sensing Range (r_s)	15 – 25 m
Communication Range (r_c)	2* (r_s)
Probabililty p	0.15
Initial Energy	0.5 J
Message Size	25 Bytes
$E_{Elect.}$ - Energy spent in electronics	50 *n J /bit*
ε_{fs} - Constant for free space propagation	10 *p J/bit/m*^{2}
ε_{mp} - Constant for multi-path propagation	.0013 *p J/bit/m*^{4}

Table 1. Simulation Parameters

Figure 4 demonstrates results from a series of experiments performed for different network sizes (200 to 600 nodes). A sensing range of 20 m was used in these experiments over 20 random topologies. Our metric of interest here is the number of nodes in the active cover set. For each network size both mean and standard deviation are reported. It can be clearly observed that significant improvements are made by reducing the number of nodes in the active cover set. For 200 nodes the cover set is 60 nodes and for 400 nodes the cover set is about 72 nodes. If the network size is increased to 600, the cover set contains about 80 nodes resulting in a saving of 86.6%. It is not surprising to notice an improvement of approximately 17 % when the network size is increased from 200 nodes to 600 nodes. The DCA algorithms ensures that there is only one active nodes within one sensing range, therefore an increase in the network size (more node density per unit area) yields a little increase in the active cover set.

The resulting topology produced by the algorithm with respect to connectivity was also evaluated. We define connectivity of a node as its ability to communicate either directly or indirectly to at least one of its neighbors. Figure 5 shows results where nodes use a sensing range that varies between 5 and 25 meters. These experiments were conducted for network sizes of 200, 300 and 400 nodes. It can be seen that a sensing range of 15m (or greater) results in a topology where 99.9% connectivity is achieved. These results corroborate perfectly with the analytical estimates discussed in the previous section.

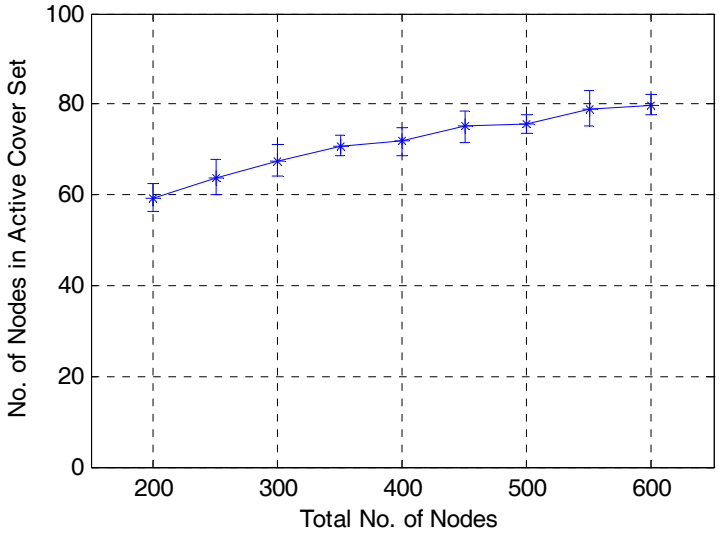


Fig. 4. Number of nodes in the active cover set for different network sizes

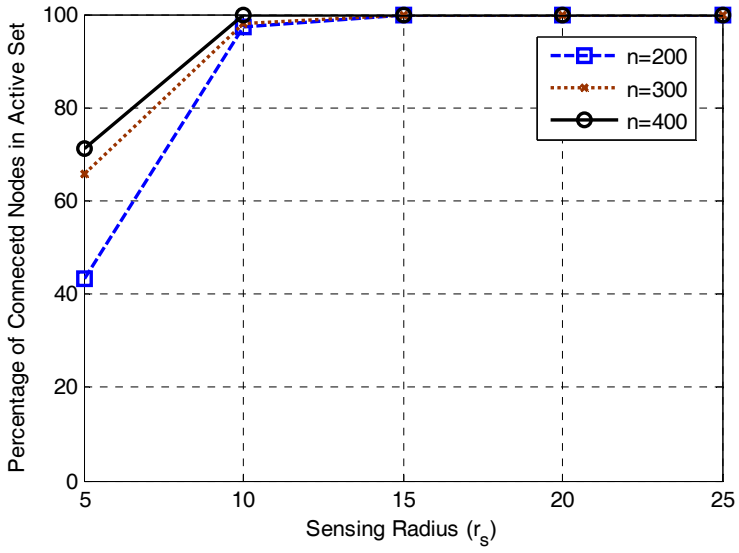
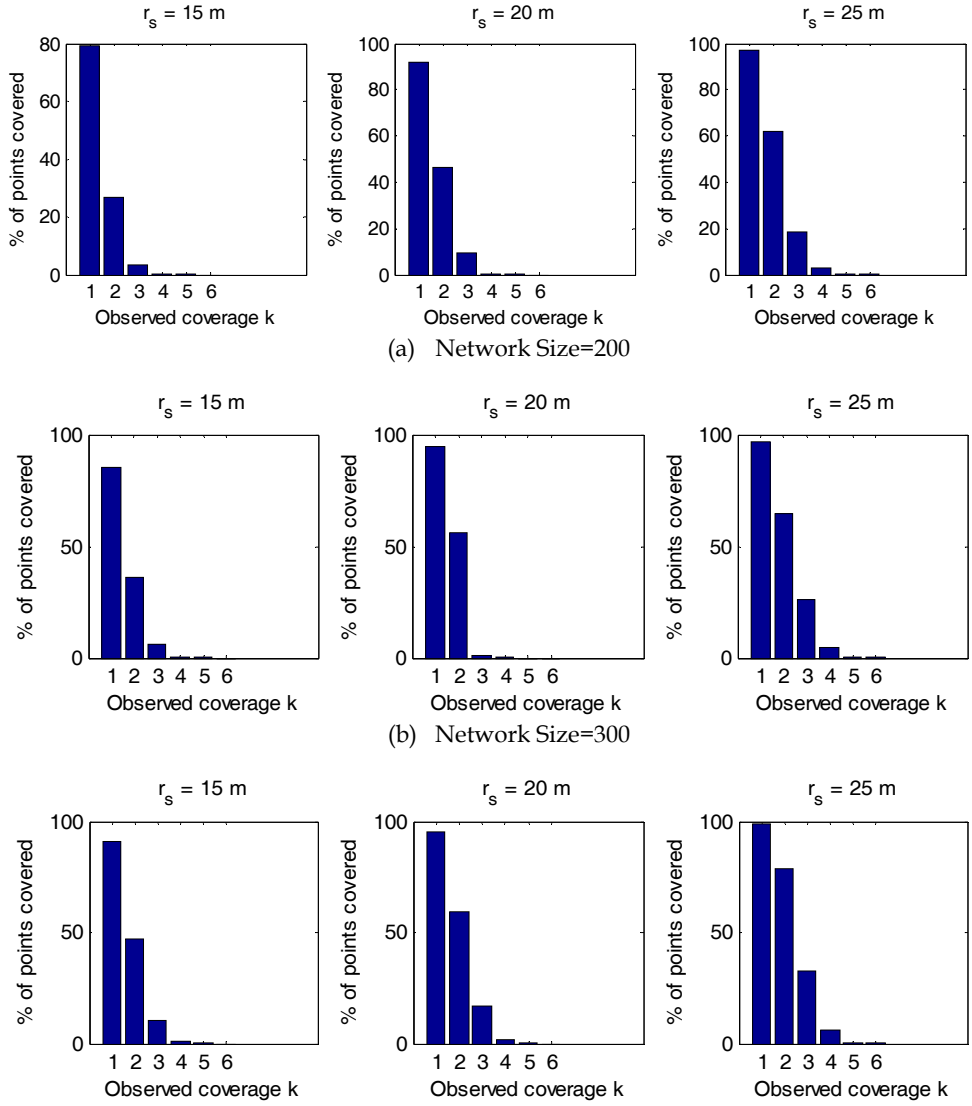


Fig.. 5. Percentage of connected nodes in the active cover set vs. sensing range (r_s)

An important evaluation criteria of coverage alogorithms is how well the target region is covered by the sensor nodes. Figure 6 presents results for the observed coverage.



(a) Network Size=200

(b) Network Size=300

(c) Network Size=400

Fig. 6. Percentage of point covered with respect to Observed coverage k in a) N=200, b)N=300 and c) N=400 nodes

As discussed in Section 4, a simple case is when a point is covered by at least one sensor, the resultant coverage is said to be of the order 1. Although the DCA is designed with the object to provide best 1-coverage (k=1) in the target region, we ran a number of experiments to estimate the coverage of higher oders i.e k > 1. For this set of experiments, three network sizes of 200, 300 and 400 nodes were selected. Simulations for each network size were

further repeated with three different values of sensing radius. The results from these experiments are presented in Figure 6. It can be observed that these results are in agreement with our analytical results presented in Section 4, we observe that for a sensing range of 25 m provides us a toplogy where 99% of nodes are covered by at least one sensor node. Moreover, the the same value of sensing range yield the topolgy where approximately 60% of the points are 2-covered (i.e k=2).

Figure 7 and Figure 8 depict the resultant topology and connectivity graph before and after the execution of DCA. It can be clearly seen that the DCA preserves connectivity while reducing extra nodes within a given deployment region.

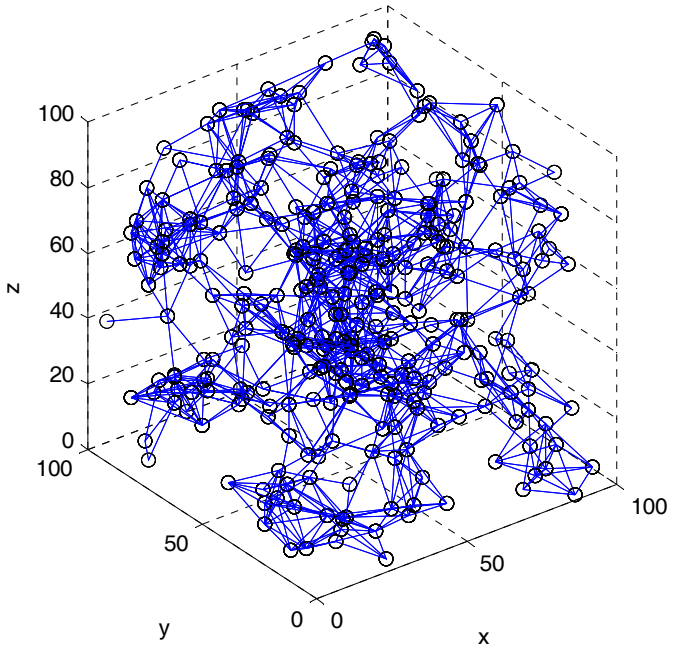


Fig. 7. Network topology and connectivity graph before the execution of DCA (network size =300 nodes, r_s=20 m)

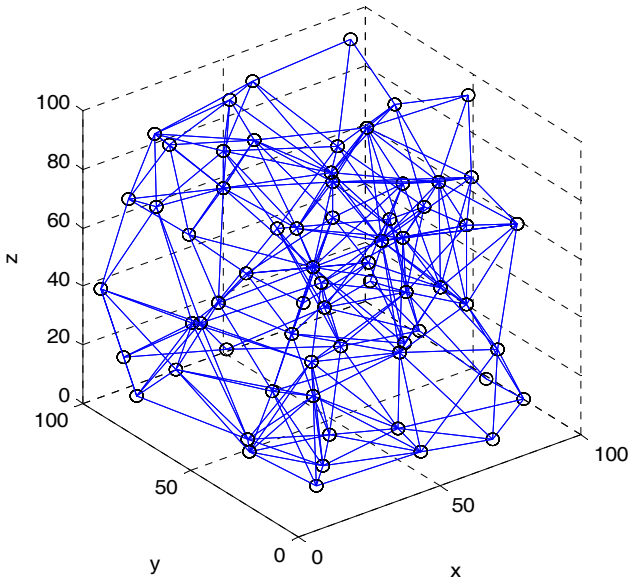


Fig. 8. Network topology and connectivity graph after the execution of DCA (network size =300 nodes, r_s=20 m)

Besides coverage and conenctivity, network lifetime is also an important performance metric for WSNs. To estimate network lifetime we used the following operation model. For each experiment nodes are deployed randomly over the target region. After the intial neighnor discovery step the operation proceeds in rounds. In each round a set of active nodes is selected according to the proposed DCA. This selection of active nodes is followed by data transmission where each active node sends 10000 bytes. Modeling the network operation in this manner allows measurement of the network life in number of rounds until the very first node runs out of its energy or a percentage of nodes completely exhaust their battery and die. The lifetime on an individual sensor node is measured in the number of rounds before its energy is depleted. The lifetime of a network can be defined in either the number of rounds until the first node dies or a certain percentage of nodes die. We ran a number of experiments to estimate network lifetime in percent of alive nodes for network sizes of 200, 300, 400 and 500 nodes. These results for metric were collected using a sensing radius of 15 m and p=0.15. While it is intutive to note that selecting a subset of active node will significantly improve over the case where all nodes remain active, the results present in Figure 9 provide insight to the perfromance of the network with different network sizes. We observe that all cases display a fairly consistent behavior with respect to the first node deatth. We also note that the rate at which node exhust their energy is also consistent. To elaborate, 50% of nodes die in round 238, 280, 336 and 390 for network size of 200, 300, 400 and 500 respectively. This gradual increase is attributed to more nodes present in the system.

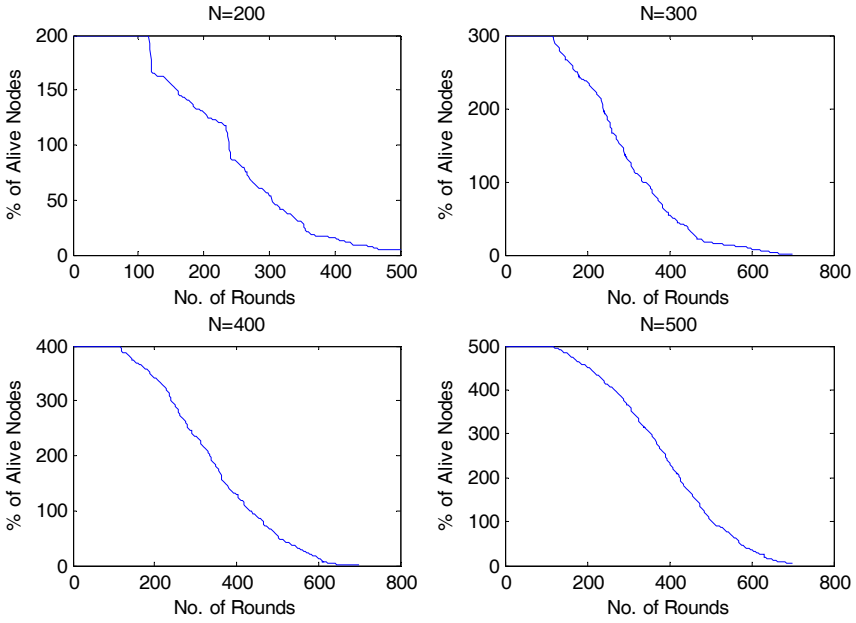


Fig. 9. Network lifetime in percentage of alive nodes for N=200, N=300, N=400 and N=500

6. Conclusions

In this work we presented a distributed algorithm for coverage and connectivity in three dimensional WSNs. The DCA algorithm presents a solution to the problem of selecting a minimum set of nodes from random deployment such that nodes remain connected while maximizing the coverage. The key feature of the algorithm is its simplicity and ability to be executed in a distributed manner. Sensor nodes executing this algorithm exchange messages with their one-hop neighbors to decide the nodes in the active cover set. We derived mathematical relations that were used to estimate the sensing range r_s, a key parameter for DCA. Simulation results provide strong evidence that for appropriate values of r_s, DCA maximizes both coverage and connectivity. Our future work will include incorporating real world deployment models and into the current framework. We plan to extend the current DCA framework to provide higher order coverage in our future work.

7. Rererences

Akyildiz, I. F., D. Pompili, et al. (2005). "Underwater acoustic sensor networks: research challenges." Ad Hoc Networks **3**(3): 257-279.

Alam, S. M. N. and Z. J. Haas (2006). Coverage and connectivity in three-dimensional networks. 12th annual international Conference on Mobile Computing and Networking Los Angles, CA, USA, ACM New York, NY, USA.

Andersen, T. and S. Tirthapura (2009). Wireless sensor deployment for 3D coverage with constraints. Sixth International Conference on Networked Sensing Systems (INSS).

Bai, X., S. Kumar, et al. (2006). Deploying wireless sensors to achieve both coverage and connectivity. ACM Mobihoc, ACM New York, NY, USA.

Cardei, M. and J. Wu (2006). "Energy-efficient coverage problems in wireless ad-hoc sensor networks." Computer communications **29**(4): 413-420.

Cayirci, E., H. Tezcan, et al. (2006). "Wireless sensor networks for underwater survelliance systems." Ad Hoc Networks **4**(4): 431-446.

Chen, F., P. Jiang, et al. (2008). "Probability-Based Coverage Algorithm for 3D Wireless Sensor Networks." Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques, Communications in Computer and Information Science **15**.

Heinzelman, W. B., A. P. Chandrakasan, et al. (2002). "An application-specific protocol architecture for wireless microsensor networks." IEEE Transactions on wireless communications **1**(4): 660-670.

Huang, C. F., Y. C. Tseng, et al. (2004). The coverage problem in three-dimensional wireless sensor networks. IEEE Global Telecommunications Conference.

I. F. Akyildiz, W. Su, et al. (2002). " A Survey on Sensor Networks." IEEE Communications Magazine **40**(8): 102-114.

Iyengar, R., K. Kar, et al. (2005). Low-coordination topologies for redundancy in sensor networks, ACM.

Kim, S., S. Pakzad, et al. (2006). "Wireless sensor networks for structural health monitoring." Proceedings of the 4th international conference on Embedded networked sensor systems: 427-428.

Liu, B. and D. Towsley (2004). A study of the coverage of large-scale sensor networks. IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)

Lynch, J. P. and K. J. Loh (2006). "A summary review of wireless sensors and sensor networks for structural health monitoring." Shock and Vibration Digest **38**(2): 91-130.

Mainwaring, A., D. Culler, et al. (2002). Wireless sensor networks for habitat monitoring, ACM.

MEMSIC. (2011). "IRIS Mote Data Sheet." from http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

MEMSIC. (2011). "TelosB Data Sheet." from http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

Poduri, S., S. Pattem, et al. (2006). Sensor network configuration and the curse of dimensionality. The Third IEEE Workshop on Embedded Networked Sensors (EmNets), Cambridge, MA, USA.

Szewczyk, R., E. Osterweil, et al. (2004). "Habitat monitoring with sensor networks." Communications of the ACM **47**(6): 34-40.

Xing, G., X. Wang, et al. (2005). "Integrated coverage and connectivity configuration for energy conservation in sensor networks." ACM Transactions on Sensor Networks (TOSN) **1**(1): 36-72.

Yang, S., F. Dai, et al. (2006). "On connected multiple point coverage in wireless sensor networks." International Journal of Wireless Information Networks **13**(4): 289-301.

Zhang, H. and J. C. Hou (2005). "Maintaining sensing coverage and connectivity in large sensor networks." Ad Hoc & Sensor Wireless Networks **1**(1-2): 89-124.

Part 3

Quality of Service Management and Time synchronization

12

Mechanism and Instance: a Research on QoS based on Negotiation and Intervention of Wireless Sensor Networks

Nan Hua[1] and Yi Guo[2]
[1]*Institute of Telecommunication Engineering, Air Force Engineering University China*
[2]*East China University of Science and Technology China*

1. Introduction

The definition of QoS (Quality of Service) varies with the concerned network techniques (wired networks, wireless access networks, wireless Ad hoc networks or wireless sensor networks, etc) and the viewpoint of observation (application level or network level) (Chen & Varshney, 2004; Crawley et al.,1998). The concerned topics of QoS in traditional networks are all end-to-end, and the bandwidth utilization is a core issue of QoS mechanism due to the requirements of multimedia applications. Although there are differences among the specific realization techniques, the research models of QoS are similar and the metrics for evaluating and describing QoS are roughly the same (Chen & Varshney, 2004).

Today, the research on the QoS of traditional networks is mature considerably in theory and practice. In wireless sensor networks (WSN), due to the features such as the limited resource (including energy, bandwidth, cache ability, storage capacity, processing capacity, transmission power, etc), high data redundancy, dynamic topology of network and specific application, the QoS problems are different from that of the traditional networks in the design and implementation. For example, in IP networks, a primary intention of QoS is to ensure that the traffic streams which have different grades or types can get corresponding and predictable transmission services. The grade of service can be classified into best-effort service, differentiated service and guaranteed service. In WSN, because of the unpredictable behavior of edge-to-edge, it is not realistic to provide predictable and reliable transmission service for traffic stream. Hence the QoS of WSN is based on unreliable and best-effort data transmission, but it does not exclude the expression method of traffic (task) stream based priority level. Moreover, WSN reduces the requirements for the packet loss rate to a certain degree; the main concerned issues are no longer the efficient utilization of bandwidth, and the QoS is not always end-to-end.

The researches on QoS mainly involve two aspects: mechanisms and metrics. The classical QoS research results of WSN were summarized by Chen and Shearifi. (Chen & Varshney,

2004; Sharifi et al., 2006). In addition, the issues about QoS of WSN are involved or taken into account in many papers in recent years, while conducting the research on the routing and clustering (topology control) protocol, MAC protocol, as well as application issues, etc (Fapojuwo & Cano-Tinoco, 2009; Hoon & Sung-Gi, 2009; Zytoune et al., 2009; Peng et al., 2008; Chen and Nasser, 2008; Yao et al., 2008; Gelenbe & Ngai, 2008; Navrati et al., 2008; Youn et al., 2007; Zhang et al., 2007; Zhang & Xiong, 2007). The QoS issues involved mainly focus on the instantaneity, fault tolerance capacity and energy consumption of networks, and are studied with the respective research fields of these papers conjointly. All these researches on QoS mentioned above belong to the research field of metrics, these researches neither focus on the QoS mechanism nor discuss the QoS issues of WSN specially and systematically from the basis and architecture. To the best of our knowledge, in the research field of QoS mechanisms of WSN, few distinctive researches are conducted at the present time. In these researches, some QoS schemes based on cross-layer QoS optimization (Cai and Yang, 2007), adaptable mobile agents (Spadoni et al., 2009), cloud model (Liang et al., 2009) and limited service polling discipline analytical model (Aalsalem et al., 2008), and so on, were presented, but are not very mature yet.

In this chapter, we focus our research domain on the mechanisms, the concrete QoS metrics is beyond our discussion scope. In this chapter, we bring forward an Active QoS Mechanism (AQM), the core of it is the negotiation between applications and network and the active intervention for them. On this basis, we conduct a further research, present and realize a common QoS infrastructure as an instance of AQM, named QISM (QoS Infrastructure base on Service and Middleware). The application, state and role oriented QoS optimization scheme, the middleware and service based architecture, the Topic and functional domain based expression method are important characteristics of QISM. Proved by simulation of a typical scenario, QISM has good QoS control ability and flexibility, can support complex applications, and is independent of network architectures.

The rest of chapter is organized as follows. In section 2, we present two QoS levels of WSN and analyze the relationship between the essential problems and QoS. In section 3, we bring forward the concept of AQM, and the working processes, the fundamental of state evaluation and strategy generation are discussed. In section 4, the design philosophy and important characteristics of QISM are studied. In section 5, the infrastructure and realization of QISM are presented and analyzed from four aspects in detail. Then, the simulation results are illustrated in section 6. Finally, we conclude this chapter in section 7.

2. Essential Problems and QoS of WSN

2.1 Three Essential Problems of WSN

We present three essential research problems which should be considered seriously in the applications of WSN through a representative application scenario:

In order to deploy WSN nodes in hostile battlefield or terrible conditions, we normally use airdrop to execute this task. After the nodes bestrewn, it is possible that quite part of them cannot work properly, which leads to heterogeneous distribution of the nodes. Furthermore, it is impossible to supply power when the node energy is exhausted. So, when the network is established, we should face three essential problems as follows:

1) Network Organization
When old nodes invalidated or new nodes joined, the network will be reorganized. Reorganization of network involves many complex processes, such as route rebuilding (the route optimization), topology reconstruction (the selection between the plane architecture and the hierarchical architecture of network, and the transformation from one to another) and task transference (new joined nodes or other working nodes resume the tasks of the disabled nodes), etc.

2) Lifetime of Network and Nodes
To prolong the lifetime of whole network, nodes should work in an energy-efficient way, which includes node dormancy and exchanges of node roles (for example, cluster head, cluster member and router node are three different roles of the nodes, which node acts as which role can be decided through elections and the role of node should alternate periodically). Through these methods, it is mostly possible to average energy consumption of the nodes and ensure the lifetime of key nodes.

3) Quality of Service
We must get tradeoff between lifetime and QoS demand of the network. For example, for the nodes in a lower-density region or executing key tasks, we should find a way to get the necessary tradeoff between application quality and node energy consumption, ensure the achievement of application and the maximum lifetime of network.

2.2 Two QoS Levels of WSN

WSN is a fully distributed network, the QoS of it can be divided into two correlative levels as follows:

1) Network (Application) QoS Level
This level focuses on the whole network, and considers quality of service with a global view of network. The concerned issues involve network organization, network lifetime, and so on. Since Application is a concept correlative with Network, the issue about the analyses of application quality and network state should also be considered in this level.

2) Node (Task) QoS Level
This level focuses on the network nodes, regulates nodes based on the analyses of metrics and data of concrete nodes under the direction of network (application) QoS level, and feeds back data to it for the problem solving of network (application) QoS level. Since Task is a concept correlative with Node, the issue about the analyses of task quality and node state should also be considered in this level.

These two levels of QoS are correlative. For example, the node energy consumption (an issue in node (task) QoS level) is closely related to the network lifetime (an issue in network (application) QoS level), while the energy saving strategy of network (an issue in network (application) QoS level) would affect the lifetime of single node (an issue in node (task) QoS level). The problems in network (application) QoS level have no way to be solved just through the data of some isolated nodes, but the acquisition and analyses of global network situation. The problems in node (task) QoS level generally are the basis of the problems solving of network (application) QoS level, but it is also independent to a certain extent.

2.3 Relationship between Essential Problems and QoS of WSN

Each essential problem of WSN described in 2.1 is not isolated, but is correlative and interact as both cause and effect. Each problem can be divided vertically into two levels: network and node, which is also correlative and affect each other. Hence, we can consider and design a mechanism that could synthetically consider the problems of network organization, lifetime and quality of service of WSN. Above all, this mechanism should associate the regulation in network level with the adjustment in node level and make them become an organic whole, which will guarantee the achievement of applications and prolong the lifetime of network furthest, meanwhile the requirement of application for network behavior is satisfied as far as possible. As discussed in 2.2, the QoS of WSN is composed of two correlative levels: network and node, so we have reason to believe that a specially designed QoS mechanism is a good way to solve the problems mentioned above.

3. Active QoS Mechanism

Generally speaking, the core of QoS mechanism in traditional networks (for example IP networks) is that how to satisfy the requirements of applications for network capability through given methods and mechanisms. The basic process of it can be described that network try its best to satisfy the requirement proposed by application; if the requirement cannot be satisfied, the network will degrade the quality of service and feeds back it to the user. We call this traditional QoS mechanism.

However, the traditional QoS mechanism will bring some problems in WSN. For example, under the circumstance of battlefield supervision application, traditional QoS mechanism will terminate the application and return errors when the object node executing key tasks or the cluster head is disabled. But actually, the application can be achieved if we reorganize network in right time and transfer the tasks in disable nodes to other normal nodes properly.

3.1 Theory of AQM

The key to solving problems mentioned above is that a feedback and negotiation mechanism must be established between the applications and network when the support of network to applications or / and the applications demand to network is / are changed. This mechanism regulates the network and applications under certain strategies dynamically, makes the applications adapt to network and network support applications furthest, and improves the support ability of WSN to applications and adaptability of applications to WSN. This feedback and negotiation mechanism between network and applications is named Active QoS Mechanism (AQM) by us.

The key of AQM is the process of active intervention for applications and network. This process is built on the analysis and evaluation for the states of applications and network, which involves two aspects: the regulation of applications to network and the reaction of network to applications. Collecting information from applications and network, and analyzing / evaluating the states of them with the information collected is the foundation of AQM.

This mechanism is not necessary in traditional networks, but it is directly related to the lifetime of applications and network in WSN. The fundamental reason of this lies in the unreliable network elements, the instability and resource-constrained nature of WSN.

3.2 Working Process

The working process of AQM involves four phases: initialization phase, surveillance phase, negotiation phase and regulation phase. The relationship of these phases is illustrated in Fig. 1. Besides, the relationship of application, network, AQM and main output in each phase are presented in Fig. 2.

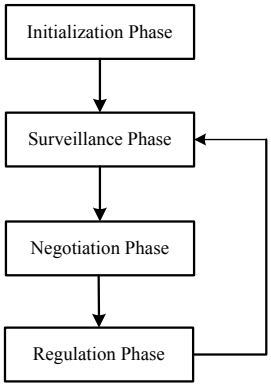


Fig. 1. Four phases in working processes of AQM

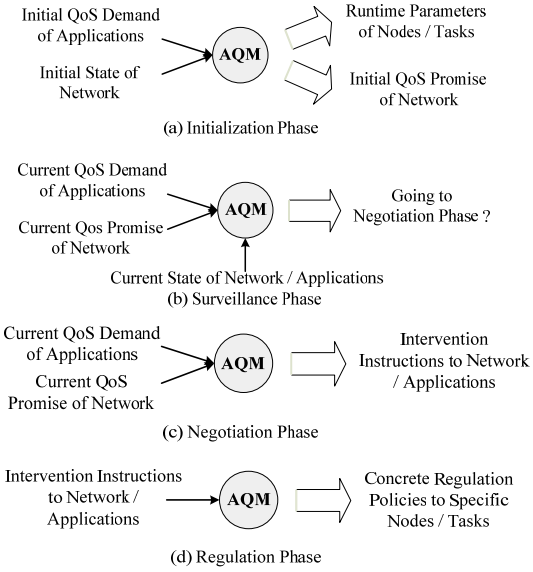


Fig. 2. Main input and output of AQM in different working processes

1) Initialization Phase
Combined with the initialization process of network, AQM generates the initial QoS promise according to the requirements of applications for QoS and the initial state of network, and sets the runtime parameters of nodes and tasks according to the initial QoS promise.

2) Surveillance Phase
AQM traces the state of applications and network constantly, and monitors the QoS demand of applications. When there is a conflict between current QoS demand of applications and current QoS promise of network, AQM goes to negotiation phase.

3) Negotiation Phase
Through AQM, a negotiation and tradeoff is achieved according to the QoS demand of applications and the QoS promise of network, and then the intervention instructions to the network and / or applications are generated. AQM goes to regulation phase.

4) Regulation Phase
According to the intervention instructions to the network and / or applications, the concrete regulation policies to specific nodes and / or tasks are generated and the runtime parameters of specific nodes and / or tasks are modified by AQM, AQM goes to surveillance phase.

3.3 State Evaluation and Strategy Generation

AQM produces the evaluation to the state of applications and network, generates regulation strategy to applications (network) and tasks (nodes). This is a process of analyzing and optimizing applications and network according to the states of them combining with the requirement of applications, this process is application, state and role oriented. We can regard state evaluation and strategy generation function of AQM as a black box, which owns a predefined method set. The input of this black box is correlative with the application demand to network, current application state, current and previous network state and current QoS promise of network. The output of it involves the intervention instructions to network and / or applications, the concrete regulation policies to specific nodes and / or tasks (in the form of runtime parameters), as shown in Fig. 3.

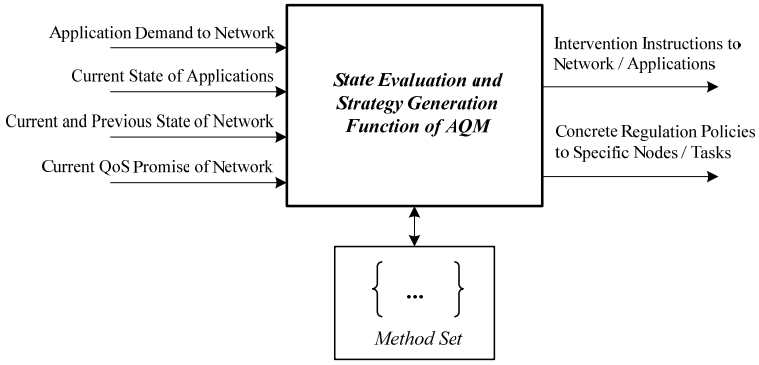


Fig. 3. Fundamental of state evaluation and strategy generation of AQM

4. QISM: an Instance of AQM

From this section, we design and realize a common QoS infrastructure as an instance of AQM, named QISM (QoS Infrastructure base on Service and Middleware) by us. The design philosophy of QISM is as follows:

4.1 Application, State and Role oriented QoS Optimization Scheme

The core of AQM is negotiation and intervention, which is based on the analyses of previous accomplishment quality of applications, current requirements of applications for the quality of service, the current and previous states of network, as well as the current service promise of network. These analyses are based on applications, states and roles. Since the application, state and role are time variant in WSN, these analyses are dynamic too.

1) Application-oriented

The main idea is to distinguish task streams, and different kind of task stream should acquire the support of different QoS in different time. This assignment of QoS should consider the previous and current states of network. Not only the distribution according to need but also the possible carrying capacity of network should be considered.

2) State-oriented

The previous and current states of network (applications) and nodes (tasks) should be considered when negotiation and intervention is proceeding; even previous data packets should be analyzed if necessary.

3) Role-oriented

The Regulations to network and nodes should consider the status and functions of nodes in current network. For example, the nodes that carry out a key sensing task should avoid becoming cluster head or router node in order to save energy and prolong its lifetime.

4.2 Middleware and Service based Architecture

Currently, there are close coupling between software and hardware, as well as applications and operating system of WSN, which has brought inconvenience for the task transference as well as the development and adjustment of hardware and software. Middleware is a software layer, which can provide services for various applications and enable different application processes to communicate via network under the circumstances of shielding difference among platforms. Through the middleware, it is convenient to provide standard system services, support and coordinate multiple runtime environments, and efficiently utilize the resource of network. The architecture of QISM based on middleware is shown in Fig.4

When an application is being performed, the application is decomposed into relatively independent tasks firstly, and then the services are abstracted from tasks. The system requests and subscribes the services, gets the required data and completes the requested functionality. Service is a concept about “set”, it is a logical abstraction of homogeneous tasks from the viewpoint of network. Service indicates “what to do” and implies the functional domains related with service. Task is concept about “individual”, including not only “what to do” but also “how to do”. For instance, for the service such as “temperature”, many nodes possibly support the task of temperature acquisition. But how to acquire, i.e.

"how to do", such as the thresholds and sampling frequency setting, is related with the tasks and nodes. Different nodes probably have different parameter values, which are decided by their runtime parameters. The relationship between services and tasks is shown in Fig. 5.

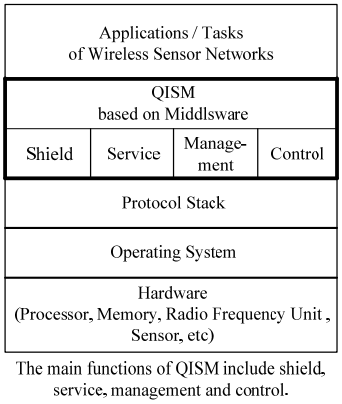


Fig. 4. Architecture of QISM based on middleware

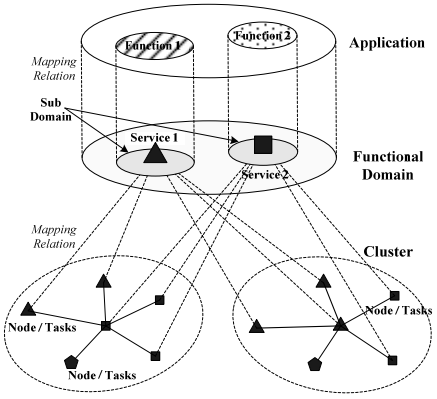


Fig. 5. Relationships among application, services, tasks and functional domain

4.3 *Topic* and Functional Domain based Expression Method

Topic is always associated with the concept application, an application can have more than one *Topic*, and a *Topic* can be associated with multiple applications. The syntax of *Topic* is defined as follows:

Topic < AppName > [< AppName > [...]] < TpStyle > < TpDesp > [< TpDesp > [...]]

where *AppName* is the name of an application and unique in the network, which is the distinction from other *Topic*s of applications. The style of *Topic* is identified by *TpStyle* and *TpDesp* is the specific description of the content of the *Topic*. *TpDesp* can be Interests and Events of WSN, or other control information related with the application, such as various

commands or messages. The control information is denoted as *SysCtrlInfo*. Different from Interest and Event, *Topic* is based on the application (network) level while Interest and Event is in the task (node) level.

Functional domain is a node set that involves all nodes which provide all kinds of services requested by a specific application, no matter whether the tasks of the nodes are working or not. The node subset that provides different services is a sub domain of the functional domain of the specific application. Functional domain is related with specific application and associated with specific Interest and Event. For example, for the application of fire monitoring, if we wants to acquire the data of temperature and smoke fume, the functional domain related with fire alarm application is the node set that involves temperature and smoke sensor nodes, the sub domain of it are the node subset that involves temperature sensor nodes and the node subset that involves smoke sensor nodes only, respectively associated with the Interest and Event of temperature and with that of smoke.

Functional domain is presented from the viewpoint of application and is unrelated with the architecture models that the network uses. In the hierarchical architecture model of network, such as cluster, a functional domain or its sub domains can cover several clusters. The relationships among application, services, tasks, and functional domain are shown in Fig. 5.

5. Infrastructure and Realization of QISM

5.1 Architecture and Function

According to the discussion in 4.2, QISM is base on middleware and is a software layer that located between the protocol stack and applications, communicating with application / task and protocol stack through standard API. QISM is composed of six modules: application analysis, application / task regulation and control, strategy generation / analysis, state analysis, service management, *Topic* generation / resolving. The hierarchical relationship of the above-mentioned modules is shown in Fig. 6. Each module lies in sink and (or) sensor node, as shown in Table 1.

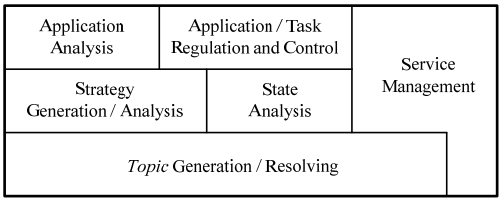


Fig. 6. Hierarchical architecture of QISM

Module Name		Location	Function
Application Analysis		Sink	Decomposing application into tasks according to the description of application, and determining whether the tasks are supported by existing available services through Service Management Module. If necessary, indexing and subscribing related services through Service Management Module.
Application / Task Regulation and Control	Application Regulation and Control	Sink	Analyzing implementation status depending on functional domain states and services states, evaluating whether or not the network supports application, and completing application regulation and control.
	Task Regulation and Control	Sensor Node	Completing task regulation and control through setting runtime parameters of task.
Strategy Generation / Analysis	Strategy Generation	Sink	Generating runtime parameters of tasks according to application requirements as well as current application and node state in the states library.
	Strategy Analysis	Sensor Node	Resolving runtime parameters, determining whether current node is in specific functional domain.
State Analysis		Sink	Analyzing task implementation status, determining functional domain and service state, evaluating network state, maintaining the states library.
Service Management		Sink, Sensor Node	Realizing service publication and subscription mechanism, and functions of service discovery, indexing and maintenance.
Topic Generation / Resolving		Sink, Sensor Node	Packing and unpacking *Topic*.

Table 1. Main modules and functions of QISM

5.2 Service Management

The functions of service management of QISM, which consist of publication, subscription, inquiry, index and maintenance of services, are implemented through Service Management Module. The service publication and subscription mechanism is the basis of QISM and the main usage mode of service, where the task side (sensor node) publishing services initiatively and the application side (sink) subscribing and using them. Furthermore, the service inquiry and index mechanism provides the methods that can acquire the state of service, and the methods of requesting and activating service from the application side. The function of service maintenance is used in recording and maintaining the services which are published in the network already, and the function is realized in sink and sensor nodes locally. In sink, table *TASvc* and *TOSvc* have the records of current available services and subscribed services respectively; in sensor node, the subscribers of node services are recorded in table *TSvcOd*. Subscription, inquiry and index function are implemented in the sink, publication function is done in sensor nodes, maintenance function both in the sink and sensor nodes.

The processes of service publishing, subscribing, inquiring and indexing in QISM are illustrated as Fig. 7.

1) Publication and subscription of service

Publication and subscription of service involve two kinds of messages: *MsgSvc* and *MsgSvcOd,* their syntaxes are defined as follows:

MsgSvc < SvcName > < SvcPrvdID > [< SvcDesp >]

MsgSvcOd <AppName> < SinkID > < SvcName > [< SvcPrvdID > < SvcDesp >]

where *SvcName* is the name of service; *SvcPrvdID* and *SinkID* are the IDs of the service provider and the sink respectively, which can be addresses, domains or coordinates and so on; *SvcDesp* is the description of the service.

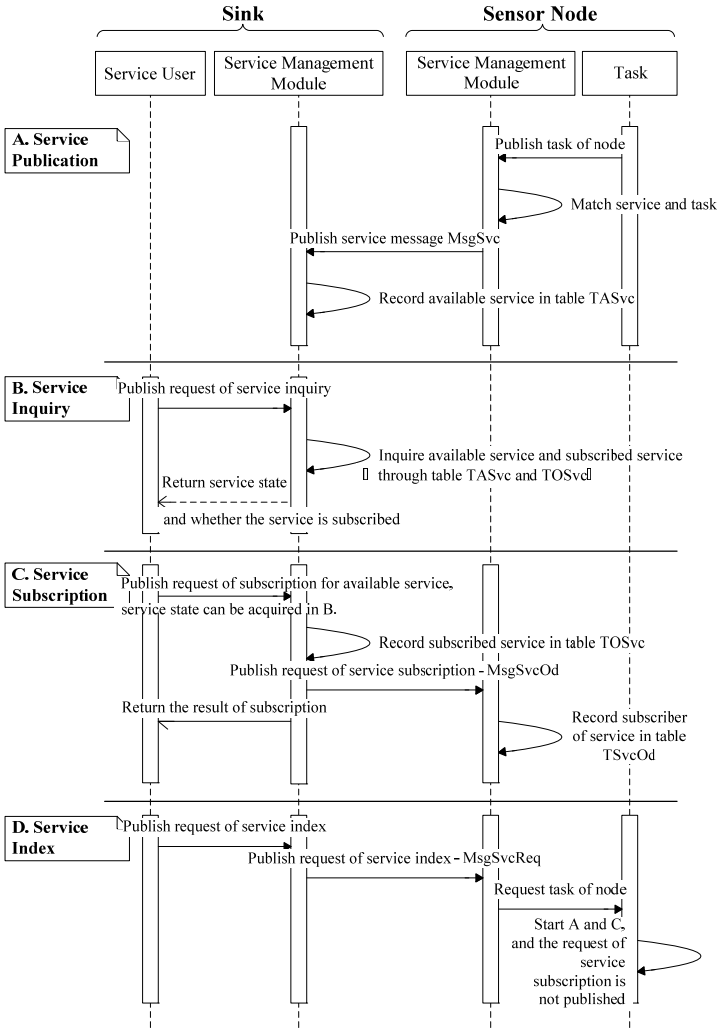


Fig. 7. Processes of service publishing, subscribing, inquiring and indexing in QISM

After the network deployed, sensor node will publish and broadcast the tasks (which can be performed by it) through *MsgSvc* in the form of service; after received by sink, the services are saved in *TASvc* and determined whether to be subscribed according to the requirements of application. If the service is useful, the sink sends message *MsgSvcOd* to *SvcPrvdID* to subscribe it, and records the subscribed service in *TOSvc*. After the sensor node receives *MsgSvOd* which is sent to it, it records the subscriber in *TSvcOd*. Based on the consideration of resource saving and network survivability, sensor node dose not record *MsgSvc*s that are sent by other nodes.

If *SvcPrvdId* is specified in *MsgSvcOd*, which means the sink subscribes the service that is provided by specific sensor node; otherwise, which means the sink subscribes all the same services that are provided by all nodes in the network. When sending service data, sensor node will specify the data receiver. In the case of multiple sinks, the sink that did not subscribe the service, will discard service data directly after the service data is received.

2) Inquiry and index of service

The state of service is either *Available* or *Unavailable*; the state of specific service can be acquired through inquiring *TASvc* in sink. If a service is available, it can be used through subscribing. Otherwise, it means that the service has not been published by any nodes yet. In this case, if we want to use the service, we should start the service index mechanism in sink. The sink sends message *MsgSvcReq* to the network firstly, then the sensor nodes that are capable of providing the service publish the service, finally the sink subscribes the service and uses it. The syntax of *MsgSvcReq* is defined as follows, where *SvcReg* stands for the region where the service is located.

MsgSvcReq < SvcName > < SinkID > [< SvcReg > < SvcDesp >]

3) Maintenance of service

The service maintenance functions of QISM mainly include the table maintenance and update of *TASvc*, *TOSvc* and *TSvcOd*, as well as service cancelling and unsubscribing. When sensor node is unable to provide services, such as under the circumstances that sensor is damaged, *MsgSvcFail* is broadcasted and *TSvcOd* is cleared by the sensor node. After the sink receives *MsgSvcFail*, *TASvc* and *TOSvc* (if the service is subscribed already) are updated in order to cancel the service. The syntax of *MsgSvcFail* is defined as follows:

MsgSvcFail < SvcName > < SvcPrvdID > [< SvcDesp >]

When the application no longer needs a specific service, the sink sends message *MsgSvcCancel*, and deletes the corresponding service from *TOSvc*. The sensor node that provides the service maintains a user counter, and when it receives *MsgSvcCancel*, the corresponding counter of the service is decreased by one and *TSvcOd* is updated at the same time. When the counter is reduced to 0, the sensor node broadcasts *MsgSvcFail*. The syntax of *MsgSvcCancel* is defined as follows:

MsgSvcCancel < SvcName > < SinkID > [< SvcPrvdID > < SvcDesp >]

It should be noted that the service publication only means that sensor node has the ability of carrying out a task, but when to start or to terminate the task, as well as how to implement the task depends on the runtime parameters. More specifically, under the control of the application, task-performing is achieved through the built-in mechanism of QISM by

correlative modules generating, sending and implementing the runtime parameters, and it is unrelated with service management module. Moreover, the runtime parameters of tasks are not saved in service management module. Besides, the above-mentioned messages related with service, are sent directly through network protocol stack by service management module.

5.3 Basic Working Process

From the viewpoint of the operator of QISM, QISM includes two basic working processes: dynamic adjustment of application and active regulation of task, as shown in Fig. 8. Both are associated closely and reciprocal causation, as a unified organic whole.

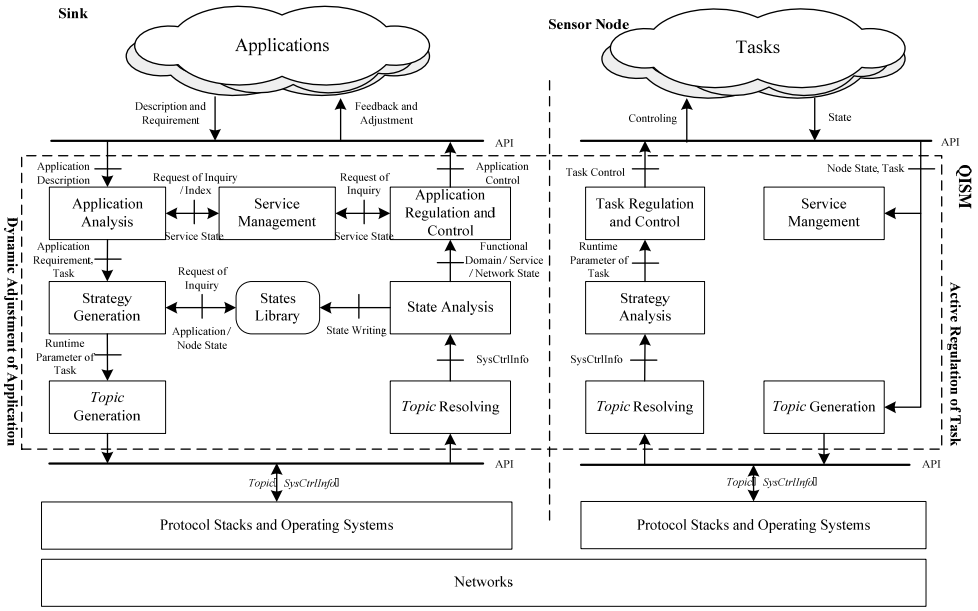


Fig. 8. Data stream of QISM

QISM first completes the service subscription process according to the description and requirement of the application, and then generates the runtime parameters. Afterwards, QISM publishes the runtime parameters of the tasks, and starts the processes of regulations of application (network) and task (node). In sensor node side, QISM intervenes the execution of tasks by setting runtime parameters of tasks, and feeds back the states of nodes and tasks to sink; QISM regulates the application after state analysis process, and then generates the new requirements and (or) descriptions of the application. Such a repetition will form a closed loop until the ends of tasks.

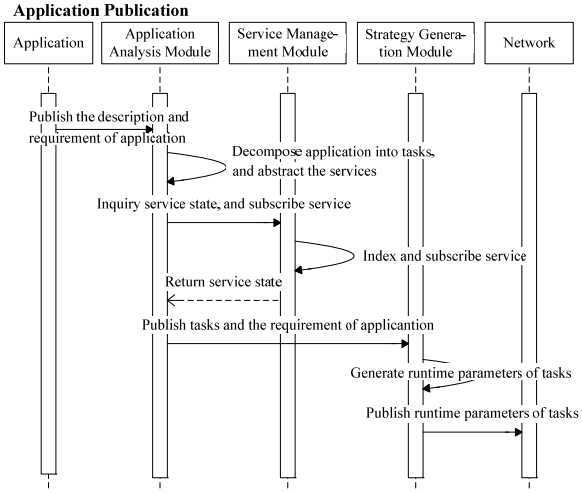
It should be noted that, we do not reflect the processing methods and flow direction of the Interest and Event in Fig. 8 and in the following discussion. In fact, since Interest and Event is a kind of organization and representation method of data, the requirements and descriptions of application may contain the content of Interest, and the states that fed back to QISM from tasks may include a part of data of Event. Transmission of Interest and Event

can be implemented by *Topic* mechanism or other methods. A detailed discussion of Interest and Event is beyond the scope of this chapter.

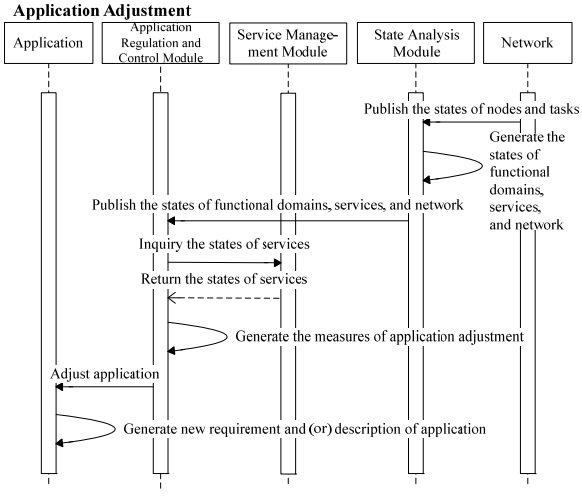
1) Dynamic adjustment of application

Dynamic adjustment of application, whose operator is sink, consists of two processes: application publication and application adjustment, as shown in Fig. 9(a) and (b).

- *Application publication (the downlink process from application to network)*



(a) Application publication



(b) Application adjustment

Fig. 9. Basic working process of QISM - Dynamic adjustment of application

Firstly, QISM decomposes application into several independent tasks according to the description of the application, and abstracts the service corresponding to the tasks. For example, for the fire monitoring application, temperature monitoring and smoke monitoring are two tasks that need to be accomplished; in the node level, the services that are provided by the nodes with the ability of sensing temperature and sensing smoke are temperature sensor service and smoke sensor service respectively. The division, abstraction and correspondence of task and service, is based on the pre-defined rules, which are fixed when the network is deployed.

Secondly, QISM subscribes services. If the services are available, they can be used after subscription; if not available, they can be activated by service index mechanism and then be subscribed. Eventually, all the services required by the application should be available; otherwise, QISM will terminate the application and cancel all the tasks.

Thirdly, QISM generates the runtime parameters of the tasks according to the request of application. The runtime parameters, including functional domain, sampling frequency, thresholds and so on, have great influence on the service quality and execution manner of tasks. In addition, energy strategy is also an essential parameter. The death of some important nodes whose functions are irreplaceable, such as the cluster headers in hierarchical structure, the key routing nodes in multi-hop routing, the key sensor nodes, and so on, may cause the failure of the application or the collapse of the network. So the energy strategy should be established in order to prolong the lifetime of nodes.

Finally, QISM publishes the runtime parameters of tasks to the network in terms of *Topic* (*SysCtrlInfo*), for sensor node receiving and performing.

- *Application adjustment (the uplink process from network to application)*

The *Topic* (*SysCtrlInfo*) received by sink from network includes the current state information of tasks and nodes; its specific content is determined by the pre-defined rules and is different with different tasks. The above-mentioned state information is the basis of application adjustment.

Firstly, QISM confirms that *SysCtrlInfo* is for this application (sink) through resolving the domain of *Topic AppName*, for there are multiple applications (multiple sinks) in the network probably.

Secondly, the state information of a single node is transformed into measurable QoS metrics, and on this basis, the state of functional domains and that of services are generated and the network state is evaluated. The related QoS metrics consist of network delay, packet loss rate, data reliability of node, node lifetime, node energy consumption per bit, packet transmission delay of node, invalid packet rate of node and node remnant energy, etc.

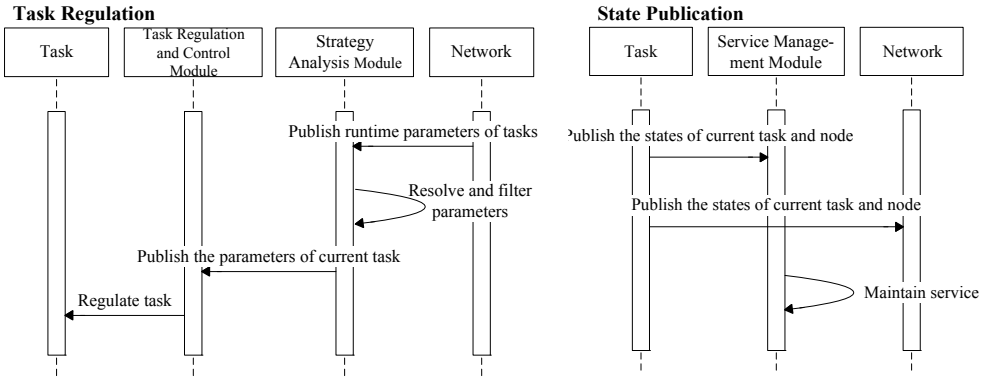
Finally, QISM generates adjustment measures (i.e. intervention instructions to network / applications) for application and informs application to perform, based on the state analysis results, current states of functional domain / service / network and current requirements of application. Application adjustment is faced to functional domain, network and service, not single node and its tasks, though its basis is the information collection and analysis of single node and its tasks. The measures of application adjustment include resuming application, pausing application, resuming application after adjustment, ceasing application, etc.

2) Active regulation of task

Active regulation of task, whose operator are sensor nodes, consists of two processes: task regulation and state publication, as shown in Fig. 10(a) and (b).

- *Task regulation (the uplink process from network to task)*

In sensor node side, *Topic* (*SysCtrlInfo*) received from network consists of the requirements of application for task in the form of runtime parameters of task (i.e. regulation policies to specific nodes / tasks) sent from sink. First of all, QISM confirms that *SysCtrlInfo* is for the functional domain where current node is located through resolving the domain of *Topic AppName*. And then, QISM completes task regulation by setting runtime parameters of the task.



(a) Task regulation (b) State publication

Fig. 10. Basic working process of QISM - Active regulation of task

- *State publication (the downlink process from task to network)*

During the implementation of task, sensor node needs to inform QISM of the current task state (such as whether the task is completed or not, the implementation progress of task) and node state (such as working state of sensor, remnant energy of node). On the one hand, QISM adjusts current services of node according to this, e.g. service is canceled when sensor node is disabled; on the other hand, QISM sends the states to related sink through network for the preparation of state evaluation.

5.4 Task (Node) Refactoring

Through the generation of concrete regulation policies to specific nodes and tasks based on the intervention instructions to applications and network, QISM realizes the task and node refactoring by means of resetting the runtime parameters of specific tasks and nodes. The so-called refactoring means that the functions and performance of tasks and nodes are modified through the reset of runtime parameters of them, which leads to the change of the support ability of network to applications and the QoS demand of applications to network.

The more ideal methods for the implementation of task (node) refactoring involve three schemes as follows, but the concrete implementation method in QISM should be studied more deeply in our further research:

1) Self-adaptive Adjustment of Protocol Architecture
The protocol stack involves several components (protocol elements) which are served for different purposes or applications and have different performances and functional characteristics. When external conditions are changed, the QISM selects and applies proper the protocol element automatically.

2) Software Component Technology
Component is a kind of reusable software element which can be used to construct other software. Software component technology is an object-oriented technical system, which builds applications through the combination of different components and involves a series of correlative operations and services. The core of it is the concept of PnP (Plug and Play) soft component that can work immediately after it is embedded.

3) Downloading and Updating of Protocol and Application
QISM downloads new protocols and updating programs dynamically and on demand from the base station (for example the sink). This method is more flexible but need the coordination with the base station or service center.

6. Simulation and Analysis

QISM has a complex active regulation process for application and task, and its specific logics, including application analysis, application / task regulation and control, strategy generation / analysis, state analysis and service management, etc, depend on specific application and specific realization of system. So we only prove the feasibility of QISM through the simulation for fire monitoring application below.

In fire monitoring application, the network consists of temperature sensor nodes and smoke sensor nodes, crossly deployed in the adjacent regions A and B, as shown in Fig. 11. After the network is deployed, system performs the tasks of temperature and smoke sensing on the support of QISM.

We used ns2 v2.27 to simulate the above scenarios with Linux Red Hat 9. Thirty-six static nodes deployed uniformly in a grid-like plane scene, the temperature sensor nodes and smoke sensor nodes were crossly deployed. The clustering algorithm was DSCO (Hua & Shi, 2007) and cluster head did not alternate. The protocol of MAC layer was 802.11b, Interface Queue (IFQ) length was 50, and Two-ray Ground Reflection was as wireless transmission model. To be brief and without loss of generality, the single-hop communication was adopted between the cluster head and sink.

After cluster organization is completed, the simulation uses the following logic to control and regulate the application and network:

Logic 1: Service publication. Node publishes temperature and smoke service to sink through cluster head.

Logic 2: Application publication, service decomposition and service subscription. Application (sink) subscribes the temperature service *Svc_Tmp* and smoke service *Svc_Fg* of nodes in region A through QISM.

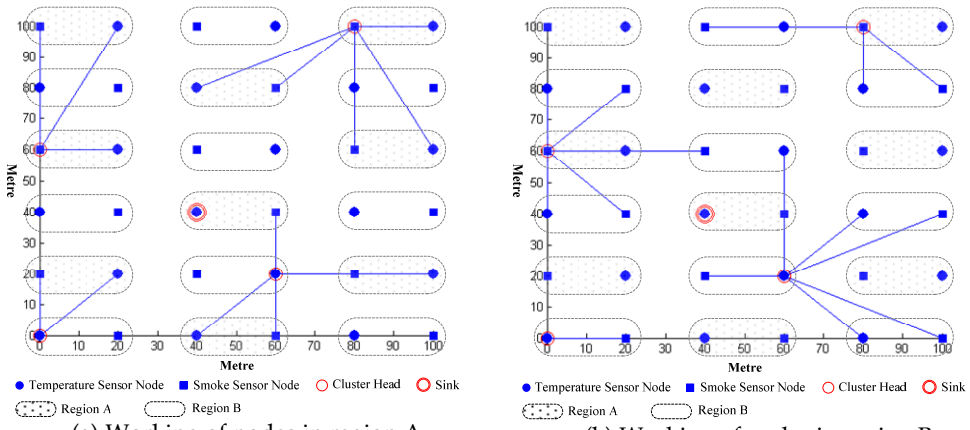
Logic 3: Task runtime parameters generation and task control. The nodes in region A are activated by QISM through dispatching the task runtime parameters (such as sampling frequency f_s) to them, as shown in Fig. 11(a).

Logic 4: Node state and service state publication. Nodes in region A report current node states (such as remnant energy E_r) to QISM meanwhile they feed back the sensing data (such as temperature and smoke concentration) to application through sink.

Logic 5: State analysis of task and node, application active regulation, task regulation and control. QISM ceases the data acquisition task in region A according to pre-defined logics when the energy of 50% nodes decrease to $E_r/3$, and subscribes services *Svc_Tmp* and *Svc_Fg* of region B. The nodes in region B are activated and replace the work of nodes in region A, as shown in Fig. 11(b). Then logic 1-4 are repeated, where nodes in region A is replaced by nodes in region B.

An important reason for designing logic 5 is to prove that active regulation of QISM for application and service can effectively prolong the lifetime of network and application. The results of simulation shows, in the above simple working model based on energy, the lifetime of cluster members are longer than that of members which do not use QISM (all deployed nodes working synchronously) by 30%. The longer lifetime of node is, the longer lifetime of network and application is.

It should be noted that in the above-mentioned simulation, we have not considered the lifetime of cluster head. Energy consumption of cluster heads can be averaged to prolong its lifetime through dynamic alternating cluster head in cluster organization algorithm (Hua & Shi, 2007). The study on dynamic cluster organization is beyond the scope of this chapter.



(a) Working of nodes in region A (b) Working of nodes in region B

Fig. 11. Simulation results of QISM

We can get the following conclusions through above simulation:

1) Simulation process covers the main work processes of QISM, and the mechanism of QISM is feasible.

2) The illustration of main functions of QISM in the simulation, including switch of node working state of region A and region B, node working parameters (runtime parameters) setting, feedback and analysis of node state, modifying application logic, etc, have proved that the flexibility and ability of QISM in QoS control aspect. Complex application can be supported by more complex control logic.

3) In the simulation, the nodes were organized as cluster, and the nodes in the same region (region A or region B) spread in different clusters, which proved QISM is unrelated with network architecture and two kinds of network architecture plane and hierarchy are all supported.

4) The lifetime of network and application can be prolonged through reasonable dynamic regulation for the application and tasks, for example, nodes in region A and those in region B alternated working under specific energy strategy.

7. Conclusions

Although the research on the QoS of traditional networks (such as IP networks) is mature considerably, but due to the features of WSN such as the limited resource, high data redundancy, dynamic topology of network and specific application, and so on, the research on QoS of it is different from the traditional networks in design and implementation. In this chapter, we focus our research on the QoS mechanism of WSN, and bring forward an Active QoS Mechanism (AQM), the core of which is the negotiation between applications and network and the active intervention for them. On this basis, we conduct a further research, present and realize a common QoS infrastructure as an instance of AQM, named QISM (QoS Infrastructure base on Service and Middleware). The application, state and role oriented QoS optimization scheme, the middleware and service based architecture, the *Topic* and functional domain based expression method are important characteristics of it. Proved by simulation of a typical scenario, QISM has good QoS control ability and flexibility, can support complex applications, and is independent of network architectures.

In further research, we will focus on the "full" realiazation of the mechanism proposed by us, but many theoretical and technical difficulties should be solved firstly. For example, the negotiation between applications / network, and the active intervention for them is the core of AQM, the concept of "cognition" can be very helpful for them. But how to achieve "cognition" is a more challenging work.

8. References

Aalsalem, M. Y.; Iftikhar, M.; Taheri, J. & Zomaya, A. Y. (2008). On the provisioning of guaranteed QoS in wireless sensor networks through limited service polling models, *Proceedings of the 5th IFIP International Conference on Wireless and Optical Communications Networks 2008 (WOCN '08),* pp. 1-7, 5-7 May 2008.

Cai, Wen-Yu & Yang, Hai-Bo. Cross-layer QoS optimization design for wireless sensor networks, *Proceedings of IET Conference on Wireless, Mobile and Sensor Networks 2007 (CCWMSN07)*, pp.249-252, 12-14 Dec. 2007.

Chen, D. & Varshney, P. K. (2004). QoS support in wireless sensor networks: a survey, *Proceedings of International Conference on Wireless Networks (ICWN)*, 2004, Las Vegas.

Chen, Yunfeng & Nasser, N. (2008). Enabling QoS multipath routing protocol for wireless sensor networks, *Proceedings of IEEE International Conference on Communications 2008 (ICC '08)*, pp. 2421-2425, 19-23 May, 2008.

Crawley, E. et al. (1998). A framework for QoS-based routing in the internet, RFC 2386, http://www.ietf.org/rfc/rfc.2386.txt.

Fapojuwo, A. O. & Cano-Tinoco, A. (2009). Energy consumption and message delay analysis of QoS enhanced base station controlled dynamic clustering protocol for wireless sensor networks, *IEEE Transactions on Wireless Communications,* Vol. 8, No. 10, pp. 5366-5374.

Gelenbe, E. & Ngai, E. C.-H. (2008). Adaptive QoS routing for significant events in wireless sensor networks, *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems 2008 (MASS 2008)*, pp. 410-415, Sept. 29 - Oct. 2 2008.

Hoon, Kim & Sung-Gi, Min (2009). Priority-based QoS MAC protocol for wireless sensor networks, *Proceedings of IEEE International Symposium on Parallel and Distributed Processing 2009 (IPDPS 2009)*, pp. 1-8, 23-29 May, 2009.

Hua, Nan & Shi, HaoShan (2007). DSCO: a simple distributed cluster organization algorithm of wireless sensor networks, Chinese Journal of Sensors and Actuators, vol. 20, No. 6, June, 2007, pp. 1397-1403.

Liang, Jun-bin; Chen, Ning-jiang & Yu, Min-min (2009). A cloud model based multi-dimension QoS evaluation mechanism for wireless sensor networks, *Proceedings of the 4th International Conference on Computer Science & Education 2009 (ICCSE '09)*, pp. 348-352, 25-28 July 2009.

Navrati, Saxena; Abhishek, Roy & Jitae, Shin (2008). Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks. *Computer Networks,* Vol. 52, No. 13, 17 September 2008, pp. 2532-2542.

Peng, Shanghong; Yang, S. X.; Gregori, S. & Tian, Fengchun (2008). An adaptive QoS and energy-aware routing algorithm for wireless sensor networks, *Proceedings of International Conference on Information and Automation 2008 (ICIA 2008)*, pp. 578-583, 20-23 June, 2008.

Sharifi, M.; Taleghan, M. A. & Taherkordi, A. (2006). A middleware layer mechanism for QoS support in wireless sensor networks, *Proceedings of International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, pp. 118-118, 2006.

Spadoni, I. M. B.; Araujo, R. B. & Marcondes, C. (2009). Improving QoS in wireless sensor networks through adaptable mobile agents, *Proceedings of IEEE INFOCOM Workshops 2009,* pp. 1-2, 19-25 April 2009.

Yao, Lan; Wen, Wenjing & Gao, Fuxiang (2008). A real-time and energy aware QoS routing protocol for multimedia wireless sensor networks, *Proceedings of the 7th World Congress on Intelligent Control and Automation 2008 (WCICA 2008)*, pp. 3321-3326, 25-27 June, 2008.

Youn, MyungJune; Oh, Young-Yul; Lee, Jaiyong & Kim, Yeonsoo (2007). IEEE 802.15.4 based QoS support slotted CSMA/CA MAC for wireless sensor networks, *Proceedings of International Conference on Sensor Technologies and Applications 2007 (SensorComm 2007)*, pp. 113-117 ,14-20 Oct. 2007.

Zhang, Xuemin & Xiong, Zenggang (2007). Research on pertinence of QoS metrics based on IEEE 802.15.4 in wireless sensor networks, *Proceedings of the third International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2007 (IIHMSP 2007)*, pp. 663-666, Vol. 2, 26-28 Nov. 2007.

Zhang, Ye; Chen, He & Jiang, Lingge (2007). Energy and QoS trade-off analysis of S-MAC protocol in wireless sensor networks, *Proceedings of IET Conference on Wireless, Mobile and Sensor Networks 2007 (CCWMSN07)*, pp. 76-79, 12-14 Dec. 2007.

Zytoune, O.; Fakhri, Y. & Aboutajdine, D. (2009). An energy aware QoS routing protocol for wireless sensors network, *Proceedings of International Conference on Multimedia Computing and Systems 2009 (ICMCS '09)*, pp. 245-248, 2-4 April, 2009.

13

A Reliable and Flexible Transmission Method in Wireless Sensor Networks

Dae-Young Kim and Jinsung Cho
Kyung Hee University
S. Korea

1. Introduction

Recent advances in wireless communication have enabled multifunctional tiny nodes to construct a wireless network by themselves Akyildiz et al. (2002). The network is called a wireless sensor network. The tiny sensor nodes are densely deployed in a physical space. They monitor physical phenomena, deliver information, and cooperate with neighbor nodes Akyildiz et al. (2002); Culler et al. (2004); Hac (2003); Zhao and Guibas (2004); Chong and Kumar (2003). The communication systems in end-to-end data transmission of wireless sensor networks employ a recovery mechanism for lost data during data transmissions because reliable data transmissions are required for various sensor network applications.
Two types of retransmission have been proposed for the recovery, namely end-to-end loss recovery (E2E) and hop-by-hop loss recovery (HBH). In these mechanisms, lost packets are retransmitted from a source node or an intermediate node. If a retransmit request for lost packets is sent to a source node, the end-to-end delay may increase because channel error accumulates exponentially over multi-hops Wan et al. (2002). The well-known HBH mechanisms are PSFQ Wan et al. (2002) and RMST Stann & Heidemann (2003). PSFQ is based on ACK message and RMST is on NACK message. In HBH, when intermediate nodes cache data packets into storage, retransmissions can be requested to an intermediate relay node to reduce end-to-end delays. Because sensor nodes have limited resources, however, it is difficult for all sensor nodes to find sufficient space in their routing paths to cache data packets. There is therefore a tradeoff between end-to-end delays and memory requirements.
Because data traffic on sensor networks requires a variety of levels of communication reliability (CR) depending on the application, a loss recovery method to guarantee the desired CR should be provided. Traditional loss recovery mechanisms consider only 100% reliability. In this letter, we propose a flexible loss recovery mechanism to guarantee various CRs and we discuss the tradeoff between end-to-end delays and memory requirements for various CRs. The proposed method can be widely used for the design of wireless sensor networks that require a variety of CRs.

2. A Reliable and Flexible Transmission Method in Wireless Sensor Networks: Active Caching

As mentioned previously, E2E involves large end-to-end delays for 100% reliability because of high packet loss during multi-hop transmissions. To guarantee high reliability and minimal

$RELIABLE-TRANSMIT(CR, i, p_i, P_{tx}(i-1), F(i-1))$

1. $P_{tx}[i] \leftarrow P_{tx}[i-1] \cdot (1-p_i)$
2. if $P_{tx}[i] > CR$
3. then $F[i] \leftarrow false$
4. else $F[i] \leftarrow true$
5. $P_{tx}[i] \leftarrow (1-p_i)$
6. cache data packets to a node n_i

Fig. 1. Active caching algorithm at i-th node, n_i.

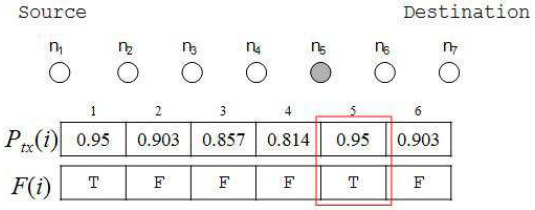


Fig. 2. An example of active caching.

end-to-end delays, HBH caches data in every node over a routing path resulting in large memory requirements. When only some nodes cache data on a routing path, there exists a tradeoff between the end-to-end delays and the memory requirements. For applications which do not require 100% reliability, every node needs not cache data via HBH. When a target CR is given, we need a flexible method to guarantee the given CR while minimizing the memory requirement. In this section, we present such a method - active caching (AC).

The proposed scheme allows various CRs of application services. It determines positions where data caching occurs using a dynamic programming algorithm, which solves every subproblem just once and then saves its answer in a table to avoid the work of recomputing the answer Cormen et al. (2001). If there are holes in sequence numbers of received data, a caching node recognizes packet loss Karl & Willig (2005). The caching node sends a NACK message to a previous caching node along the path and the previous caching node retransmits lost packets selectively.

First, we define the problem and subproblems for the active caching as a dynamic programming algorithm to guarantee an end-to-end reliable data transmission as:

Problem: $P_{tx}(H) > CR$.
Subproblem: $P_{tx}(h) > CR$, where $h = 1, 2, \cdots, H$.

The packet delivery rate $P_{tx}(H)$ during total hop counts H should be greater than the desired communication reliability CR. To do that, the packet delivery rate $P_{tx}(h)$ during hop counts h in each hop should be greater than the CR. The key idea for solving the problem is to cache data packets if the probability of packet transmission does not satisfy the desired communication reliability. By solving the subproblems, we can solve the entire problem.

Figure 1 shows the proposed active caching algorithm for loss recovery. Each node solves the subproblem using the tables for the packet delivery rate $P_{tx}(i)$ until i-th hop and the caching flag of i-th node $F(i)$. Both $P_{tx}(i-1)$ and $F(i-1)$ of the tables are piggybacked in data packets and they are delivered to the next node. In a source node ($i = 1$), $P_{tx}(1)$ is $1 - p_1$ as the packet delivery rate at the 1st hop and $F(1)$ is *true*. *Line 1-3:* n_i calculates $P_{tx}(i)$ using $P_{tx}(i-1)$, where $P_{tx}(i)$ accumulates the packet delivery rate $1 - p_i$ of i-th hop while packets are transmitted. After that, it compares $P_{tx}(i)$ with CR. If $P_{tx}(i)$ satisfies the desired CR, n_i is not a caching node ($F(i)$ is $false$). *Line 4-6:* If $P_{tx}(i)$ does not guarantee the desired CR, n_i becomes a caching node ($F(i)$ is *true*). In this case, $P_{tx}(i)$ compensates for its packet delivery rate as the reliability instead of accumulating $P_{tx}(i)$ and data packets are cached onto n_i's buffer. Each node runs the algorithm of Figure 1 and the total active caching over a routing path is performed by the dynamic programming algorithm. Figure 2 shows an example of the active caching when seven sensor nodes are deployed sequentially and they have an average 5% packet loss rate and 80% CR. Every node satisfies 80% CR and data caching occurs at n_5. When packet loss happens between a source node n_1 and the caching node n_5, the caching node requests retransmission to the source node. When packet loss happens between the caching node and a destination node n_7, the destination node requests retransmission to the caching node.

3. Analysis

A packet loss rate occurs due to wireless link and contention errors. Since all the packets are destined to the sink node in wireless sensor networks, the contention error in links close to the sink node may increase. To model the packet loss rate at i-th hop, we assume the uniform link error p_l and the contention error which is proportional to the square of transmission hop counts.

$$p_i = p_l + \alpha i^2, \tag{1}$$

where α is the contention failure factor. Then the packet delivery rate during h hops from the s-th node is

$$P_{tx}(s,h) = \prod_{i=s}^{s+h-1} (1 - p_i). \tag{2}$$

Data caching occurs when $P_{tx}(s,h)$ is lower than CR. When the number of nodes N over a route and CR are given, the hop counts h from a caching node s and the number of caching nodes N_c are obtained by the function in Figure 3. Φ represents a set of (s,h) tuples and the (s,h) tuples are used to compute the retransmission counts of lost packets. For example in Figure 2, $\Phi = \{(1,4),(5,2)\}$.

$$\Phi = \{(s_j, h_j) \mid j = 1, \cdots, N_C\}. \tag{3}$$

If the retransmission counts for h hops from a caching node s is given by $\psi(s,h)$, the total retransmission counts $E[C]$ between a source node and a sink node are represented by the sum of $\psi(s,h)$ as

$$E[C] = \sum_{j=1}^{N_c} \psi(s_j, h_j). \tag{4}$$

Because the retransmitted packets can also experience transmission failure, we should consider repeated retransmissions for $\psi(s,h)$. Let $\Gamma_f(j,s,h)$ indicate the number of transmitted packets at the j-th retransmission. Then $\psi(s,h)$ can be represented as

```
CalcHopCounts(N, CR)
1.  n ← 1, s ← 1, h ← 1, N_c ← 0
2.  Φ = φ
3.  loop: n < N
4.      if P_tx(s,h) > CR
5.          then n ← n + 1, h ← h + 1        //no caching
6.          else h ← h − 1                   //caching
7.              if (h = 0)
8.                  then h ← 1, n ← n + 1
9.              add (s,h) to Φ, N_c ← N_c + 1
10.             s ← n, h ← 1
11. end loop
12. if (h > 1)
13.     then add (s, h − 1) to Φ, N_c ← N_c + 1
```

Fig. 3. Function to obtain (s,h) tuples.

$$\psi(s,h) = \sum_{j=1}^{\infty} \big(h \cdot \Gamma_f(j,s,h) \cdot P_{tx}(s,h)\big). \tag{5}$$

If we let $\Gamma_s(k,s,h)$ be the number of successfully transmitted packets among k packets during h hops from node s, $\Gamma_f(j,s,h)$ can be represented recursively as

$$\Gamma_f(j,s,h) = \Gamma_f(j-1,s,h) - \big[\Gamma_s\big(\Gamma_f(j-1,s,h),s,h\big)\big]^1, \tag{6}$$

where $\Gamma_f(0,s,h) = K$ and K is the number of total packets which is generated in a source node.
The number of successfully transmitted packets $\Gamma_s(k,s,h)$ can be calculated by the probability of successful transmission of Bernoulli trials $P_s(k,m,s,h)$ as

$$\Gamma_s(k,s,h) = \sum_{m=1}^{k} m \cdot P_s(k,m,s,h). \tag{7}$$

If m data packets are transmitted successfully among k packets to deliver across h hops from a caching node s, the probability of successful transmissions can be obtained by Bernoulli trials as

$$P_s(k,m,s,h) = \binom{k}{m} \cdot P_{tx}(s,h)^m \cdot \big(1 - P_{tx}(s,h)\big)^{k-m}. \tag{8}$$

The memory requirement B is defined as the caching rates of intermediate nodes including a source node. It is computed by N_c and the number of relay nodes over a routing path:

$$E[B] = \frac{N_c}{N-1}. \tag{9}$$

[1] $[x]$ is n, in case of $n - 0.5 \le x < n + 0.5$

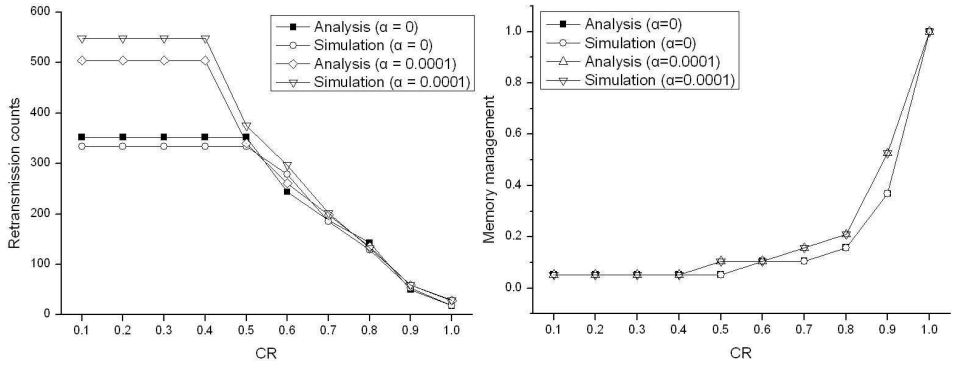


Fig. 4. Validation of our analysis (p=0.03).

A high $E[C]$ indicates large end-to-end transmission delays and $E[B]$ represents the memory requirements of buffers on the data transmission routes. Because both $E[C]$ and $E[B]$ can be estimated by CR of traffic through Eq.(4) and Eq.(9), a flexible data transmission system can be designed.

4. Evaluation

In this section, we validate the analysis through simulations and compare the performance of active caching (AC) with that of E2E and HBH. For the simulation, we assume 20 sensor nodes are deployed sequentially and the wireless channel has both link and contention error as described in Section 3. The contention failure factor α is determined as 0.0001 by considering total hop counts. So, p_i in Eq.(1) ranges from 0.03 to 0.07 when p is 0.03 in our experiments. The sensor nodes employ AODV as a routing protocol. Assuming a packet is 30 bytes and the data rate is 250kbps, we perform the analysis and simulation by varying CR from 10% to 100%. AC with CR from 0.1 to 1 is expressed as AC0.1 to AC1.

Figure 4 shows the results of the analysis and the simulation of the retransmission counts and the memory requirements when a source transmits 40 packets. The results of the analysis and the simulation show an average of 94% similarity. Figure 4 also represents the tradeoff as mentioned earlier. The high CR requires a high memory requirement for reliability and it decreases the retransmission counts. When the memory requirement is the lowest, the retransmission counts are the highest and AC runs as E2E. In short, we can design wireless sensor networks that take the desired CR and memory requirements into consideration through the proposed active caching.

Figure 5 shows the performance comparison of E2E, HBH, and AC. Because AC with the highest memory requirement caches data to every intermediate node, it operates as HBH. When AC does not perform data caching, it operates as E2E. That is, AC switches between HBH and E2E while showing the performance tradeoff between them. In addition, it has a tolerable end-to-end delay to minimize the memory requirement depending on CR. In Figure 5, the end-to-end delays of E2E increase when the wireless channel has a high link error rate. However, the end-to-end delay of AC maintains similar values because AC increases the memory requirements to ensure CR. An evaluation has been performed for 10 and 50 nodes

deployed over a route, and the results are similar to the case of 20 nodes. These results have been omitted due to the page limitation.
Figure 6 shows the ratio of caching nodes over relay nodes. Because the contention error increases when the density of nodes increases, the ratio of caching nodes increases when the number of sensor nodes increases.

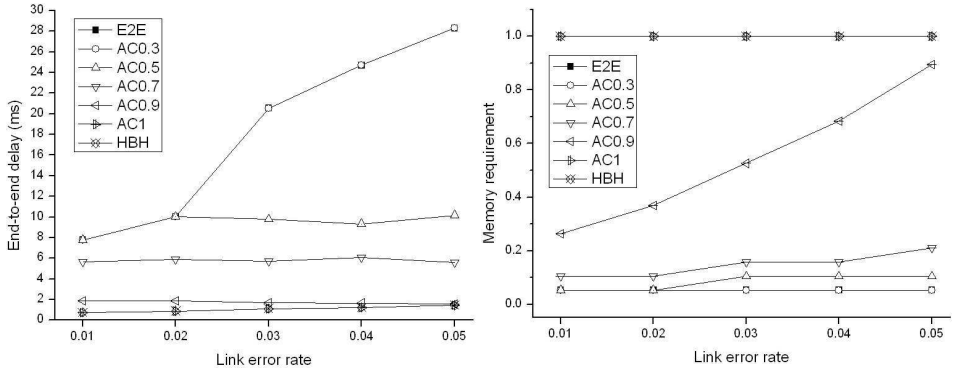


Fig. 5. Performance comparison of E2E, HBH, and AC.

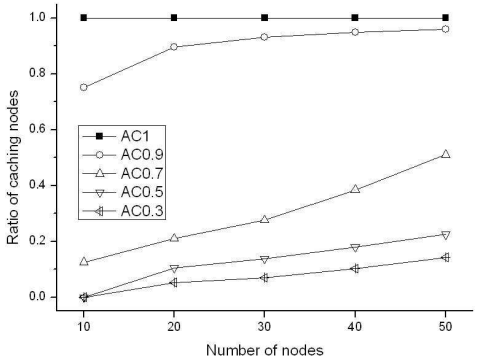


Fig. 6. The ratio of caching nodes.

5. Conclusion

Wireless sensor networks transmit data through multiple hops. End-to-end data transmission must recover lost data for reliable data transmissions. Active caching (AC) provides more flexible end-to-end delays and memory requirements for a given reliability than the existing recovery mechanisms (i.e., E2E, HBH). By using the proposed dynamic loss recovery with active caching, a flexible end-to-end data transmission system can be designed.

6. Acknowledgement

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-(C1090-1021-0003))

7. References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine*, Vol. 40(No. 8): pp. 102–114, August 2002.

Culler, D., Estrin, D., and Srivastava, M. (2004). Guest editors' introduction: Overview of sensor networks. *IEEE Computer*, Vol. 37(No. 8): pp. 41–49, August 2004.

Hac, A. (2003). *Wireless sensor network designs*, John Wiley & Sons, 2003.

Zhao, F. and Guibas, L. (2004). *Wireless sensor networks: An information processing approach*, Morgan Kaufmann Publishers, 2004.

Chong, C. -Y. and Kumar, S. (2003). Sensor networks: Evolution, opprtunities, and challenges, *Proceedings of the IEEE*, Vol. 91(No. 8): pp. 1247-1256, August 2003.

Wan, C. Y., Campbell, A. T., and Krishnamurthy, L. (2002). PSFQ: A reliable transport protocol for wireless sensor networks, *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1-11, September 2002.

Stann, F. and Heidemann, J. (2003). RMST: Reliable data transport in sensor networks, *Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102-112, May 2003.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*, Vol. 1, The MIT Press, 2001.

Karl, H. and Willig, A. (2005). *Protocols and architectures for wireless sensor networks*, John Wiley & Sons, 2005.

Performance Analysis of Binary Sensor-Based Cooperative Diversity Using Limited Feedback

Ali EKŞİM[1] and Mehmet E. ÇELEBİ[2]
Tubitak-BILGEM[1], Istanbul Technical University[2]
Turkey[1,2]

1. Introduction

The most important advantage of wireless sensor networks (WSNs) is their ability to bridge the gap between the physical and logical worlds by gathering certain useful information from the physical world and communicating that information to more powerful logical devices that can process it. If the ability of the WSN is suitably harnessed, it is envisioned that WSNs can reduce or eliminate the need for human involvement in information gathering in certain civilian and military applications (He et al., 2004).

It is a common belief that in the near future, many WSNs will be deployed for a wide variety of applications including monitoring and surveillance. Each sensor is powered by battery and is supposed to work for a relatively long time after deployment. The total energy cost of WSN includes all aspects of the sensor's actions. Transmission energy efficiency and reliability becomes important because wireless transceivers usually consume a major portion of battery energy (Akyildiz et al., 2002). This is true considering the severe channel fading and node failure in hostile environment (Ng et al., 2005).

Transmission energy conservation in WSN has two aspects. First, transmission protocols and algorithms should have high energy efficiency. Space-time coding and processing are helpful for enhancing transmission energy efficiency and reliability (Li & Wu, 2003). In particular, space-time block codes (STBCs) have attracted great attention because of their affordable linear complexity (Alamouti, 1998; Tarokh et al., 1999). Among the numerous STBC schemes, Alamouti's STBC (Alamouti, 1998) is probably the most famous one due to its simplicity. However, space-time techniques are traditionally based on multiple transmit antennas.

Due to insufficient antenna space, cost and hardware limitations, wireless sensors may not be able to support multiple transmit antennas. For the wireless sensors which have no multiple transmit antennas, STBC may still be used with cooperative transmission schemes (Li, 2005; Sendonaris, 2003a; Sendonaris, 2003b; Laneman & Wornell, 2003; Ohtsuki, 2006) where multiple sensors work cooperatively to form a virtual antenna array. Additional performance improvement can be achieved if limited feedback is available at the cooperating sensors. Two techniques are generally used for limited feedback; Sensor (relay) selection (SS) which selects n_1 out of n active sensor for cooperation ($n_1 \leq n$) and Extended Cooperative Balanced Space-Time Block Coding (ECBSTBC) which uses all active sensors (Eksim & Celebi, 2009a; Eksim & Celebi, 2010a).

Another important aspect of transmission energy conservation is that energy consumption rates in different parts of the WSN should be uniform or almost uniform so that the wireless sensors have approximately same lifetime. If the energy consumption rates are non-uniform, some parts of the WSN may die much sooner than the others. If these dying parts are critical for the WSN, this situation may lead to early dysfunction of the network, thus loosing Quality of Service (QoS), even if the other parts of the network still have a lot of residual energy. In the literature, this is called energy hole (Li & Mohapatra, 2007) problem.
Although SS schemes prolong the network life in uniform wireless channels, due to nature of the non-uniform wireless channels or location of the sensors, some of the sensors are more frequently selected for cooperation, so, there may be little or no energy left for their own use. Then, the energy hole problem occurs. For this problem not occurring in non-uniform wireless channels, the ideal communication protocol should distribute communication energy among the active sensors evenly without losing the QoS of the communication.
In (Ohtsuki, 2006), the performance of the statistical STBC cooperative diversity with observation noise and quantization noise is analyzed. In this work, the Alamouti`s code is used which is the only orthogonal code which achieves full diversity and full rate for two sensors, and the achievable diversity order is two when a single receive antenna is present at the fusion center. The use of the Alamouti`s code improves the bit error performance of the system when more than two active sensors are present in the transmitting side. The achievable diversity order can be increased via limited feedback. Since the limited feedback is not used in (Ohtsuki, 2006), the issue of how much feedback from a fusion center improves the performance when quantization and observation noise are present, is not analyzed. Additionally, the performance of binary sensors in non-uniform wireless channels and the impact of the energy hole problem in non-uniform wireless channels are not well investigated in the literature.
In this chapter, we show how to improve the performance of the statistical STBC with limited feedback. The effect of quantization and observation noise is also included in the analysis. Moreover, we show that SS schemes cause an energy hole problem in non-uniform wireless channels. The ECBSTBC provides an improvement to this problem since this scheme utilizes all available sensors to maintain equal power consumption among the available sensors and meets QoS of the communication until the end of the network lifetime. This increases the energy efficiency of the communication protocol in non-uniform wireless channels.
In addition, not only the ECBSTBC but also the SS schemes are adversely affected by the observation noise since it limits the bit error rate (BER) performance (Eksim & Celebi, 2010a). To improve upon this problem, we propose an ECBSTBC combined with SS scheme (Eksim, 2010b). In this scheme, an active sensor does not cooperate with other active sensors to transmit the observations if its observation is classified as "noisy". On the other hand, the sensors cooperate with each other using the ECBSTBC when their observation noise level is smaller than predefined threshold for transmission toward the fusion center. This hybrid technique yields improved performance at the fusion center compared to solely using the ECBSTBC or the SS methods.
In the following section, the system model is described, in the third section, the Extended Cooperative Balanced Space-Time Block Codes (ECBSTBCs) are explained, in the fourth

section, a performance analysis presented, and in the last section, the results of the our work and the conclusion are given.
The following notation used in this chapter: * denotes the conjugate operation; Re{.} and Im{.} are the real and imaginary part of the argument, respectively. The operator $\lceil . \rceil$ rounds to the smallest integer greater or equal than its argument.

2. System Model

The wireless sensor network consists of one source, one fusion center and N sensors which are located randomly and independently. Figure 1-2 show the wireless sensor network and its analytical model, respectively. All sensors are equipped with a single antenna and cannot communicate with each other. All channels are assumed frequency flat Rayleigh fading channel where channel gains are circularly complex Gaussian random variables and statistically independent from each other. The channels are quasi-static, namely, the fading coefficients remain constant over the duration of one frame and change independently in the following frame. h_{rid} is the channel gain from the ith active sensor to the fusion center where i=1, 2,.., n.
The fusion center is assumed to have perfect knowledge of the sensor-fusion center channels. This can be achieved via pilot tone training. However, the fusion center has no knowledge of the accuracy of the sensor measurements, since knowledge of the measurements at the fusion center requires considerable protocol overhead. Because of energy efficiency, only n sensors are active. Active sensors observe the environment. Due to the presence of the noise, the observation at each active sensor may be different. The observed data are binary quantized and transmitted by BPSK.

2.1 Battery model

The Battery Model simulates the capacity and the lifetime of the sole energy source of the sensor. In reality, the battery behavior highly depends on the constituent materials and modeling this behavior is a difficult task. Present network simulation tools use linear model (Park et al., 2001). In the linear model, the battery behaves as a linear storage of current. The maximum capacity of the battery is achieved regardless of what the discharge rate is. The simple battery model allows user to see the efficiency of the user's application by providing how much capacity is consumed by the user. Knowing the current discharge of the battery and the total capacity in Ah (Ampere×Hour), one can compute the theoretical lifetime of the battery using the equation, $t = C_{bat}/I$, where t is the battery lifetime, C_{bat} is the rated maximum battery capacity in Ah, and I is the discharge current.
In this model, sensor user having an initial amount of energy diminishes its value when a packet is sent or received. In limited battery simulations, battery counter is added (Lim et al., 2005; Buttyan & Hubaux, 2003). It represents the battery power which is left to the sensors. When a sensor`s battery is consumed, further cooperation requests will not be accepted. In addition, many short range wireless networks generally consume the available energy for receiving which is approximately 2/3rd of the energy for transmitting (Lal et al., 2005).

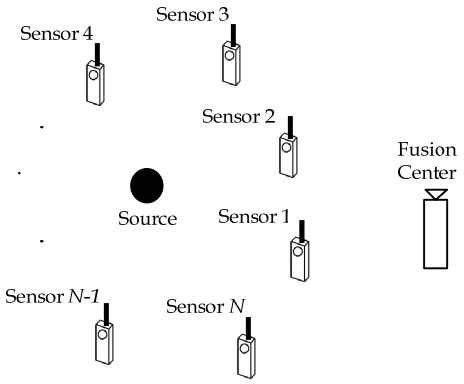


Fig. 1. Wireless sensor network

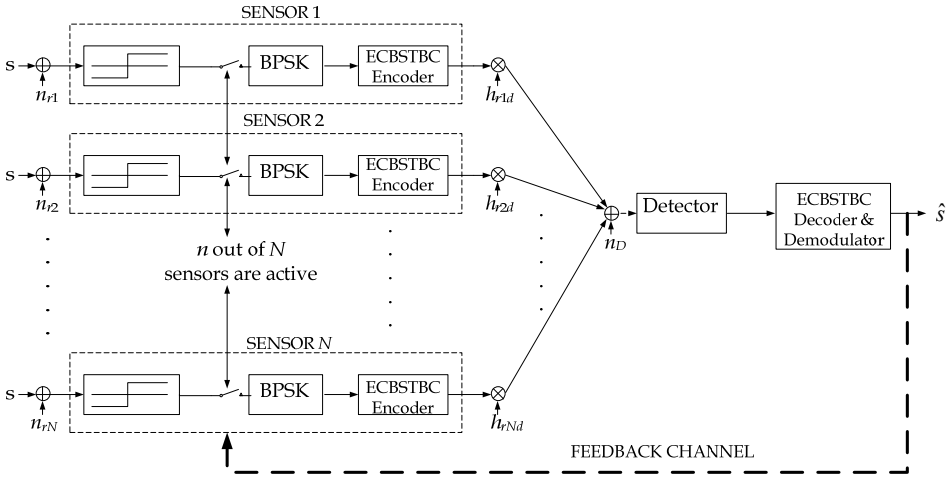


Fig. 2. Analitical model of wireless sensor network

2.2 Channel model

We assume that all parallel wireless channels are independent but they have statistically uniform paths with have identical means and variances (Cetinkaya, 2007). That is to say that the sensors-fusion center channels have equal variance and mean. This is not true for realistic scenarios, since some of the parallel channels have non-uniform statistical properties (Cetinkaya, 2007). In the non-uniform wireless channel simulations, the parallel channels may contain "better" or "worse" channels. When the *i*th active sensor-fusion center channel`s variance is much higher than the *j*th active sensor-fusion center channel`s variance ($\sigma^2_{rid} >> \sigma^2_{rjd}$ where j=1,..,n and $j \neq i$), this channel can be considered as "better" channel. On the contrary, when the *i*th sensor-fusion center channel`s variance is much lower than the *j*th sensor-fusion center channel`s variance ($\sigma^2_{rid} << \sigma^2_{rjd}$ where j=1,..,n and $j \neq i$), this channel can be called as "worse" channel (Ibrahim et al., 2008).

3. Extended Cooperative Balanced Space-Time Block Codes

The ECBSTBCs can be obtained from an OSTBC multiplied by an extension matrix. Since Alamouti`s code is the only orthogonal code with rate one and minimum delay, the ECBSTBCs can be obtained as an extension of the Alamouti`s code (Alamouti, 1998) as

$$\boldsymbol{C}=\boldsymbol{X}\boldsymbol{W}. \tag{1}$$

Here $\boldsymbol{X}$ is the Alamouti`s code matrix, $\boldsymbol{W}$ is a 2xn (n>2) matrix whose columns are 2x1 standard basis vectors, and the rank of $\boldsymbol{W}$ must be 2. The following example shows how to generate the ECBSTBCs for three active sensors. Consider the ECBSTBC pair with transmission matrix

$$\boldsymbol{C_1}=\begin{bmatrix} s_1 & s_2 & as_2 \\ -s_2^* & s_1^* & as_1^* \end{bmatrix} \tag{2}$$

where $a=e^{j2\pi m/q}$, q is the extension level and m=0, 1,…q-1. The columns and rows of $\boldsymbol{C_1}$ denote symbols transmitted from three active sensors in two signaling intervals, respectively. $\boldsymbol{C_1}$ is obtained from the Alamouti code using Equation (1) where

$$\boldsymbol{X}=\begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix} \quad \boldsymbol{W}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & a \end{bmatrix}. \tag{3}$$

In this fashion, arbitrary number of the ECBSTBCs can be generated by increasing the extension level. For that reason, the fusion center needs n+d feedback bits (n≥3) to select any possible ECBSTBCs where $d=\left\lceil (n-2)\log_2 q \right\rceil - 1$ (Eksim & Celebi, 2009b; Eksim, 2010b). n-2 feedback bits are needed to achieve full diversity as in Cooperative Balanced Space-Time Block Codes (CBSTBC) (Eksim & Celebi, 2007). The rest of the d+2 feedback bits provide additional coding gain.

The ECBSTBCs can be used in WSN. The ECBSTBC contains two phases: Measurement and cooperation. There are many measurement and cooperation phases respectively within a frame. Additionally, each frame includes an initialization phase. In the initialization phase, which occurs at the beginning of the each frame, the fusion center informs the active sensors about which ECBSTBC would be utilized within the frame using feedback channel. The selected code is fixed over one frame. In the measurement phase, each cooperating sensor makes two consecutive observation and binary quantization. The observation at each sensor is assumed to be Gaussian random variable with mean ±m and variance σ^2. In the cooperation phase of the ECBSTBCs, the fusion center receives the signal, $\boldsymbol{r_D}$,

$$\boldsymbol{r_D}=\sqrt{\frac{P}{N}}\boldsymbol{C}\boldsymbol{h_{rd}}+\boldsymbol{n_D}. \tag{4}$$

Here $\boldsymbol{h_{rd}}$ is the channel coefficient vector that contains path gains from the sensors to the fusion center, $\boldsymbol{n_D}$ is additive white Gaussian noise vector whose components are complex zero-mean with variance σ_D^2, P is the average total transmit power of the active sensors and $\boldsymbol{C}$ is the ECBSTBC matrix.

3.1 Three active sensors

Due to energy efficiency, when three sensors are active in the wireless environment, then, $\boldsymbol{C_1}$, $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$ are available ECBSTBC matrices. These matrices are

$$\boldsymbol{C_1}=\begin{bmatrix} s_1 & s_2 & as_2 \\ -s_2^* & s_1^* & as_1^* \end{bmatrix}\ \boldsymbol{C_2}=\begin{bmatrix} s_1 & s_2 & as_1 \\ -s_2^* & s_1^* & -as_2^* \end{bmatrix}\ \boldsymbol{C_3}=\begin{bmatrix} s_1 & as_1 & s_2 \\ -s_2^* & -as_2^* & s_1^* \end{bmatrix}. \tag{5}$$

Here a is the coefficient as defined previously. The fusion center selects the ECBSTBC $\boldsymbol{C}_j$, j=1,2,3 and the feedback bit a that gives the maximum coding gain. In this case, two bits of feedback is needed to select the ECBSTBC matrices and k bit of is needed to select the feedback bit a where $k=\lceil \log_2 q \rceil$.

The decoding of the ECBSTBCs is similar to CBSTBCs (Eksim & Celebi, 2007). Assume that the $\boldsymbol{C_1}$ matrix gives maximum coding gain. The received signals at fusion center are given as

$$\begin{aligned} r_{D,1} &= \sqrt{\frac{P}{3}}\left[h_{r1d}\overline{r}_{r1,1}+h_{r2d}\overline{r}_{r2,2}+ah_{r3d}\overline{r}_{r3,2}\right]+\eta_1 \\ r_{D,2} &= \sqrt{\frac{P}{3}}\left[-h_{r1d}\overline{r}^*_{r1,2}+h_{r2d}\overline{r}^*_{r2,1}+ah_{r3d}\overline{r}^*_{r3,1}\right]+\eta_2. \end{aligned} \tag{6}$$

Here $\overline{r}_{ri,j}$ is the observed data which includes observation and quantization noise by the ith active sensor at the jth symbol interval. Here η_1 and η_2 are noise at the fusion center. The fusion center estimates s_1 and s_2 by linear processing

$$\begin{aligned} \hat{s}_1 &= h^*_{r1d}r_{D,1}+(h_{r2d}+ah_{r3d})r^*_{D,2} \\ \hat{s}_2 &= (h_{r2d}+ah_{r3d})^* r_{D,1}-h_{r1d}r^*_{D,2}\ . \end{aligned} \tag{7}$$

Substituting $r_{D,1}$ and $r_{D,2}$ in Equation (7),

$$\begin{aligned} \hat{s}_1 &= \sqrt{\frac{P}{3}}\left[\left(\begin{array}{l} \left|h_{r1d}\right|^2+\left|h_{r2d}\right|^2+\left|h_{r3d}\right|^2 \\ +2\max\left(\mathrm{Re}\left\{ah^*_{r2d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r2d}\right\}\right)\end{array}\right)\right]s_1+\varphi_1 \\ \hat{s}_2 &= \sqrt{\frac{P}{3}}\left[\left(\begin{array}{l} \left|h_{r1d}\right|^2+\left|h_{r2d}\right|^2+\left|h_{r3d}\right|^2 \\ +2\max\left(\mathrm{Re}\left\{ah^*_{r2d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r2d}\right\}\right)\end{array}\right)\right]s_2+\varphi_2 \end{aligned} \tag{8}$$

where φ_1 and φ_2 are the noise terms which include both observation and quantization noise at the active sensors and the noise at the fusion center. The contribution of the $2\max\left(\mathrm{Re}\left\{ah^*_{r2d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r3d}\right\},\mathrm{Re}\left\{ah^*_{r1d}h_{r2d}\right\}\right)$ term in Equation (8) will always be positive and the gain will be greater than the sum of the magnitude squares of all path gains $\left(\left|h_{r1d}\right|^2+\left|h_{r2d}\right|^2+\left|h_{r3d}\right|^2\right)$. If the observation noise is very low, then, the diversity order

approaches to 3. It can be easily shown that the diversity order of the ECBSTBC approaches to n if n sensors are active when the observation noise is very low. A proof can be found in Appendix A.

4. Performance Evaluations

In the cooperative communication, transmitting only from selected relays is called distributed transmit antenna selection (DTAS) (Michalopoulos et al., 2008) which may be seen as an alternative approach to the ECBSTBCs. The criterion in selecting a single active sensor is the best instantaneous sensor-fusion center channel gain (Luo et al., 2005), and this is called as sensor selection (SS n:1) (Eksim & Celebi, 2009a; Eksim & Celebi, 2010a). To maximize signal-to-noise ratio (SNR) at the fusion center, two active sensors are chosen out of all active sensors and then the selected sensors transmit the received signals using the Alamouti scheme (Gore & Paulraj, 2002). In the simulations, the best active sensor pair which has the best instantaneous sensor-fusion center channel pair is selected. This is called as the sensor selection with Alamouti (SS n:2) (Eksim & Celebi, 2009a; Eksim & Celebi, 2010a).

The bit error probabilities of the ECBSTBC, SS, SS with Alamouti and statistical STBC cooperative diversity (Ohtsuki, 2006) are evaluated by computer simulations. A frame of 100 symbols is used. For meaningful comparison, the total transmission power and bandwidth are fixed, namely, the power is divided equally among cooperative active sensors. Each active sensor is assumed to observe either of two events H_0 and H_1 with equal probability. The observation at each sensor is assumed to be Gaussian random variable with mean $\pm m$ and variance σ^2. The noisy observation is quantized by the active sensors independently. Then, the quantized observation is transmitted according to selected transmission scheme.

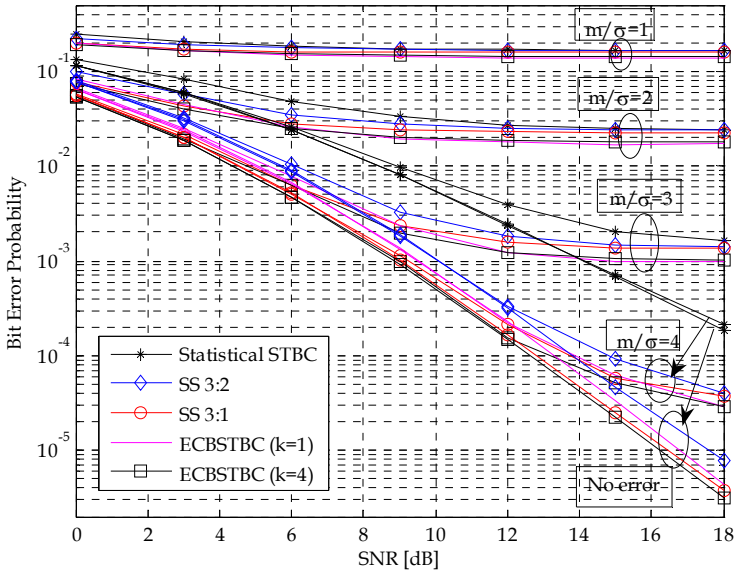


Fig. 3. The BER of three active sensors.

In Figure 3, the bit-error probability curves are shown for three active sensors. It is assumed that the ratio between the mean and the standard deviation of the observation in each active sensor (m/σ) is in the range of 1 and 4, and for comparison purposes no observation noise in each active sensor is also included in Figure 3. When m/σ is equal to 1 and 2, all transmission protocols give approximately similar performance since the observation noise limits the diversity gain. When m/σ is equal to 3, compared to the statistical STBC cooperative diversity (Statistical STBC), the SS with Alamouti's scheme (SS 3:2) provides an SNR advantage of approximately 3.73dB for a bit error rate (BER) value of P_b=2x10^{-3}. The SS scheme, the ECBSTBCs with one bit extension of feedback (ECBSTBC (k=1)), and the ECBSTBCs with four bit extension of feedback (ECBSTBC (k=4)) give additional 1.27dB, 1.77dB and 2.5dB SNR gains, respectively, compared to the SS with Alamouti´s scheme. If the value of m/σ increases, the diversity order of the statistical STBC cooperative diversity approaches to 2. However, the limited feedback schemes´ diversity order approaches to 3.
In Figure 4, the bit-error probability curves are shown for four active sensors. It is assumed that the ratio between the mean and the standard deviation of the observation in each active sensor (m/σ) is in the range of 1 and 4. When m/σ is equal to 1, all transmission protocols give approximately similar performance. For m/σ is being equal to 2, the statistical STBC cooperative diversity (Statistical STBC), the SS with Alamouti's scheme (SS 4:2) and the SS scheme (SS 4:1) reach to an error floor at BER value of P_b=2.3x10^{-2}. On the other hand, the ECBSTBCs with one bit extension of feedback (ECBSTBC (k=1)) and the ECBSTBCs with four bit extension of feedback (ECBSTBC (k=4)) reach to an error floor at BER value of P_b=7.65x10^{-3} and P_b=5.97x10^{-3}, respectively. When m/σ is equal to 3, compared to the

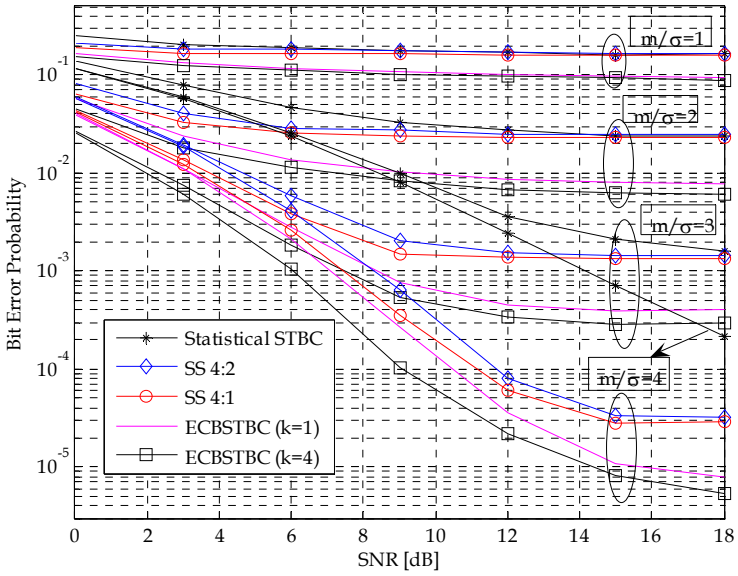


Fig. 4. The BER of four active sensors.

statistical STBC cooperative diversity, the SS with Alamouti's scheme (SS 4:2) provides an SNR advantage of approximately 6.26dB for a BER value of P_b=2x10^{-3}. The SS scheme (SS 4:1), the ECBSTBCs with one bit extension of feedback (ECBSTBC (k=1)) and the ECBSTBCs with four bit extension of feedback (ECBSTBC (k=4)) give additional 1.19dB, 2.54dB and 3.46dB SNR gains, respectively, compared to the SS with Alamouti´s scheme. When the value of m/σ increases, again, the diversity order of the statistical STBC cooperative diversity approaches to 2 because it utilizes only 2 active sensors. However, the diversity order of the limited feedback schemes approaches to 4.

In Figures 5-6, it is assumed that the sensor`s battery is limited. The linear battery model which is described in Section 2.1 is used. Four sensors are present in the wireless environment and all of them are active. It is assumed that the ratio between the mean and the standard deviation of the observation in each active sensor is equal to 3 (m/σ=3) and the sensors-fusion center channels´ SNR are 10dB. In Figure 5, four uniform sensor-fusion center channels are present in the wireless environment and their variances are equal to 1. Statistical STBC yields a BER value of P_b =7x10^{-3}. However limited feedback schemes such as the SS with Alamouti's (SS 4:2) and the SS (SS 4:1) yield BER values of P_b=1.8x10^{-3} and P_b=1.4x10^{-3}, respectively. The ECBSTBCs with one and four bit extension of feedback generate the BER values of P_b=5.74x10^{-4} and P_b=4.36x10^{-4}, respectively. Since the channels are uniform, all schemes sustain the QoS until the lifetime of the WSN.

In the Figure 6, two uniform, one "better" and one "worse" sensor-fusion center channels present in the wireless environment. The channel variances are 1, 10 and 0.1, respectively.

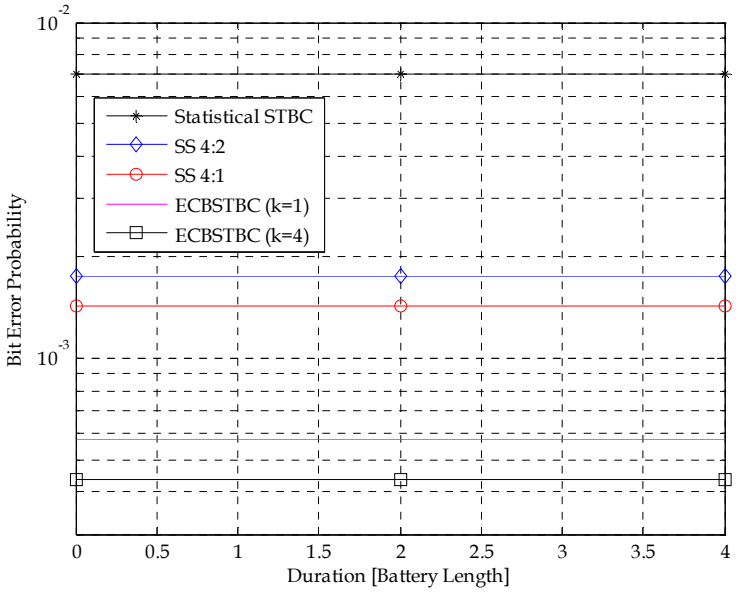


Fig. 5. The BER of four active sensors. The sensor-fusion center channels are 10dB and the parallel channels are uniform.

The SS scheme generally selects the active sensor which is present in the "better" sensor-fusion center channel. For this reason, the SS generates a BER value of P_b=1.3x10^{-3} until first sensor`s battery runs out. For this reason, the energy hole problem occurs. Then, the SS scheme generally selects two active sensors which are present in the uniform sensor-fusion center channels and the BER value increases to $P_b$=3.7x10$^{-3}$. Finally, the last active sensor`s battery runs out that is present in the "worse" sensor-fusion center channel. In this case, the BER value increases to P_b=0.1477. Due to the energy hole problem, similar scenario is valid for the SS with Alamouti's scheme. Statistical STBC generates a BER value of P_b=1.4x10^{-2}. The ECBSTBC with one and four bit extension of feedback result in BER values of P_b=1.2x10^{-3} and P_b=1.1x10^{-3}, respectively. In the non-uniform wireless parallel channels, the ECBSTBCs support QoS requirements until all sensors` batteries run out. This can be achieved via optimal distribution of transmission power among active sensors.

In Figures 7-8, four active sensors are present in the wireless environment and each active sensor transmits 1 feed forward bit to the fusion center (Eksim & Celebi, 2008). In this case, hybrid scheme which is proposed in (Eksim, 2010b) can be applied. This feed forward bit informs the fusion center that the observation noise at the active sensor is lower or higher according to a specified threshold value. When the active sensor´s observation noise is lower than the threshold, this active sensor will be selected for cooperation (Eksim, 2010b). When two active sensors observation noise is lower than the threshold, two active sensors employ Alamouti´s code to transmit their observations. If all active sensors observation noise is higher than the threshold, all active sensors are selected for cooperation. The selected ECBSTBC information is transmitted to the selected active sensors and they transmit their observations according to the selected ECBSTBC throughout the frame. Similar to the hybrid scheme, 1 feed forward bit can be utilized by the SS schemes. In this case, the SS schemes lead to lower BER values at the fusion center.

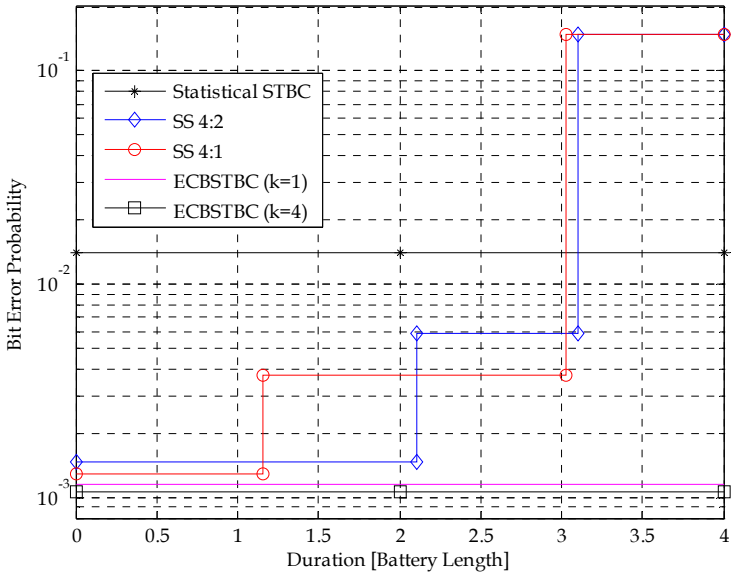


Fig. 6. The BER performances of four active sensors. The sensor-fusion center channels are 10dB and the parallel channels are non-uniform.

In Figure 7-8, it is assumed that the ratio between the mean and the standard deviation of the observation in each active sensor (m/σ) is equal to 2 and 3. In Figure 7, it can be observed that when m/σ is equal to 2, the statistical STBC cooperative diversity (Statistical STBC), the SS with Alamouti's scheme (SS 4:2) and the SS scheme (SS 4:1) reach to an error floor at BER value of P_b=2.3x10^{-2}. The ECBSTBCs with four bit extension of feedback (ECBSTBC (k=4)) reach to an error floor at the BER value of P_b =7.65x10^{-3}. On the other hand, the hybrid scheme with threshold 0.5m, not only the ECBSTBC with four bit extension of feedback (ECBSTBC (k=4, Th=0.5m)) but also the SS scheme (SS 4:1 (Th=0.5m)) and the SS scheme with Alamouti (SS 4:2 (Th=0.5m)) have error floors at lower BER values. In Figure 8, it can be observed that when m/σ is equal to 3, the statistical STBC cooperative diversity (Statistical STBC), the SS with Alamouti's scheme (SS 4:2) and the SS scheme (SS 4:1) cannot reach to the BER value of P_b =1x10^{-3}. The ECBSTBCs with four bit extension of feedback (ECBSTBC (k=4)) reach to an error floor at BER value of P_b =3x10^{-4}. On the other hand, the hybrid scheme with threshold 0.4m, the ECBSTBC with four bit extension of feedback (ECBSTBC (k=4, Th=0.4m)), the SS scheme (SS 4:1 (Th=0.4m)) and the SS scheme with Alamouti (SS 4:2 (Th=0.4m)) do not reach to an error floor even if signal-to-noise ratio is equal to 18dB.

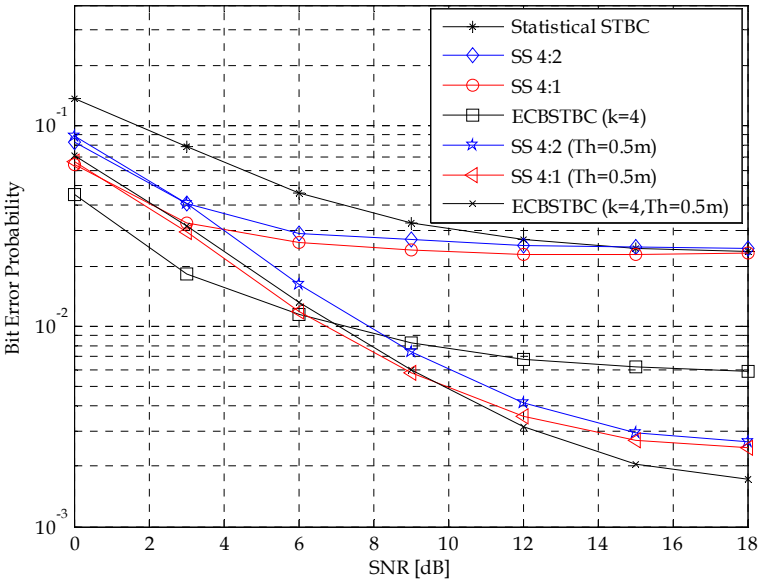


Fig. 7. The BER of four active sensors when m/σ=2.

5. Conclusions

In this chapter, methods increasing reliability of communications in WSNs are suggested. They are based on statistical cooperative diversity generating space-time block codes with limited feedback. It is shown that both SS schemes and ECBSTBC improve the performance of the statistical STBC with limited feedback, but the ECBSTBC have better signal-to-noise

ratio improvement compared to the SS schemes. Binary quantization is used and the quantization and the observation noise are taken into account. It is well known that the observation noise limits the BER performance. To diminish the effects of the observation noise, the ECBSTBC combined with SS scheme is proposed to improve the BER performance (Eksim, 2010b). This hybrid technique yields improved performance at the fusion center compared to solely using the ECBSTBC or the SS methods.
It is always assumed that when all of the sensor-fusion center channels are uniform or the sensors have unlimited battery. Then, the energy hole problem does not occur in WSN. This situation cannot be realized all the time in wireless environment and the energy hole problem occurs if the SS schemes are utilized. This problem is very significant in WSNs, since, in that case, the QoS cannot be maintained during the network lifetime. As opposed to the SS schemes, the ECBSTBC is a useful tool to alleviate the energy hole problem inherently. Since the ECBSTBC utilizes all active sensors to distribute transmission power among active sensors evenly when all active sensors present in non-uniform wireless channels.

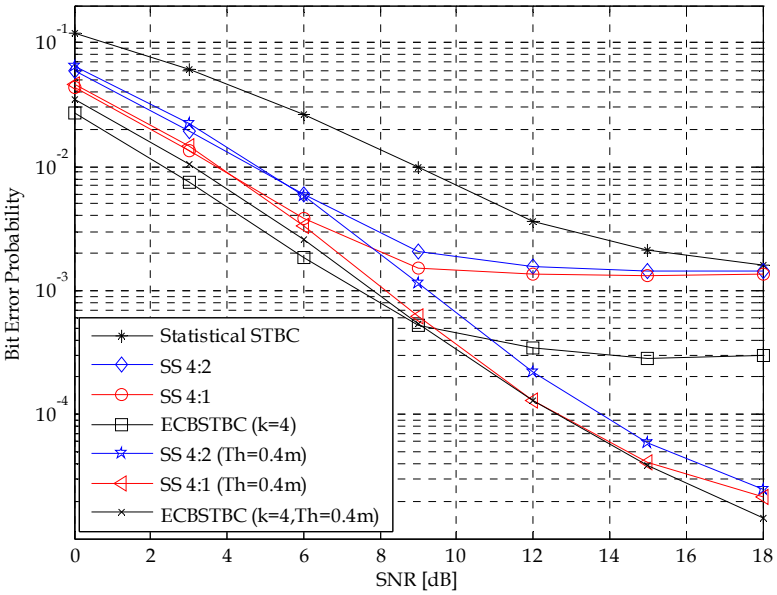


Fig. 8. The BER of four active sensors when m/σ=3.

6. Acknowledgement

The work of Mehmet Ertuğrul Çelebi is supported partially by the Scientific and Technological Research Council of Turkey (TUBITAK), Project No.107E022. The work of Ali Ekşim is supported partially by the European Commission (EC), FP-7 Project ICE, Project No. 206546.

Appendix A: Derivation of BER Upper Bound for ECBSTBC and Diversity

When three sensors are active, the value of m/σ is high and BPSK is used modulation scheme; Instantaneous signal-to-noise ratio at the fusion center, SNR_{fc}, can be written as follows

$$SNR_{fc} = SNR\frac{\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2 + 2\max\left(\mathrm{Re}\left\{ah_{r2d}^* h_{r3d}\right\}, \mathrm{Re}\left\{ah_{r1d}^* h_{r3d}\right\}, \mathrm{Re}\left\{ah_{r1d}^* h_{r2d}\right\}\right)\right)^2}{3\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2\right)} \quad \text{(A.1)}$$

Here $SNR=E_b/N_o$ is the signal-to-noise ratio per bit without fading. To find an upper bound, Equation (A.1) can be re-written as follows

$$SNR_{fc} \geq \frac{SNR}{3}\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2\right) \quad \text{(A.2)}$$

The bit error probability of BPSK is given in (Proakis, 2001).

$$P_b = Q\left(\sqrt{2SNR}\right) \quad \text{(A.3)}$$

where $Q(x)$ is the Q-function. Then, Put Equation (A.2) in place of Equation (A.3), the bit error probability is upper bounded by Q-function.

$$P_b \leq Q\left(\sqrt{2\frac{SNR}{3}\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2\right)}\right) \quad \text{(A.4)}$$

As it is well-known, the Q-function is upper bounded with exponential, thus, the BER can be upper bounded as follows

$$P_b \leq \exp\left(-\frac{SNR}{3}\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2\right)\right) \quad \text{(A.5)}$$

The BER upper bound averaged over channel statistics is given as

$$P_b \leq E\left\{\exp\left(-\frac{SNR}{3}\left(\left|h_{r1d}\right|^2 + \left|h_{r2d}\right|^2 + \left|h_{r3d}\right|^2\right)\right)\right\}. \quad \text{(A.6)}$$

Since the fading statistics h_{r1d}, h_{r2d} and h_{r3d} are independent; Equation (A.6) can be written as follows

$$P_b \leq E\left\{\exp\left(-\frac{SNR}{3}\left|h_{r1d}\right|^2\right)\right\}E\left\{\exp\left(-\frac{SNR}{3}\left|h_{r2d}\right|^2\right)\right\}E\left\{\exp\left(-\frac{SNR}{3}\left|h_{r3d}\right|^2\right)\right\}. \quad \text{(A.7)}$$

Evaluating Equation (A.7), we obtain the BER upper bound at the fusion center

$$P_b \le \left(\frac{3}{(SNR+3)} \right)^3 . \quad \text{(A.8)}$$

Above equation can be expanded to arbitrary number of active sensors, thus, the BER upper bound for n active sensors is given as

$$P_b \le \left(\frac{n}{(SNR+n)} \right)^n . \quad \text{(A.9)}$$

From Equation (A.9), the diversity is n when the value of m/σ is high.

7. References

Akyildiz, A. ; Su, W. ; Sankarasubramaniam, Y. & Çayırcı, E. (2002). A survey on sensor networks. *IEEE Commun. Mag.*, Vol. 40, No. 8, pp. 102-114

Alamouti, S.M. (1998). A simple transmit diversity technique for wireless communications. *IEEE J. Select. Areas Commun.*, Vol. 16, No. 8, pp. 1451-1458

Buttyan, L. & Hubaux, J.P. (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, Vol. 8, No. 5, pp. 579-592

Cetinkaya, C. (2007). Improving the efficiency of multipath traffic via opportunistic traffic scheduling. *Computer Networks*, Vol. 51, No. 8, pp. 2181-2197

Eksim, A. & Celebi, M.E. (2007). Diversity enchancement with cooperative balanced space-time block coding. *Proceedings of IEEE Int`l Symposium on Personal, Indoor & Mobile Communications*.

Eksim, A. & Celebi, M.E. (2008). Improvement on cooperative balanced space-time block coding with relay selection, IEEE 6th Int. Symp. on Communication Systems, Networks and Digital Signal Processing, CSNDSP, Graz, Austria, (2008).

Eksim, A. & Celebi, M.E. (2009a). Extended cooperative balanced space-time block coding for increased efficiency in wireless sensor networks (Work in Progress), Proceedings, Networking 2009, 456-467.

Eksim, A. & Celebi, M.E. (2009b). Extended balanced space-time block coding for wireless communications. *IET Signal Processing*, Vol. 3, No. 6, pp. 476-484.

Eksim, A. & Celebi, M.E. (2010a). Performance improvement of binary sensor based statistical STBC cooperative diversity using limited feedback. *IETE Technical Review*, Vol. 27, No. 1, pp. 60-67.

Eksim, A. (2010b). Extended Balanced Space-Time Block Coding in Wireless Networks, Ph.D. Thesis, Istanbul Technical University, Istanbul, Turkey, 2010.

Gore, D. & Paulraj, A. (2002). Antenna subset selection with space-time coding. *IEEE Transactions on Signal Processing*, Vol. 50, No. 10, pp. 2580-2588.

He, T. ; Krishnamurthy, S. ; Stankovic, J.A. ; Abdelzaher, T ; Luo, L. ; Stoleru, R. & et al. (2004). An energy-efficient surveillance system using wireless sensor networks, Proceedings, MobiSYS`04, 270-283.

Ibrahim, A.S. ; Sadek, A.K. ; Su, W. & Liu, K.J.R. (2008). Cooperative communications with relay-selection: When to cooperate and whom to cooperate with?. *IEEE Transactions on Wireless Communications*, Vol. 7, No. 7, pp. 2814-2827.

Lal, D., Manjeshwar, A., Herrman, F., Biyikoglu, E.U., ve Keshavarzian, A., (2003). Measurement and characterization of link quality metric in energy constrained wireless sensor networks, Proceedings, IEEE Global Telecommunications Conference, 446-452.

Laneman, J.N. & Wornell, G.W. (2003). Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Transactions on Information Theory*, Vol. 49, No. 10, pp. 2415-2425.

Li, X., ve Wu, N.E., (2003). Power efficient wireless sensor networks with distributed transmission-induced space spreading, Proceedings, 37th Asilomar Conference on Signals, Systems and Computers, 1698-1702.

Li, X. ; Chen, M. & Liu, W. (2005). Application of STBC-encoded cooperative transmissions in wireless sensor networks. *IEEE Signal Process. Lett.*, Vol. 12, No. 2, pp. 134-137.

Li, J. & Mohapatra, P. (2007). Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive and Mobile Computing*, Vol. 3, No. 3, pp. 233-254.

Lim, A., Srinivasan, V., ve Tham, C-K. (2005). A comparative study of cooperative algorithms for wireless ad hoc networks, Proceedings, REALMAN 2005.

Ng, F., Hwu, M., Chen, M., ve Li, X., (2005). Asynchronous space-time cooperative communications in sensor and robotic networks, Proceedings, International Conference on Mechatronics and Automation, 1624-1629.

Ohtsuki, T. (2006). Performance analysis of statistical STBC cooperative diversity using binary sensors with observation noise. *IEICE Trans. Commun.*, Vol. E89-B, No. 3, pp. 970-973.

Park, S., Savvides, A., ve Srivastava, M.B., (2001). Simulating networks of wireless sensors, Proceedings, Winter Simulation Conference, 1330-1338.

Proakis, J.G., (2001). Digital communications, McGraw-Hill, 4th Edition.

Sendonaris, A. ; Erkip, E. & Aazhang, B. (2003a). User cooperation diversity part I: System description. *IEEE Trans. Commun.*, Vol. 51, No. 11, pp. 1927-1938.

Sendonaris, A. ; Erkip, E. & Aazhang, B. (2003b). User cooperation diversity part II: Implementation aspects and performance analysis. *IEEE Trans. Commun.*, Vol. 51, No. 11, pp. 1939-1948.

Tarokh, V. ; Jafarkhani, H. & Calderbank, A.R. (1999). Space-time block codes from orthogonal designs. *IEEE Transactions on Information Theory*, Vol. 45, No. 5, pp. 1456-1467.

15

Time Synchronization in Wireless Sensor Networks

Jonggoo Bae and Bongkyo Moon
Dongguk University-Seoul
South Korea

1. Introduction

Recently small smart devices start to be embedded into the various environments in order to monitor the events occurred in the areas such as homes, plantations, oceans, rivers, streets, and highways. These tiny and low power devices which enable sensing and communication tasks have made sensor networks emerged. In wireless sensor networks (WSNs), especially, wireless devices get together and spontaneously form a network without any infrastructure. Due to the absence of infrastructure such as router in traditional network, nodes in a sensor network have to cooperate for communication by forwarding each other's packets from a source to its destination. Thus this yields a multi-hop communication environment.

Meanwhile, the knowledge of time between the sensor nodes is essential that detect the events such as target tracking, speed estimating, and ocean current monitoring. Hence, the sensed data often loses valuable context without accurate time information. With time synchronization, voice and video data from the different sensor nodes can be fused and displayed in a meaningful way at the sink. Time synchronization is a critical middleware service required for consistent distributed sensing and control in large-scale distributed systems such as sensor networks. That is, time synchronization in a WSN aims at providing a common time scale for local clocks of nodes in the network. Moreover, common services in WSNs, such as coordination, communication, security, power management and distributed logging also depend on the global time scale.

The most widely adapted time synchronization protocol in the internet domain is the Network Time Protocol (NTP) devised by Mills (Mills, 1991). Nodes could also be equipped with a global positioning system (GPS) to synchronize them (Hofmann-Wellenhof et al. 1997; Mannermaa et al. 1999). It is used to provide network-wide agreement among a large group of nodes in the Internet. NTP works well synchronizing the computers on the Internet, but is not designed with the energy and computation limitations of sensor nodes in mind. A GPS device may be too expensive to attach on cheap sensor devices, and GPS service may not be available everywhere, such as inside the buildings or under the water. Consequently, it may be useful to use NTP to discipline sensor nodes, but traditional synchronization schemes such as NTP or GPS are not suitable for use in sensor networks because of complexity and energy issues, cost and size factors. Therefore, without further adaptation, NTP is suitable only for WSN applications with low precision demands.

Time synchronization is a key service for many applications and operating systems in distributed computing environments. WSNs are large-scale distributed systems, but traditional distributed algorithms cannot be considered for problems due to their unique characteristics, especially the severe resource constraints. In this chapter, the mechanisms to synchronize the local clocks of the nodes in WSN have been extensively investigated.

2. Backgrounds and Related Works

A landmark study in computer clock synchronization is Lamport's work that elucidates the importance of *virtual clocks* in systems where causality is more important than absolute time (Lamport, 1978). Though Lamport's work focused on giving events a total order rather than quantifying the time difference between them, it has emerged as an important influence in sensor networks. Many sensor applications require only relative time, for example, timing the propagation delay of sound (Girod & Estrin, 2001), and thus absolute time may not be needed. Mills' NTP (Mills, 1991) stands out by virtue of its scalability, self-configuration in large multi-hop networks, robustness to failures and sabotage, and ubiquitous deployment. NTP allows construction of a hierarchy of time servers, multiply rooted at canonical sources of external time.

Post-facto synchronization was a pioneering work by Elson and Estrin (Elson & Estrin, 2001). In this approach, unlike in traditional synchronization schemes such as NTP, each node's clock is normally unsynchronized with the rest of the network; a beacon node periodically broadcasts beacon messages to the sensor nodes in its wireless range. When an event is detected, each node records the time of the event (timestamp with its own local clock). After the event (hence the name), upon receiving the reference beacon message, nodes use it as time reference and adjust their event timestamps with respect to that reference. This synchronization scheme has led afterwards to their RBS (Reference Broadcast Synchronization) protocol.

Elson et al. propose a scheme called Reference-Broadcast Synchronization (RBS), in which a node sends reference broadcast beacons to its neighbors using physical layer broadcasts (Elson et al., 2002). RBS gets around the non-determinism of packet send time, access time, and propagation time, while depending only on the packet receive time. Since the packet receive time is the same for all receivers, this reference broadcast packet can be used to synchronize a set of receivers with one another. This scheme can also be extended to a multi-hop scenario. However, the impact of the translation errors and delays on the multi-hop synchronization, which can be provided by translating the time between different broadcast domains, still needs to be studied. In addition, they do not consider global synchronization over the entire network.

A more recently developed *Time-Sync protocol for Sensor Networks* (TPSN) (Ganeriwal et al., 2003) is based on similar methodology as the NTP, where the sensor nodes are organized into multiple levels and synchronized to the root node of the hierarchy. Unlike the Internet, the root node and nodes at different levels responsible for synchronization may fail often, which may cause synchronization problems. In addition, mobile nodes may disrupt the predefined level-by-level synchronization procedure. On typical WSN platforms using the TPSN protocol, such as the Mica2 mote, it is possible to access directly to the MAC layer, and message time-stamping can be performed during message transmission and reception. This immediately eliminates the same three main sources of uncertainties as in RBS. With a

two-way handshake of synchronization messages, the TPSN protocol eliminates the unknown propagation time as well. Although the propagation time has been eliminated, the encoding and decoding times are not because they might not be the same on the sender and receiver side. It is important to point out that both the RBS and TPSN protocols suffer from the two largest sources of uncertainty of MAC layer time-stamping: the jitter of interrupt handling and decoding time.

On the other hand, the flooding time synchronization protocol (FTSP) effectively reduces all sources of time stamping errors except for the propagation time. The FTSP (Maroti et al. 2004) was designed for a sniper localization application requiring very high precision (Simon et al. 2004). FTSP achieves the required accuracy by utilizing a customized MAC-layer time stamping and by using calibration to eliminate unknown delays. FTSP is robust to network failures, as it uses flooding both for pair-wise and global synchronization. Linear regression from multiple timestamps is used to estimate the clock drift and offset. The main drawback of FTSP is that it requires calibration on the hardware actually used in the deployment (thus is not a software solution purely independent of the hardware). FTSP also requires intimate access to the MAC layer for multiple timestamps. However, if well-calibrated, the FTSP's precision is impressive (less than $2\mu s$).

Su and Akyildiz proposed the time-diffusion synchronization protocol (TDP) for network-wide time synchronization (Su & Akyildiz, 2005). The main idea of TDP is to start from a master node, adjust the clocks of its neighbors, and diffuse this clock adjustment to other nodes. TDP maintains global time synchronization within an adjustable bound based on the application requirements. One of the benefits of TDP is that the performance of voice and video applications can be improved when multiple sources send data back to the sink through flooding or *directed diffusion* (Intanagonwiwat et al. 2003). It achieves global synchronization by multi-hop flooding: The base station initiates the protocol by sending a special timing message to the entire network. Some of the nodes, upon receiving the message, become masters by using a leader election procedure. The master nodes start the time-diffusion procedure involving electing diffused leaders, multi-hop flooding, and iterative weighted averaging of timings from different master nodes. TDP handles node mobility and failures by using a peer evaluation procedure.

3. Time Synchronization

3.1 Clocks and Synchronization

3.1.1 Sensor Node Clock

Every sensor node maintains its own clock and this is the only notion of time that a node has. The clock is an ensemble of hardware and software components; it is essentially a timer that counts the oscillations of a quartz crystal running at a particular frequency. Computing devices are mostly equipped with a hardware oscillator assisted computer clock, which implements an approximation $C(t)$ of real-time t. Let us represent the clock for node A by $C_A(t)$. The difference in the clocks of two sensor nodes (i.e., A and B) is referred as the offset error between them. There are three reasons for the nodes to be representing different times in their respective clocks (Ganeriwal et al. 2008): 1) The nodes might have been started at different times, 2) the quartz crystals at each of these nodes might be running at slightly different frequencies, causing the clock values to gradually diverge from each other (termed

as the skew error), or 3) the frequency of the clocks can change differently over time because of aging or ambient conditions such as temperature (termed as the drift error). These errors can be summarized as follows:

$$\text{Offset: } \delta = C_A(t) - C_B(t) \tag{1}$$

$$\text{Skew: } \eta = \frac{\partial C_A(t)}{\partial t} - \frac{\partial C_B(t)}{\partial t} \tag{2}$$

$$\text{Drift: } \lambda = \frac{\partial^2 C_A(t)}{\partial t^2} - \frac{\partial^2 C_B(t)}{\partial t^2} \tag{3}$$

The angular frequency of the hardware oscillator determines the rate at which the clock runs. The rate of a perfect clock, which can be denoted as dC/dt, would equal 1, however, all clocks are subject to a clock drift; oscillator frequency will vary unpredictably due to various physical effects. Even though the frequency of a clock changes over time, it can be approximated with good accuracy by an oscillator with fixed frequency (Sichitiu & Veerarittiphan, 2003). Then, for some node *i* in the network, we can approximate its local clock as:

$$C_i(t) = a_i t + b_i \tag{4}$$

where $a_i(t)$ is the clock *drift*, and $b_i(t)$is the *offset* of node *i*'s clock. *Drift* denotes the rate (frequency) of the clock, and *offset* is the difference in value from real time *t*. Using equation (4), we can compare the local clocks of two nodes in a network, say node i and node j as:

$$C_i(t) = a_{ij} \cdot C_j(t) + b_{ij} \tag{5}$$

We call a_{ij} the *relative drift*, and b_{ij} the *relative offset* between the clocks of node i and node j.

If two clocks are perfectly synchronized, then their relative drift is i (meaning the clocks have the same rate) and their relative offset is zero (meaning they have the same value at that instant). Some studies in the literature use "skew" instead of "drift", defining it as the *difference* (as opposed to *ratio*) between clock rates. Also, the "offset" may equivalently be mentioned as "phase offset". Fig. 1 shows the relationship between relative drift and offset.

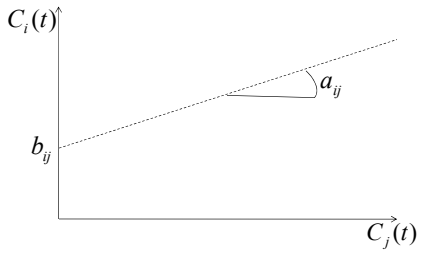


Fig. 1. The relation between relative drift and offset

Although each sensor node is equipped with a hardware clock, these hardware clocks can usually not be used directly, as they suffer from severe drift. No matter how well the

hardware clocks will be calibrated at deployment, the clocks will ultimately exhibit a large skew. Since all hardware clocks are imperfect, local clocks of nodes may drift away from each other in time, hence observed time or durations of time intervals may differ for each node in the network. To allow for an accurate common time, nodes need to exchange messages from time to time, constantly adjusting their clock values. Furthermore, nodes can convert the current hardware clock reading into a logical clock value and vice versa (Sommer & Wattenhofer, 2009).

- Hardware Clock

Each sensor node i is equipped with a hardware clock $H_i(\cdot)$. The clock value at time t is defined as

$$H_i(t) = \int_{t_0}^{t} h_i(\tau) d\tau + \Phi_i(t_0)$$

where $h_i(\tau)$ is the hardware clock rate at time τ and $\Phi_i(t_0)$ is the hardware clock offset at time t_0. It is assumed that hardware clocks have bounded drift, i.e., there exists a constant $0 \le \rho <$ 1 such that $1 - \rho \le h(t) \le 1 + \rho$ for all times t. This implies that the hardware clock never stops and always makes progress with at least a rate of $1 - \rho$. This is a reasonable assumption since common sensor nodes are equipped with external crystal oscillators which are used as clock source for a counter register of the microcontroller. These oscillators exhibit drift which is only gradually changing depending on the environmental conditions such as ambient temperature or battery voltage and on oscillator aging. This allows assuming the oscillator drift to be relatively constant over short time periods. Crystal oscillators used in sensor nodes normally exhibit a drift between 30 and 100 ppm (Sommer & Wattenhofer, 2009).

- Logical Clock

Since other hardware components may depend on a continuously running hardware clock, its value should not be adjusted manually. Instead, a logical clock value $L_i(\cdot)$ is computed as a function of the current hardware clock. The logical clock value $L_i(t)$ represents the synchronized time of node i. It is calculated as follows:

$$L_i(t) = \int_{t_0}^{t} h_i(\tau) \cdot l_i(\tau) d\tau + \theta_i(t_0)$$

where $l_i(\tau)$ is the *relative logical clock rate* and $\theta_i(t_0)$ is the clock offset between the hardware clock and the logical clock at the reference time t_0. The logical clock is maintained as a software function and is only calculated on request based on a given hardware clock reading (Sommer & Wattenhofer, 2009).

3.1.2 Definition of Clock Synchronization

The synchronization problem on a network of n devices corresponds to the problem of equalizing the computer clocks of the different devices. The synchronization can be either *global;* trying to equalize $C_i(t)$ for all $i = 1::n$ or it can be *local;* trying to equalize $C_i(t)$ for some set of the nodes that are spatially close. Equalizing just the instantaneous values of clocks by correcting the offsets is not enough for synchronization since the clocks will drift away

afterwards. Therefore a synchronization scheme should either equalize the clock rates as well as offsets, or it should repeatedly correct the offsets in order to keep the clocks synchronized over a time period (Sivrikaya & Yener, 2004) .
The above definition of synchronization actually defines the strictest form of synchronization, where one seeks perfect matching of time on different clocks, but this definition can be relaxed to different degrees according to the needs of an application. In general, the synchronization problem can be classified into three basic types (Ganeriwal et al. 2003). First form of synchronization deals only with ordering of events or messages. The aim of such an algorithm is to be able to tell whether an event $E1$ has occurred before or after another event $E2$, i.e. just to compare the local clocks for order rather than having them synchronized. The algorithm proposed in (Romer, 2003) is an example to this type of synchronization. Second type of synchronization algorithms targets maintaining relative clocks. In this scheme, nodes run their local clocks independently, but they keep information about the relative drift and offset of their clock to other clocks in the network, so that at any instant, the local time of the node can be converted to some other node's local time and vice versa. Most of the synchronization schemes proposed for sensor networks use this model (Elson et al. 2002; Sichitiu & Veerarittiphan, 2003). The third form of synchronization is the "always on" model where all nodes maintain a clock that is synchronized to a reference clock in the network. The goal of this type of synchronization algorithms is to preserve a global timescale throughout the network. The synchronization scheme of (Ganeriwal et al. 2003) conforms to this model, but the use of "always on" mode is not mandatory in the scheme.

3.2 Design Factors for Time Synchronization

Some of the factors influencing time synchronization in wireless sensor networks are temperature, phase noise, frequency noise, asymmetric delays, and clock glitches (Su & Akyildiz, 2005).

- Temperature: Since sensor nodes are deployed in various places, the temperature variations throughout the day may cause the clock to speed up or slow down. For a typical sensor node, the clock drifts few *parts per million (ppm)* during the day (Mills, 1998). For low-end sensor nodes, the drifting may be even worse.
- Phase noise: Some of the causes of phase noise are access fluctuations at the hardware interface, response variation of the operating system to interrupts, and jitter in the network delay. The jitter in the network delay may be due to medium access and queueing delays.
- Frequency noise: The frequency noise is due to the unstability of the clock crystal. A low-end crystal may experience large frequency fluctuation, because the frequency spectrum of the crystal has large sidebands on adjacent frequencies.
- Asymmetric delay: Since sensor nodes communicate with each other through the wireless medium, the delay of the path from one node to another may be different than the return path. As a result, an asymmetric delay may cause an offset to the clock that cannot be detected by a variance type method (Levine, 1999). If the asymmetric delay is static, the time offset between any two nodes is also static. The

asymmetric delay is bounded by one-half the round trip time between the two nodes (Levine, 1999).

• Clock glitches: Clock glitches are sudden jumps in time. This may be caused by hardware or software anomalies such as frequency and time steps. Besides dealing with these factors, a time synchronization protocol for sensor networks should be *automatically self-configured* and be *sensitive to energy requirement*.

3.3 Synchronization Problems in WSNs

Network time protocol (NTP) (Mills 1991) has been widely used in the Internet for decades. The NTP clients synchronize their clocks to the NTP time servers with accuracy in the order of milliseconds by statistical analysis of the round-trip time. The time servers are synchronized by external time sources, typically using GPS. The NTP has been widely deployed and proved to be effective, secure and robust in the internet. However, traditional synchronization schemes and GPS-equipped systems are not suitable for use in WSNs due to the specific requirements of those networks (Yoon et al. 2007):

• Precision: WSNs may require much higher precision than traditional networks depending on the deployed applications. For example, a precision of a few milliseconds is satisfactory for NTP in the Internet, while microsecond precision may be required in order to significantly improve the performance of the WSN beam-forming application.

• Cost: Nodes in WSNs typically have limited batteries, computational resources, and storage capacity. However, most of the protocols designed for wired environments need to exchange many messages and also store them for statistical processing.

The problem in a modern sensor network scenario is that nodes can only communicate locally to their neighbors. The localized communication makes the problem much harder in that: 1) a valid consensus has to be computed locally and 2) the local consensus must be conveyed to other parts of the network; this is even harder because the relay nodes may be faulty or malicious. In order to provide network-wide time synchronization, the time differences among the sensor nodes must be minimized before protocols requiring time-stamps (e.g., security applications, flow control protocols, target tracking, voice fusion, video fusion, and environmental data fusion) are realizable. In addition, the time synchronization protocol must be robust to node failures as well as energy consumption in the network .

Typically the synchronization problems in wireless sensor networks need to be addressed for the following reasons (Sivrikaya & Yener, 2004). First, sensor nodes need to coordinate their operations and collaborate each other in order to achieve a complex sensing task. That is, data fusion is made through aggregating data collected from different nodes for a meaningful result. Second, power saving function requires synchronization for increasing network lifetime. For power saving, sensors may *sleep* by turning off their sensors and/or transceivers at appropriate times, and wake up at coordinated times. However, the radio receiver of a sensor node is not turned off in the case that there are some data directed to it. This requires a precise timing between sensor nodes. Third, scheduling algorithms in WSNs are used to share the transmission medium in the time domain to eliminate transmission

collisions and conserve energy. However, non-determinism in transmission time caused by the Media Access Channel (MAC) layer of the radio stack can introduce several hundreds of milliseconds delay at each hop. Thus, synchronization is an essential part of transmission scheduling.

3.4 Uncertainties and Errors in Time Synchronization

Time synchronization schemes rely on some sort of message exchange between nodes in WSN. Non-determinism in the network dynamics such as propagation time or physical channel access time makes the synchronization task a big challenge in many systems. Note that in short distance multi-hop broadcast, the data processing time and its variation contribute the most to time fluctuations and differences in the path delays. Also, the time difference between two sensor nodes may become large over time due to the wandering effect of the local clocks. Latency estimates are actually confounded by random events that lead to asymmetric round-trip message delivery delays; this delay prevents the receiver from exactly comparing the local clocks of the two nodes and accurately synchronizing to the sender node. To better understand the source of these errors, it is useful to decompose the source of a message's latency. Kopetz and Ochsenreiter (Kopetz & Ochsenreiter, 1987) introduced firstly four distinct components for analyzing the sources of the message delivery delays and later extended in (Ganeriwal et al. 2003).

- *Send Time:* The time spent at the sender to construct the message. This includes kernel protocol processing and variable delays introduced by the operating system (e.g., context switches and system call overhead occurred by the synchronization application), and the time to transfer the message from the host to its network interface for transmission.
- Access Time: Each packet faces some delay at the MAC (Medium Access Control) layer before actual transmission. This delay is specific to the MAC protocol in use, but some typical reasons for delay are waiting for the channel to be idle or waiting for the TDMA slot for transmission.
- Propagation Time: This is the time spent in propagation of the message between the network interfaces of the sender and the receiver. When the sender and receiver share access to the same physical media (e.g., neighbors in an ad-hoc wireless network, or on a LAN), this delay is very small as it is simply the physical propagation time of the message through the media.
- Receive Time: This is the processing time required for the receiver's network interface to receive the message from the channel and notify the host of its arrival. This is typically the time required for the network interface to generate a message reception signal. If the arrival time is time-stamped at a enough low level in the host's operating system kernel, this delay does not include the overhead of system calls, context switches, or even the message transfer from the network interface to the host.
- Transmission Time: The time it takes for the sender to transmit the message. This time is in the order of tens of milliseconds depending on the length of the message and the speed of the radio.

• Reception Time: The time it takes for the receiver to receive the message. It is the same as the transmission time. The transmission and reception times overlap in WSN as pictured in Fig. 2.

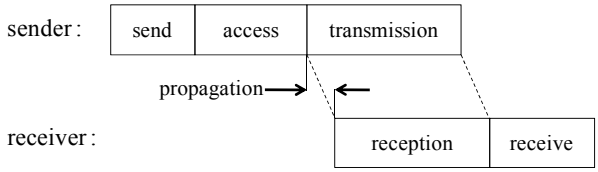


Fig. 2. Decomposition of the message delivery delay over a wireless link (Maroti, et al. 2004)

• Interrupt Handling Time: The delay between the radio chip raising and the microcontroller responding to an interrupt. This time is mostly less than a few microsecond (waiting for the microcontroller to finish the currently executed instruction), however, when interrupts are disabled this delay can grow large.

• Encoding Time: The time it takes for the radio chip to encode and transform a part of the message to electromagnetic waves starting from the point when it raised an interrupt indicating the reception of the idealized point from the microcontroller. This time is deterministic and is in the order of a hundred microseconds.

• Decoding Time: The time it takes for the radio chip on the receiver side to transform and decode the message from electromagnetic waves to binary data. It ends when the radio chip raises an interrupt indicating the reception of the idealized point. This time is mostly deterministic and is in the order of hundred microseconds. However, signal strength fluctuations and bit synchronization errors can introduce jitter.

• Byte Alignment Time: The delay incurred because of the different byte alignment of the sender and receiver. This time is deterministic and can be computed on the receiver side from the bit offset and the speed of the radio.

Fig. 3 summarizes the decomposition of delivery delay of the idealized point of the message as it traverses over a wireless channel. Each line represents the time line of the layer as measured by an ideal clock. The dots represent the time instance when the idealized point of the message crosses the layers. The triangles on the first and last line represent the time when the CPU makes the time-stamps. Depending on the specific hardware the time stamp is usually recorded by the microcontroller when it handles the radio chip interrupts both on the sender and receiver sides. Alternatively, capture registers provided by some hardware can be employed to eliminate the interrupt handling time (Maroti, et al. 2004).

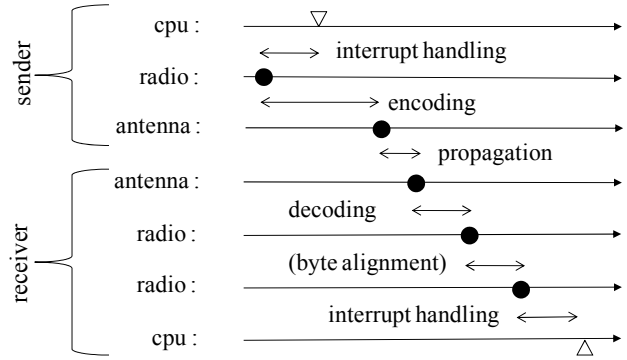


Fig. 3. The timing of the transmission of an idealized point in the software (cpu), hardware (radio chip) and physical (antenna) layers of the sender and the receiver (Maroti, et al. 2004)

Table 1 summarizes the magnitudes and distribution of the various delays in message transmissions on the Mica2 platform. The block codes are used, and the idealized point of the message can also be assumed to be at a block boundary (Maroti, et al. 2004).

Time	**Magnitude**	**Distribution**
Send & Receive	0 – 100 ms	nondeterministic, depends on the processor load
Access	10 – 500 ms	nondeterministic, depends on the channel contention
Transmission & Reception	10 – 20 ms	deterministic, depends on message length
Propagation	< 1μs for distances up to 300 meters	deterministic, depends on the distance between sender and receiver
Interrupt Handling	< 5μs in most cases, but can be as high as 30μs	nondeterministic, depends on interrupts being disabled
Encoding plus Decoding	100 – 200 μs < 2 μs variance	deterministic, depends on radio chipset and settings
Byte Alignment	0 – 400μs	deterministic, can be calculated

Table 1. The sources of delays in message transmissions (Maroti, et al. 2004)

3.5 Metrics for Evaluating Time Synchronization Schemes

The requirements for the synchronization problem can be regarded as the metrics for evaluating synchronization schemes on wireless sensor networks. Combining with the criteria that sensor nodes have to be energy efficient, low-cost, and small in a multi-hop environment, this requirement becomes a challenging problem to solve. However, a single synchronization scheme may not satisfy them all together since there are actually tradeoffs between the requirements of an efficient solution (Sivrikaya & Yener, 2004).

• Energy Efficiency: As with all of the protocols designed for sensor networks, synchronization schemes should take into account the limited energy resources contained in sensor nodes.

• Scalability: Most sensor network applications need deployment of a large number of sensor nodes. A synchronization scheme should scale well with increasing number of nodes and/or high density in the network.

• Precision: The need for precision, or accuracy, may vary significantly depending on the specific application and the purpose of synchronization. For some applications, even a simple ordering of events and messages may suffice whereas for some others, the requirement for synchronization accuracy may be on the order of a few ^{1}secs.

• Robustness: A sensor network is typically left unattended for long times of operation in possibly hostile environments. In case of the failure of a few sensor nodes, the synchronization scheme should remain valid and functional for the rest of the network.

• Lifetime: The synchronized time among sensor nodes provided by a synchronization algorithm may be instantaneous, or may last as long as the operation time of the network.

• Scope: The synchronization scheme may provide a global time-base for all nodes in the network, or provide local synchronization only among spatially close nodes. Because of the scalability issues, global synchronization is difficult to achieve or too costly (considering energy and bandwidth usage) in large sensor networks. On the other hand, a common time-base for a large number of nodes might be needed for aggregating data collected from distant nodes, dictating a global synchronization.

• Cost and Size: Wireless sensor nodes are very small and inexpensive devices. Therefore, as noted earlier, attaching a relatively large or expensive hardware (such as a GPS receiver) on a small, cheap device is not a logical option for synchronizing sensor nodes. The synchronization method for sensor networks should be developed with limited cost and size issues in mind.

• Immediacy: Some sensor network applications such as emergency detection (e.g. gas leak detection, intruder detection) require the occurring event to be communicated immediately to the sink node. In this kind of applications, the network cannot tolerate any kind of delay when such an emergency situation is detected. This is called the immediacy requirement, and might prevent the protocol designer from relying on excessive processing after such an event of interest occurs, which in turn requires that nodes be *pre-synchronized* at all times.

4. Time Synchronization Methods

Time synchronization has been a seminal topic in distributed systems (Dolev et al. 1984; Halpern et al. 1984; Lundelius et al. 1984; Lamport et al. 1985), but designing clock synchronization algorithms in the context of a sensor network is challenging for several reasons. First, traditional distributed systems assume that all the nodes in a network can communicate directly with each other. A sensor network, however, is subject to spatial

constraints. Nodes only communicate directly with their neighbors. Communication between two remote nodes is accomplished by message relay using intermediate nodes. Second, nodes in a sensor network generally rely on less information about the system than traditional distributed systems, where nodes have access to the clock values of all the other members of the system, including the faulty nodes. Third, a sensor node has only limited processing capability. The computation intensive signature algorithms, such as RSA, are not suitable for sensor networks. Instead, some light-weight algorithms (such as using a one-way key chain or a key management scheme) are more suitable. The spatial constraints, the communication cost and delay, and the diminished computational capability are key reasons why localized algorithms that involve lightweight computations are preferred for sensor networks.

4.1 RBS(Reference Broadcast Synchronization)

The main advantage of RBS is that it eliminates transmitter-side non-determinism. The disadvantage of the approach is that additional message exchange is necessary to communicate the local time-stamps between the nodes. Eventually the RBS approach completely eliminates the send and access times, and with minimal OS modifications it is also possible to remove the receive time uncertainty. This leaves the mostly deterministic propagation and reception time in wireless networks as the sole source of error. The main strength of RBS is its broad applicability to commodity hardware and existing software in sensor networks as it does not need access to the low levels of the operating system (Elson et al. 2002).

The novel idea in RBS scheme is to use a third party for synchronization instead of synchronizing the sender with a receiver. This scheme synchronizes a set of receivers with one another. Although its application in sensor networks is novel, the idea of *receiver-receiver synchronization* was previously proposed for synchronization in broadcast environments. In RBS scheme, nodes send reference beacons to their neighbors. A reference beacon does not include a timestamp, but instead, its time of arrival is used by receiving nodes as a reference point for comparing clocks (Sivrikaya & Yener, 2004).

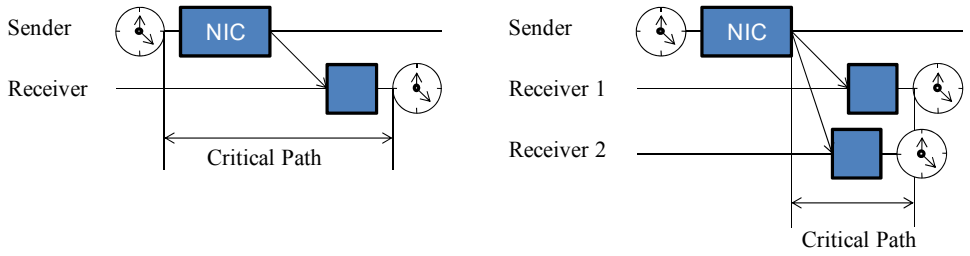


Fig. 4. Critical path analysis between traditional time synchronization protocol (*left*) and RBS (*right*) (Elson et al. 2002)

By removing the sender's non-determinism from the critical path (Fig. 4), RBS scheme achieves much better precision compared to traditional synchronization methods that use two-way message exchanges between synchronizing nodes. As the sender's non-determinism has no effect on RBS precision, the only sources of error can be the non-

determinism in propagation time and receive time. In this scheme, a single broadcast will propagate to all receivers at essentially the same time, and hence the propagation error is negligible. This is especially true when the radio ranges are relatively small (compared to speed of light times the required synchronization precision), as is the case for sensor networks. So the only receive time errors are handled when the accuracy of RBS model is analyzed (Elson et al. 2002; Sivrikaya & Yener, 2004) .
In the simplest form of RBS, a node broadcasts a single pulse to two receivers. The receivers, upon receiving the pulse, exchange their receiving times of the pulse, and try to estimate their relative phase offsets. This basic RBS scheme can be extended in two ways: 1) allowing synchronization between *n* receivers by a single pulse, where *n* may be larger than two, 2) increasing the number of reference pulses to achieve higher precision.

4.2 TPSN (Timing-Sync Protocol for Sensor Network)

The TPSN algorithm first creates a spanning tree of the network and then performs pair-wise synchronization along the edges. Each node gets synchronized by exchanging two synchronization messages with its reference node one level higher in the hierarchy. The TPSN achieves two times better performance than RBS by time-stamping the radio messages in the Medium Access Control (MAC) layer of the radio stack (Ganeriwal et al., 2003) and by relying on a two-way message exchange. The shortcoming of TPSN is that it does not estimate the clock drift of nodes, which limits its accuracy, and does not handle dynamic topology changes.
The first step of the algorithm is to create a hierarchical topology in the network. Every node is assigned a level in this hierarchical structure, and a node belonging to level *i* can communicate with at least one node belonging to level *i-1*. Only one node is assigned to level *0*, which is called the "root node". This stage of the algorithm is called as the "level discovery phase". Once the hierarchical structure has been established, the root node initiates the second stage of the algorithm, which is called the "synchronization phase". In this phase, a node belonging to level *i* synchronize to a node belonging to level *i-1*. Eventually every node is synchronized to the root node and network-wide time synchronization is achieved (Ganeriwal et al., 2003).

4.2.1 Level Discovery Phase

This phase of the algorithm occurs at the onset, when the network is deployed. The root node is assigned a level *0* and it initiates this phase by broadcasting a *level_discovery* packet. The *level_discovery* packet contains the identity and the level of the sender. The immediate neighbors of the root node receive this packet and assign themselves a level, one greater than the level they have received i.e., level *1*. After establishing their own level, they broadcast a new *level_discovery* packet containing their own level. This process is continued and eventually every node in the network is assigned a level. On being assigned a level, a node neglects any such future packets. This makes sure that no flooding congestion takes place in this phase. Thus a hierarchical structure is created with only one node, root node, at level *0*. A node might not receive any *level_discovery* packets owing to MAC layer collisions (Ganeriwal et al., 2003).

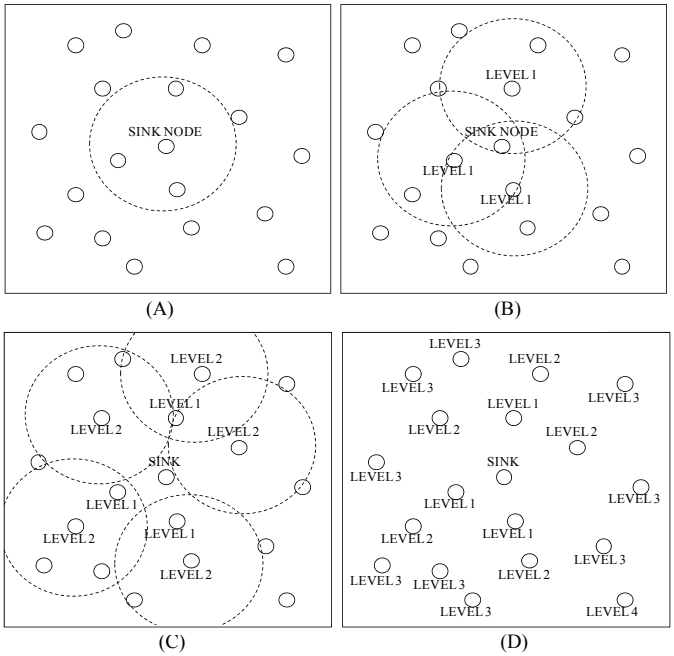


Fig. 5. The Process of level discovery phase for hierarchical topology organization in TPSN

4.2.2 Synchronization Phase

In this phase, pair wise synchronization is performed along the edges of the hierarchical structure established in the earlier phase. The classical approach of sender-receiver synchronization (Mills, 1991) is used for doing this handshake between a pair of nodes. Fig. 6 shows this message-exchange between nodes '*A*' and '*B*'. Here, *T1, T4* represent the time measured by local clock of *'A'*. Similarly *T2, T3* represent the time measured by local clock of *'B'*. At time *T1*, '*A*' sends a *synchronization_pulse* packet to '*B*'. The *synchronization_pulse packet* contains the level number of '*A*' and the value of *T1*. Node *B* receives this packet at *T2,* where *T2* is equal to *T1* + D + *d*. Here D and *d* represents the clock drift between the two nodes and propagation delay respectively. At time *T3*, '*B*' sends back an *acknowledgement* packet to '*A*'. The *acknowledgement* packet contains the level number of *'B'* and the values of *T1, T2* and *T3*. Node *A* receives the packet at *T4*. Assuming that the clock drift and the propagation delay do not change in this small span of time, *'A'* can calculate the clock drift and propagation delay as (Ganeriwal et al., 2003) :

$$\Delta = \left(\frac{T2 - T1) - (T4 - T3)}{2}\right) \; ; \; d = \left(\frac{T2 - T1) + (T4 - T3)}{2}\right) \tag{6}$$

Knowing the drift, node *A* can correct its clock accordingly, so that it synchronizes to node *B*. This is a sender initiated approach, where the sender synchronizes its clock to that of the receiver.

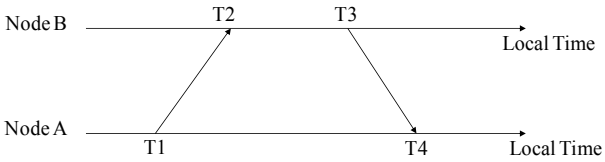


Fig. 6. Two way message exchange between a pair of nodes (Ganeriwal et al., 2003)

This message exchange at the network level begins with the root node initiating the phase by broadcasting a *time_sync* packet. On receiving this packet, nodes belonging to level *1* wait for some random time before they initiate the two-way message exchange with the root node. This randomization is to avoid the contention in medium access. On receiving back an acknowledgment, these nodes adjust their clock to the root node. The nodes belonging to level *2* will overhear this message exchange. This is based on the fact that every node in level *2* has at least one node of level *1* in its neighbor set. On hearing this message, nodes in level *2* back off for some random time, after which they initiate the message exchange with nodes in level *1* (Ganeriwal et al., 2003).
This randomization is to ensure that nodes in level *2* start the synchronization phase after nodes in level *1* have been synchronized. Note that a node sends back an *acknowledgement* to a *synchronization_pulse*, provided that it has synchronized itself. This ensures that no multiple levels of synchronization are formed in the network. This process is carried out throughout the network and eventually every node is synchronized to the root node. In a sensor network, packet collisions can take place quite often. To handle such scenario a node waiting for an acknowledgement, timeouts after some random time and retransmits the *synchronization_pulse*. This process is continued until a successful two-way message exchange has been done (Ganeriwal et al., 2003).

4.3 FTSP(Flooding Time Synchronization Protocol)

The goal of the FTSP is to achieve a network wide synchronization of the local clocks of the participating nodes. In this protocol, each node has a local clock exhibiting the typical timing errors of crystals and can communicate over an unreliable but error corrected wireless link to its neighbors. The FTSP synchronizes the time of a sender to possibly multiple receivers utilizing a single radio message time-stamped at both the sender and the receiver sides. MAC layer time-stamping can eliminate many of the errors, as observed in many previous protocols (Ganeriwal et al., 2003; Woo & Culler, 2001). However, accurate time-synchronization at discrete points in time is a partial solution only. Compensation for the clock drift of the nodes is inevitable to achieve high precision between synchronization points and to keep the communication overhead low. Linear regression is used in FTSP to compensate for clock drift as suggested in (Elson et al., 2002).
Typical WSN operate in areas larger than the broadcast range of a single node; therefore, the FTSP provides multi-hop synchronization. The root of the network, a dynamically elected single node, maintains the global time and all other nodes synchronize their clocks to that of the root. The nodes form an ad-hoc structure to transfer the global time from the root to all the nodes, as opposed to a fixed spanning-tree based approach proposed in (Ganeriwal et al.,

2003). This saves the initial phase of establishing the tree and is more robust against node and link failures and dynamic topology changes.

4.3.1 Time-stamping

The FTSP utilizes a radio broadcast to synchronize the possibly multiple receivers to the time provided by the sender of the radio message. The broadcasted message contains the sender's time stamp which is the estimated global time at the transmission of a given byte. The receivers obtain the corresponding local time from their respective local clocks at message reception. Consequently, one broadcast message provides a *synchronization point* (a global-local time pair) to each of the receivers (Maroti et al. 2004). The difference between the global and local time of a synchronization point estimates the clock offset of the receiver. As opposed to the RBS protocol, the time stamp of the sender must be embedded in the currently transmitted message. Therefore, the time-stamping on the sender side must be performed before the bytes containing the time stamp are transmitted.

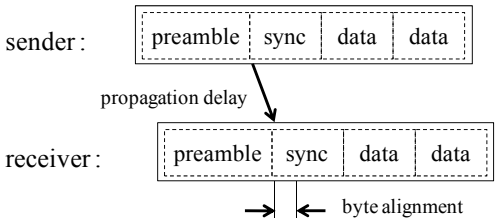


Fig. 7. Data packets transmitted over the radio channel. Solid lines represent the bytes of the buffer and the dashed lines are the bytes of packets (Maroti et al. 2004)

Message broadcast starts with the transmission of preamble bytes, followed by SYNC bytes, then with a message descriptor followed by the actual message data, and ends with CRC bytes. During the transmission of the preamble bytes the receiver radio synchronizes itself to the carrier frequency of the incoming signal. From the SYNC bytes the receiver can calculate the bit offset it needs to reassemble the message with the correct byte alignment. The message descriptor contains the target, the length of the data and other fields, such as the identifier of the application layer that needs to be notified on the receiver side. The CRC bytes are used to verify that the message was not corrupted. The message layout is summarized in Fig. 7.

The FTSP time-stamping effectively reduces the jitter of the interrupt handling and encoding/decoding times by recording multiple time stamps both on the sender and receiver sides. The time stamps are made at each byte boundary after the SYNC bytes as they are transmitted or received. First, these time stamps are normalized by subtracting an appropriate integer multiple of the nominal byte transmission time, the time it takes to transmit a byte. The jitter of interrupt handling time is mainly due to program sections disabling interrupts on the microcontroller for short amounts of time. This error is not Gaussian, but can be eliminated with high probability by taking the minimum of the normalized time stamps. The jitter of encoding and decoding time can be reduced by taking the average of these interrupt error corrected normalized time stamps. On the receiver side this final averaged time stamp must be further corrected by the byte alignment time that can be computed from the transmission speed and the bit offset (Maroti et al. 2004).

4.3.2 Clock drift management

If the local clocks had the exact same frequency and, hence, the offset of the local times were constant, a single synchronization point would be sufficient to synchronize two nodes. However, the frequency differences of the crystals used in Mica2 motes introduce drifts up to 40μs per second. This would mandate continuous re-synchronization with a period of less than one second to keep the error in the micro-second range, which is a significant overhead in terms of bandwidth and energy consumption (Maroti et al. 2004). Therefore, it is necessary to estimate the drift of the receiver clock with respect to the sender clock. The offset between the two clocks changes in a linear fashion provided the short term stability of the clocks is good. In this scheme, the stability of the 7.37 MHz Mica2 clock is verified by periodically sending a reference broadcast message that was received by two different motes. The two motes time-stamped the reference message using the FTSP time-stamping described in the previous section with their local time of arrival and reported the time-stamp (Maroti et al. 2004).

4.4 Tiny-Sync and Mini-Sync

Tiny-Sync and Mini-Sync are the two lightweight synchronization algorithms, proposed mainly for sensor networks, by Sichitiu and Veerarittiphan (Sichitiu & Veerarittiphan, 2003). The authors assume that each clock can be approximated by an oscillator with fixed frequency. As argued in previous section, two clocks, $C_1(t)$ and $C_2(t)$, can be linearly related under this assumption as:

$$C_1(t) = a_{12} \cdot C_2(t) + b_{12} \tag{7}$$

where a_{12} is the relative drift, and b_{12} is the relative offset between the two clocks. Both algorithms use the conventional two-way messaging scheme to estimate the relative drift and offset between the clocks of two nodes; node 1 sends a probe message to node 2, time stamped with t_o, the local time just before the message is sent. Node 2 generates a timestamp when it gets the message at t_b, and immediately sends back a reply message. Finally, node 1 generates a timestamp t_r when it gets this reply message. Using the absolute order between these timestamps and equation (7), the following inequalities can be obtained:

$$t_o < a_{12} \cdot t_b + b_{12} \tag{8}$$

$$t_r > a_{12} \cdot t_b + b_{12} \tag{9}$$

The 3-tuple of the timestamps (t_o, t_b, t_r) is called a "data point". Tiny-sync and mini-sync works with some set of data points, each collected by a two-way message exchange as explained. As the number of data points increases, the precision of the algorithms increases (Sichitiu & Veerarittiphan, 2003). Each data point corresponds to two constraints on the relative drift and relative offset (equations 8, 9). The constraints imposed by data points are depicted in Fig. 8. Note that the line corresponding to equation (9) must lie between the vertical intervals created by each data point. One of the dashed lines in Fig. 8 represent the steepest possible such line, satisfying equation (7). This line gives the upper bound for the relative drift (slope of the line, $\overline{a_{12}}$), and the lower bound for the relative offset (y-intercept

of the line, $\underline{b_{12}}$) between the two clocks. Similarly, the other dashed line gives the lower bound for relative drift ($\underline{a_{12}}$) and the upper bound for relative offset ($\overline{b_{12}}$). Then the relative drift a_{12} and the relative offset b_{12} can be bounded as:

$$\underline{a_{12}} \leq a_{12} \leq \overline{a_{12}} \tag{10}$$

$$\underline{b_{12}} \leq b_{12} \leq \overline{b_{12}} \tag{11}$$

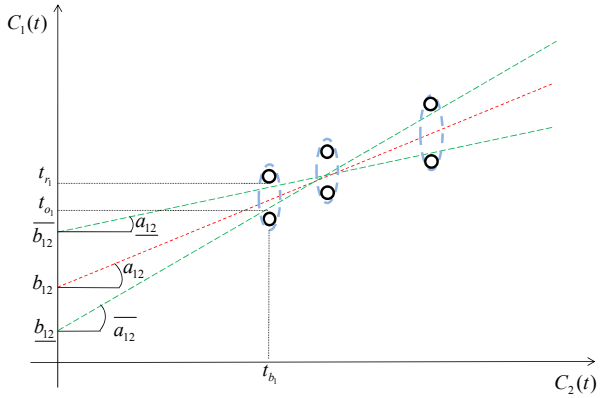


Fig. 8. The constraints imposed on a_{12} and b_{12} by data points (Sivrikaya & Yener, 2004)

The exact drift and offset values can not be determined by this method (or any other method - as long as message delays are unknown), but they can be well estimated. The tighter the bounds get, the higher the chance that the estimates will be good, i.e. the precision of synchronization will be higher. In order to tighten the bounds, one can solve the linear programming problem consisting of the constraints dictated by all data points in order to get the optimal bounds resulting from the data points. However, this approach is quite complex for sensor networks, since it requires high computation and storage for keeping all data points in memory (Sichitiu & Veerarittiphan, 2003; Sivrikaya & Yener, 2004).

The basic intuition behind *tiny-sync* and *mini-sync* algorithms is the observation that not all data points are useful. Consider, for example, the three data points in Fig. 8 the intervals $[\underline{a_{12}}, \overline{a_{12}}]$ and $[\underline{b_{12}}, \overline{b_{12}}]$ are only bounded by data points 1 and 3. Therefore data point 2 is useless in this example. Following this intuition, Tiny-sync keeps only the four constraints - the ones which yield the best bounds on the estimates- among all data points. The resulting algorithm is much simpler than solving a linear programming problem. However, the authors argue, by a counter example, that this scheme does not always give the optimal solution for the bounds: The algorithm may eliminate some data point, considering it useless, although it would actually give a better bound together with another data point that is yet to occur.

Mini-sync is an extension of Tiny-sync, such that it founds the optimal solution with an increase in complexity. The idea is to prevent the algorithm of tiny-sync for eliminating constraints that might be used by some future data points to give tighter bounds. We skip

the details here, but the authors basically define a criteria to determine if there is a chance that a constraint might be useful. A constraint is eliminated (discarded) only if it is *definitely useless*. The solutions found by Mini-sync are optimal (Sivrikaya & Yener, 2004).

5. Global Time Synchronization Algorithms

Li and Rus (Li & Rus, 2006) presented a high-level framework for global synchronization. The three methods are proposed for global synchronization in WSNs. The first two methods, all-node-based and cluster-based synchronization, use global information but are not suitable for large WSNs. In the third approach, diffusion method, each node sets its clock to the average clock time of its neighbors. The diffusion method thus converges to a global average value. A drawback of this approach is the potentially large number of messages exchanged between neighbor nodes, especially in dense networks.

5.1 All-Node-based Synchronization

This method is used on all the nodes in the system and it is most effective when the size of the sensor network is relatively small. In future sections of this paper, they describe ways to address scalability. They assume the clock cycle on each node is the same. They believe this is a reasonable assumption since most sensors are programmed with the same parameters prior to deployment. They also assume the clock tick time is much longer than the packet transmission time. Finally, they assume that the message transmission time over each link and handling time on each node are roughly the same. This time can be obtained when the network traffic is small. That is, upon its initial deployment, a sensor network allows sufficient time solely for clock synchronization. The key idea is to send a message along a loop and record the initial time and the end time of the message. Then, by using the message traveling time, they can average the time to different segments of the loop and smooth over the error of the clocks. Algorithm 1 (Li & Rus, 2006) summarizes this method.

Algorithm 1 All-Node-Based Synchronization Algorithms in a Sensor Network

1: Find a ring that passes each node at least once that need to be synchronized (suppose the ring is composed of k nodes)
2: A message is passed along the ring starting from an initiating node
3: Upon receipt of the message, each node records its current local time (t_i) and its order (i) in the ring. If the node receives messages more than once, it chooses one arbitrarily
4: After initiating node receives the message, it sends out another message informing each node on the ring the start time (t_s) and the end time (t_e) of the previous message
5: **for** each node, to adjust its local time t **do**
6: **if** $\exists\, m, m+1 \geq \frac{t_e - t_s + 1}{k} \cdot (i-1) \geq t_i \geq \frac{t_e - t_s}{k} \cdot (i-1) \geq m$ **then**
7: node n_i adjusts its time to $t - t_i + t_s + m$
8: **if** $\exists\, m, m+1 \geq \frac{t_e - t_s + 1}{k} \cdot (i-1) \geq t_i \geq \frac{t_e - t_s}{k} \cdot (i-1) \geq m-1$ **then**
9: node n_i adjusts its time to $t - t_i + t_s + m$

5.2 Cluster-based Synchronization

The synchronization method presented in Algorithm 1 has a provable bound, but it requires all the nodes to participate in one single synchronization session. This can be mitigated using a hierarchical approach. More specifically, if the network can be organized into clusters, we propose to synchronize the whole network using Algorithm 2 (Li & Rus, 2006). In Algorithm 2, the same method as in Algorithm 1 is firstly used to synchronize all the cluster heads by designing a message path that contains all the cluster heads (they are called the initiators base). Then, in the second step, the nodes in each cluster can be synchronized with their head.

Algorithm 2 The Cluster-Based Synchronization Algorithm

```
1: Run any clustering algorithm to organize the network into clusters
2: Synchronize the cluster heads with a base using Alg. 1
3: for each cluster do
4:  Synchronize the cluster members with the cluster head
```

This method can adapt to different clustering schemes. A cluster can be composed of the nodes within the transmission range of the cluster head; it can also be comprised of the nodes within some geographical area called a zone. For the first type of clustering, synchronization can be done with RBS. First, a reference broadcast is sent by the head to synchronize all the other cluster members. Then, any other node in the cluster sends out another reference broadcast to synchronize. The clock difference can be calculated with these two broadcasts and all the non head members can adjust their clocks according to the head's clock. In a zone clustering, the same method as Algorithm 1 is used to first design a cycle that includes all the nodes of the cluster and synchronize them all. The head of the cluster will be the initiator of the intra cluster synchronization (Li & Rus, 2006).

5.3 Diffusion-based Synchronization

The previous presented methods (cluster-based or all-node-based synchronization) use global time information sent to all the nodes and are not scalable for very large networks. The initiating node may encounter failure and, thus, the approach is not fault-tolerant. The nodes that participate in the synchronization must execute the related code approximately at the same time, which may be too hard in a large system. Now a diffusion method that is fully distributed and localized is introduced. In this method, synchronization is done locally, without a global synchronization initiator. It can also be done at arbitrary points in time as opposed to the strict timing requirements of the previous methods (Li & Rus, 2006).

The diffusion method achieves global synchronization by spreading the local synchronization information to the entire system. The algorithm can choose various global values to synchronize the network provided that each node in the network agrees to change its clock reading to the consensus value. An easy option is to choose the highest or lowest reading over the network. Synchronization to the highest or lowest value entails a simple algorithm (Li & Rus, 2006).

However, if there are faulty or malicious nodes, such a node may impose an abnormally high or low clock reading, which is likely to ruin the synchronization. To make the

algorithms more robust and reasonable, the following algorithms use the global average value as the ultimate synchronization clock reading. The main idea of the algorithms is to average all the clock time readings and set each clock to the average time. A node with high clock time reading diffuses that time value to its neighbors and levels down its clock time. A node with low time reading absorbs some of the values from its neighbors and increases its value. After a certain number of rounds of diffusion, the clock in each sensor will have the same value (Li & Rus, 2006).

There are two typical basic operations in diffusion-based synchronization scheme: 1) the neighboring nodes compare their clock readings at a certain time point and 2) the nodes change their clock accordingly. This, however, may be a problem because the clock comparison and the clock update cannot be done simultaneously (especially when clock comparison may take several steps). The clock updates based on the clock readings of the comparison time will be incorrect. The solution is to ask each node to keep a record of how much time elapses after the clock comparison on each node and use this time in the clock update (Li & Rus, 2006).

5.3.1 Synchronous Diffusion

Algorithm 3 (Li & Rus, 2006) shows the diffusion method. Synchronization between a sensor node and its neighbors is done by clock comparison and update operations. Because this algorithm only consider the time difference between two sensor nodes instead of the absolute clock time value, it is not required that all the sensors must do this local synchronization at the same time. In line 6, the exchanged value between sensor n_i and its neighbor n_j is proportional to the time difference between them.

Algorithm 3 Diffusion Algorithm to synchronize the whole network

1: **for** each sensor n_i in the network **do**
2: Exchange clock times with $n_i's$ neighbors
3: **for** each neighbor n_j **do**
4: Let the times of n_i and n_j be c_i and c_j
5: Change $n_j's$ time to $c_i + r_{i,j}(c_i - c_j)$
6: Change $n_i's$ time to $c_i - \sum r_{i,j}(c_i - c_j)$

5.3.2 Asynchronous Diffusion

In the previous section, a synchronous diffusion-based algorithm is presented. The synchronous algorithm is localized, but it requires a set order for all the node operations. In order to remove this constraint, the extension of the diffusion synchronization algorithm is here introduced. In this algorithm, all the nodes can perform operations in any order as long as each node is involved in the operations with nonzero probability. The following asynchronous averaging algorithm (Algorithm 4) (Li & Rus, 2006) gives a very simple average operation of a node over its neighbors. Each node tries to compute the local average

value directly by asking all its neighbors about their values; it then sends out the computed average value to all its neighbors so they can update their values.

Algorithm 4 Asynchronous Averaging Algorithm in a Sensor Network

1: **for** each sensor n_i with uniform probability **do**
2: Ask its neighbors the clock readings (read values from n_i and its neighbors)
3: Average the readings (compute)
4: Send back to the neighbors the new value (write values to n_i and its neighbors)

6. FAD(Fast Asynchronous Diffusion) Scheme

Several time synchronization algorithms have some problems when the algorithms escape their assumption and disconnection occurs in their network topology. For example, hierarchical topology has severe disadvantage when the network connection is broken. That is, all sensor nodes have to reorganize network connection and then time synchronization should be performed. Hence, asynchronous diffusion algorithm suggests new operation for global time synchronization among all the nodes in sensor networks.

$$C_{i-adjust} = \left(\frac{C_i + C_{j(neighbor)}}{2} \right) \tag{12}$$

In equation (12), $C_{i-adjust}$ presents an adjusted clock value, and $C_{j(neighbor)}$ is a clock value among neighbor sensor nodes. In asynchronous diffusion algorithm, a node n_i might have several clock values adjusted by algorithm 4 since all sensor nodes are assumed to be connected. In this case, a node n_i adjusts its local clock with the most recently received average clock value among a series of average clock values.

6.1 FAD Algorithm

Recently J. Bae and B. Moon (Bae & Moon, 2009) proposed a Fast Asynchronous Diffusion (FAD) clock synchronization algorithm in order to improve the diffusion-based asynchronous averaging algorithm (Algorithm 4). In this section, the different points about comparing asynchronous diffusion algorithm with the proposed FAD algorithm are presented. In asynchronous diffusion algorithm (Algorithm 4), each node uses the most recently received average clock value for adjusting its local clock when getting a series of average clock values. Meanwhile, the proposed scheme takes the mean of a series of average clock values received from all the neighbors under threshold for fast convergence. That is, a node adjusts its clock value with the mean of its neighbors' average clock values. Consequently the proposed algorithm (Algorithm 5) converges faster than asynchronous diffusion algorithm. The idea of FAD algorithm is expressed in equation (13).

$$C_{i-adjust} = \left(\frac{\sum_{j=1}^{N} \frac{C_i + C_{j(neighbor)}}{2}}{N} \right) \quad (13)$$

[N = number of neighbors]

FAD algorithm assumes that all the nodes in network have the same topology as asynchronous diffusion algorithm, but FAD algorithm differs from asynchronous diffusion in the process of getting average values. In other words, asynchronous diffusion scheme assumes that operating event must occurs in regular sequence, which uses average value received most recently. However, FAD algorithm doesn't consider operating sequence since it uses all the received average values (Bae & Moon, 2009).

Algorithm 5 Fast Asynchronous Diffusion(FAD) Algorithm in Sensor Network

1: **for** each node n_i with uniform probability **do**
2: Ask its neighbors the clock reading (read values from n_i and its neighbors)
3: **if** neighbor's clock < threshold
Average the reading(compute)
Send back to the neighbors the new value (write values to n_i and its neighbors)
4: **else** drop the received value
5: Each node n_i performs average operation again with all adjusted values received from its neighbors (write value to n_i)

In comparing FAD scheme with asynchronous diffusion scheme, there is actually no big difference in that more operations are required when the number of data increases in the viewpoint of algorithm complexity. But threre is an essential difference in the number of rounds needed for convergence. In next section, this difference is presented with the results of NS-2 simulation. Actually FAD has less number of rounds than asynchronous diffusion until convergence achievement is done. That is, FAD converges faster than asynchronous diffusion scheme. Generally, FAD seems to show less performce in the aspect of energy efficiency because FAD spends more time than asynchronous diffusion in getting average value. However, the time for synchronizing all the nodes in a sensor network is reduced since FAD achieves faster time synchronization than asynchronous diffusion scheme.

6.2 Performance Evaluation

The FAD scheme (Algorithm 5) is evaluated with NS-2 simulator (version 2.30) based on IEEE 802.15.4 module. The parameters such as propagation delay, collision, packet loss, and so on are considered. The simulation also includes asynchronous diffusion scheme for comparing FAD scheme with it. The simulation for time synchronization algorithms is performed within relative error of 0.01, and all nodes are assumed to have uniform distribution. The detail simulation parameters are summarized in Table 2.

Parameter	values
Number of Nodes	75, 90, 100, 125, 150, 200, 300, 400, 500
Sensor Field	100m × 100m
Transmission Range	15m
Physical Layer & MAC Layer	802.15.4
Routing Protocol	AODV
Relative Error	0.01
Uniform Probability (Mean)	0.5
Threshold (Percentage of Drift)	100%, 80%, 60%, 40%

Table 2. The Parameters for Simulation

6.3 Results and Discussions

In this simulation, the round is the number of the given algorithm performed at once. The number of operation is the sum of average operation from all nodes. In more detail, the operation ranges between zero and number of nodes participating in one round, and threshold is drift rate between received clock value and local clock value in one tick. Fig. 9 represents the comparison between asynchronous diffusion (left) and FAD (right) in the number of rounds with threshold value 40%. In this figure, the number of rounds decreases when the number of nodes increases. Each data point (*) represents the number of rounds when relative error becomes 0.01, and a line represents average value in each simulation condition.

Under this simulation, when the number of nodes is 500, FAD achieves time synchronization in average 31.7 rounds while asynchronous diffusion achieves it in average 35.8 rounds. When the number of sensor node is under 175, the time efficiency of FAD is better by 19% than asynchronous diffusion. When the number of sensor nodes is over 175, the time efficiency of FAD is better by 12% than asynchronous diffusion.

Fig. 10 shows the comparison between asynchronous diffusion (left) and FAD (right) with threshold value 40% in the number of operations. This figure represents that there is no big difference between FAD and asynchronous diffusion, and the number of total operations increases when the number of nodes increases. The reason can be explained from the results in Fig. 9. The number of rounds has exponential shape even thought the number of wireless sensor nodes increases. It means that these algorithms have to operate even though some additional rounds are not related with increasing the number of sensor nodes. That is, when the number of nodes is especially over the specific value, the number of rounds for time synchronization are not related with the number of nodes. Moreover, the number of operations increases when the number of nodes increases since the number of rounds is similar.

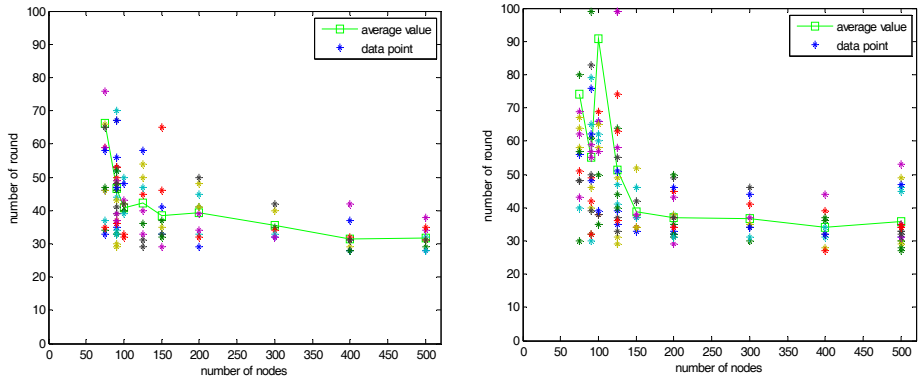


Fig. 9. Comparison between asynchronous diffusion (left) and FAD (right) in the number of rounds with threshold value 40%

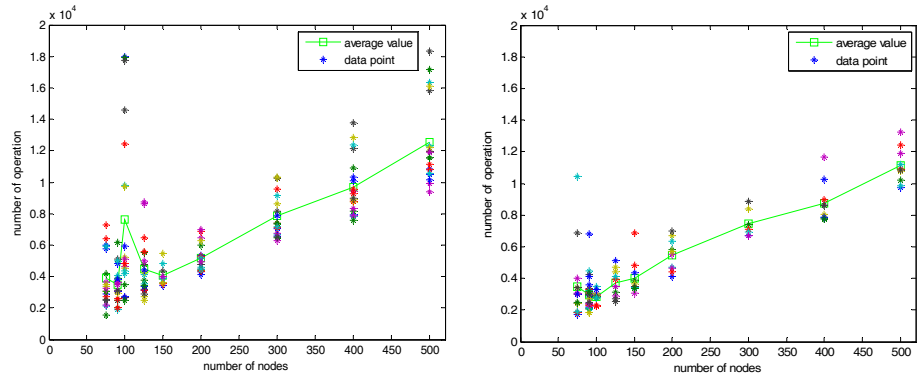


Fig. 10. Comparison between asynchronous diffusion (left) and FAD (right) with threshold value 40% in the number of operations

Fig. 11 represents the comparison between asynchronous diffusion and FAD in the average number of operations (left) and the average number of rounds (right) with threshold value(log scale). Fig. 11 (left) depicts the number of average operation in this simulation. Fig. 11 (right) shows average value of rounds. When the number of nodes is over 175, FAD uses the fewer number of operations than asynchronous diffusion. When the number of nodes is over 175, it is impossible for this simulation to compare FAD with asynchronous diffusion. However, when the number of nodes is under 175, FAD has better performance than asynchronous diffusion.

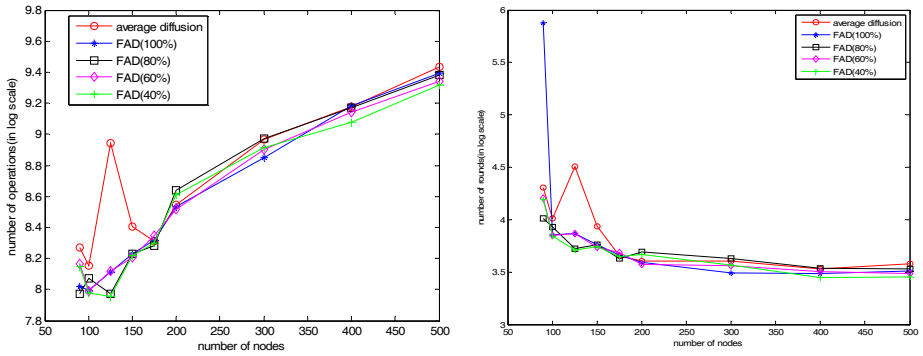


Fig. 11. Comparison between asynchronous diffusion and FAD in the average number of operations (left) and the average number of rounds (right) with threshold value(log scale)

7. Conclusion

Time synchronization is very useful function for improving device performance in WSN. In this chapter, we investigated time synchronization algorithms in WSN. Even though many algorithms are proposed until now, the best solution doesn't seem to exist since diversity devices are used in WSN. For the future research, meanwhile, time synchronization among heterogeneous devices will be new challenges.

8. References

Bae, J., & Moon, B. (2009). Time Synchronization with Fast Asynchronous Diffussion in Wireless Sensor Network, *International Conference on Cyber-enabled Distributed Computing and Knowledge Discovery (CyberC 2009)*, China, Zhangjiajie, October

Dolev, D.; Halpern, J., & Strong, H. R. (1984). On the Possibility and Impossibility of Achieving Clock Synchronization, *Proc. ACM Symp. Theory of Computing (STOC)*, May

Elson, J. & Estrin, D. (2001). Time Synchronization for Wireless Sensor Networks, *Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS),Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing*, San Francisco, California, USA, April

Elson, J.; Girod, L. & Estrin, D. (2002). Fine-Grained Network Time Synchronization Using Reference Broadcasts, *Proc. Fifth Symp. Operating System Design and Implementation(OSDI 2002),* Dec.

Ganeriwal, S.; Kumar, R. & Srivastava, M. (2003). Time Sync Protocol for Sensor Network, *The First ACM Conference on Embedded Networked Sensor System (SenSys),* Los Angeles, Nov., pp. 138–149.

Ganeriwal, S.; Pöpper, C., Čapkun, S. & Srivastava, M. (2008), Secure Time Synchronization in Sensor Networks, *ACM Transactions on Information and System Security (TISSEC)*, Vol.11, no.4, July, ISSN:1094-9224

Girod, L. & Estrin, D. (2001). Robust range estimation using acoustic and multimodal sensing, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, March

Halpern, J.; Simons, B. & Strong, R. (1984). Fault-Tolerant Clock Synchronization, *Proc. ACM Symp. Principles of Distributed Computing (PODC)*, Aug.

Hofmann-Wellenhof, B.; Lichtenegger, H., & Collins, J. (1997). Global Positioning System: Theory and Practice, *4th ed. Springer Verlag*.

Intanagonwiwat, C.; Govindan, R., Estrin, D., Heidemann, J. & Silva, F. (2003). Directed Diffusion for Wireless Sensor Networking, *IEEE Trans. Networking*, vol.11, no.1, pp.2–16, Feburuary

Kopetz, H., & Ochsenreiter, W. (1987). *Clock* Synchronization in Distributed Real-Time Systems, *IEEE Transactions on Computers*, C-36(8), p.933–939, August

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system, *Communications of the ACM*, vol.21, no.7, pp.558–565

Lamport L. & Melliar-Smith, P. M. (1985). Synchronizing Clocks in the Presence of Faults, *J. ACM*, vol.32, no.1, pp.52-78, Jan.

Levine, J. (1999). Time synchronization over the internet using an adaptive frequency-locked loop, *IEEE Trans. Ultrason., Ferroelectr., Freq. Contr.*, vol.46, no.4, pp.888–896, Jul.

Li, Q., & Rus, D. (2006). Global Clock Synchronization in Sensor Network, *IEEE Trans. Computer Society,* vol.55, pp.214-216, Feb.

Lundelius, J. & Lynch, N. (1984). A New Fault-Tolerant Algorithm for Clock Synchronization, *Proc. ACM Symp. Principles of Distributed Computing (PODC)*, pp. 75-88, Aug.

Mannermaa, J; Kalliomaki, K., Mansten, T. & Turunen, S. (1999). Timing performance of various GPS receivers, *Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum and the IEEE International Frequency Control Symposium*, pp.287–290, April

Maroti, M.; Kusy, B., Simon, G. & Ledeczi, A. (2004). The flooding time synchronization protocol, *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys'04),* ACM Press, New York, pp.39–49

Mills, D. L. (1991). Internet Time Synchronization: The Network Time Protocol, *IEEE Transactions on Communications*, COM 39, no.10, pp.1482-1493, October

Mills, D. L. (1998). Adaptive hybrid clock discipline algorithm for the network time protocol, *IEEE/ACM Transactions on Networking*, vol.6, no.5, pp.505–514, Oct.

Romer, K. (2003). Temporal Message Ordering in Wireless Sensor Networks, *IFIP MedHocNet*, Mahdia, Tunisia, June

Sichitiu, M. L., & Veerarittiphan, C. (2003). Simple, Accurate Time Synchronization for Wireless Sensor Networks, *IEEE Wireless Communications and Networking Conference (WCNC) 2003*, New Orleans, LA, USA, March, vol.2, pp.1266 – 1273

Simon, G.; Maroti, M., Ledeczi, A., Balogh, G., Kusy, B., Nadas, A., Pap, G., Sallai, J. & Frampton, K. (2004). Sensor network-based counter sniper system, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (Sen Sys),* ACM Press, New York

Sivrikaya, F. & Yener, B. (2004). Time Synchronization in Sensor Networks: A Survey, *IEEE Network*, vol.18, no.4, pp.45 – 50, July-Aug

Sommer, P. & Wattenhofer, R. (2009). Gradient clock synchronization in wireless sensor networks, *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pp. 37-48,

Su, W. & Akyildiz, I. F. (2005). Time-Diffusion Synchronization Protocol for Wireless Sensor Networks, *IEEE/ACM Transactions on Networking*, vol.13, no.2, pp.384–397, April

Woo, A., & Culler, D. (2001). A Transmission Control Scheme for Media Access in Sensor Networks, *International Conference on Mobile Computing and Networking, (Mobicom)*, pp. 221–235, July

Yoon, S.; Veerarittiphan, C. & Sichitiu, M. L. (2007). Tiny-sync: Tight time synchronization for wireless sensor networks, *ACM Transactions on Sensor Networks (TOSN)*, vol., no.2, June

16

Time Synchronization of Underwater Wireless Sensor Networks

Li Liu
Shandong University
P.R. China

1. Introduction

Large propagation delay and node movement are considered to be two significant attributes that differentiate an underwater wireless sensor network (UWSN) from a ground wireless sensor network (WSN). The acoustic-based media dramatically narrows the bandwidth of communication of UWSN due to slow propagation speed. An underwater sensor node can move out of and into another node's range frequently in an unstable underwater environment. In this chapter, the author will elaborate the study of investigating the property and the impact of these two attributes of UWSN. Then, this chapter will describe a prototype of a synchronization protocol which is suitable for UWSN considering the effects of both propagation delay and movement. With its protocol algorithm, no time synchronization is necessary if the time stamps of the received data packets are within the tolerance. In this fashion, the network underwater does not need to perform global time synchronizations frequently nor periodically, which reduces the time used to synchronize clocks among sensor nodes. Finally, this chapter will discuss simulation results which show the time cost for synchronization is linear to the data packets exchanged with this protocol.

2. Three Characteristics of UWSN Time Synchronization

2.1 Uncertain Interrelationship

The interrelationship among synchronizing parties are erratic since the underwater sensor nodes are not as stable as those on the ground due to undercurrents. In other words, underwater sensor nodes oscillate along with the jumbled waves all the time. The undetermined vertical movement is tremendously larger than the horizontal movement, and therefore this changes the topology after the network was deployed. This topology change affects the time synchronization because sensor nodes in the networks usually are synchronized with reference clock model, e.g., the Reference Broadcast Synchronization (RBS) (Elson et al., 2002). Therefore, each sensor node should know neighbors which are in its acoustic communication range and those which are not waiting for acknowledgement too long time and consuming too much power as Fig 1 shows. Node B may be thrown out from node A's acoustic range to position C in space. Once a neighbor sensor node is out of communication range, sensor A would stop trying to neither synchronize with it nor pass

data packets to it, e.g. B. When the Node B travels to another node's territory, such as position D in Fig 1, it could join another node E's data-collecting cluster.

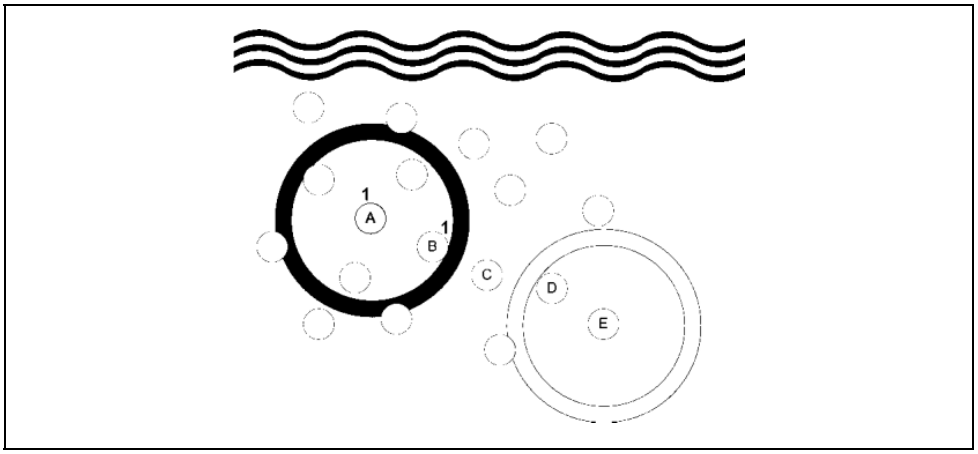


Fig. 1. Neighbour Node Uncertainty in UWSN

2.2 Synchronizing with New Node

Reliability, such as data accuracy of a new join-in node is another concern. Data collected for profiling and future analysis highly depend on a cluster of sensor nodes in a three dimensions space. High density of nodes gives better accuracy of environmental data (Xie et al., 2010). Vertical and horizontal waving undercurrents would bring a new sensor node into another sensor node's territory. However data provided by the new joining sensor to a cluster in the UWSN can be accepted only if the new node' clock is synchronized with the cluster. Therefore, there are more join-quit processes of nodes in an UWSN than those in a WSN on the ground because of the unstable issue.

Most sensor nodes in an UWSN are deployed by binding to ropes which are docked onto the bed of water or floater. The relative positions of sensor nodes are easily changed by the tension of the rope. On the other hand, the shape and weight of an underwater sensor affects the rope length caused by tension's change. Many other factors, such as temperature and mineralization of water, cause the degree of rope tension perplexed. The tension change of rope brings in uncertainty of nodes' positions mentioned above.

2.3 Propagation Delay

Third, due to the large propagation delay and low data bandwidth in an UWSN, beacon frame exchanged between two underwater sensor nodes should be simple and reduced in the total amount. Like most of radio frequency used on the ground, underwater acoustic signal also uses one channel for receiving and sending data (Kong, et al., 2005; Pompili, et al., 2006). Furthermore, propagation delay varies with many factors, e.g., the density and the purity of water, animal noise, etc.

A well-designed time synchronization scheme for underwater sensor network should be aiming to improve the synchronization process with careful consideration of these three issues.

In the rest of this chapter, the author will first talk about related background knowledge including the strategy of testing propagation delay for an UWSN, the network model, the clock model because there is little literature talking about the time synchronization issue in an underwater wireless sensor networks ever before. Then, this chapter uses an example time synchronization protocol of UWSN to illustrate how to design synchronizing algorithms which are suitable for underwater distributed systems. Overview of implementing the algorithm will also be presented. Finally, this chapter will show initial simulation results of the prototype protocol.

3. Background Knowledge

3.1 Test Strategy of Propagation Delay

Due to the uncertain factors listed above, it is difficult to find a reasonable constant value to compute the propagation delay since it varies when each of environment factors changes. Mathematics and mechanics analysis sometime are not able to depict right incidence to the UWSN. The best way to estimate the potential effect from environment is applying prototype trial measuring before deploying the whole UWSN underwater.

For the propagation delay and node mobility, we could have the following trial deployment to obtain the environmental parameters by measurements instead of estimations, e.g., the speed of acoustic and the swing amplitude of a node bound to rope, etc. Because the bottom end of a rope which ties up the sensor is anchored in a deeper position, the segment between fixed end and sensor end swung by undercurrent. The formula to calculate acoustic propagation speed is (1) in fresh water and (2) in sea water according to Kinsler's book (Kinsler, et al., 1982), respectively.

$$\begin{aligned} c(P,t) &= 1402.7 + 488t^2 - 482t^2 + 135t^3 \\ &\quad + (15.9 + 2.8t + 2.4t^2) \end{aligned} \tag{1}$$

where c = Speed of sound in meters/sec, P = Pressure in bars (1 bar = 100 kPa), and t = 0.01T where T is the temperature in Celsius.

$$\begin{aligned} c(D,S,t) &= 1449.05 + 45.7t - 5.21t^2 + 0.23t^3 \\ &\quad + (1.333 - 0.126t + 0.009t^2)(s-35) + \Delta(D) \end{aligned} \tag{2}$$

where $\Delta(D) \cong 16.3D + 0.18D^2$, at latitude 45 degrees in the oceans, where c = Speed of sound in meters/sec, t = T/10 Where T is the temperature in degrees Celsius, S = Salinity in parts per thousand, and D = Depth in kilometers.

For other latitudes in degrees, replace D with $D(1 - 0.0026\cos 2\varphi)$. This gives c with a standard deviation of 0.06 m/s down to a depth $D = 4\ km$ in oceanic waters.

A more complicated correction gives a standard deviation of 0.02 m/s:

$$\begin{aligned} (D,S) &= (16.23 + 0.253t)D + (0.213 - 0.1t)D^2 \\ &\quad + [0.016 + 0.0002(S-35)](S-35)tD \end{aligned} \tag{3}$$

Operators need to use precise device to get C (the speed of sound) and S (salinity in parts per thousand) in order to get an accurate value of acoustic propagation speed c in testing environment before deploying the whole WSN underwater since it is impossible for sensors to get accurate speed themselves.

To obtain the real value of swing amplitude which determines the mobility of node, operators could use camera to record the trail deployed sensors for a certain period of time and analyze the maximum, minimum and average swing amplitudes angle $\theta_{\max}$, $\theta_{\min}$, θ_{avg} with the help of image processing technique. Then, the maximum, the minimum and the average swing amplitudes can be calculated by formula set below.

$$d_{horizontal} = \sin\theta \cdot l \text{ and } d_{vertical} = (1 - \sin\theta) \cdot l \tag{4}$$

where l is the length between node and fixed point at the bottom.

After carrying out the trial deployment, we could get the real environmental specification. Therefore, an underwater sensor needn't try to censoring these parameters because the sensor cannot get accurate value of some parameters in statistics, e.g., the swing amplitude. An operator could assign the creditable data into formula for later computation in the sensors, instead.

3.2 Network Model

An UWSN is a dense network consisting of a large number of resource-constrained sensor nodes with neither reference nodes nor a root node. This is a realistic deployment scenario in that a WSN is inherently infrastructure-less (Hu, et al., 2008) where many sensors autonomously organize themselves into a connected structure. Thus, it is often desirable to minimize the dependency of time-synchronization on infrastructure nodes. Each node maintains a sufficient number of neighbors to accelerate the synchronization process. The number of neighbors (undercurrent moving) can be easily adjusted by changing the transmission power when the synchronization information is broadcasted. A bidirectional neighbor relationship is not needed in this scheme. For further reduction of the synchronization overhead each node piggybacks the synchronization information on beacon messages that are periodically broadcast to refresh each node's neighbor list. In the current work, we assume that there are some reliable broadcast techniques such as (Tang, et al., 2001) are used.

3.3 Clock Model

Each sensor node has its own physical clock, calculated by counting pulses of its hardware oscillator running at a particular frequency. In practice, sensors' oscillators run at slightly different frequencies and the frequency varies unpredictably, depending on ambient factors such as temperature and humidity. Hence, sensors' clocks are subject to a divergence or clock offset. Based on Sichitiu's paper (Sichitiu, et al., 2003), for a relatively extended period of time (minutes to hours), the clock can be approximated with good accuracy by an oscillator of fixed frequency. The local clock of a sensor node i can thus be approximated (Lamport, et al., 1985) as

$$T_i(t) = \alpha_i t + \beta_i \tag{5}$$

where t is the physical time like UTC, α_i is the drift rate (frequency)of *i*, and β_i is the offset between the local clock and the physical time.
Using equation (5), we can compare the local clocks of two nodes in a network, say node 1 and node 2 as:

$$C_1(t) = \alpha_{12} C_2(t) + \beta_{12} \tag{6}$$

We call α_{12} the relative drift, and β_{12} the relative offset between the clocks of node 1 and node 2. If two clocks are perfectly synchronized, then their relative drift is 1--meaning that the clocks have the same rate- and their relative offset is zero--meaning that they have the same value at that instant.

4. An Example Protocol Algorithm

The example synchronization protocol is based on the Interactive Convergence Time Synchronization (ICTS) algorithm similar to the one in (Lamport, et al., 1985). In ICTS, the network-wide synchronization is achieved by having each node first derive the time offsets between itself and all of its neighbors by exchanging messages. Each node then computes the average of the measured clock offset and uses it to adjust its own clock. As long as less than one third (half) of neighbor nodes are mis-behaving with Byzantine (non- Byzantine) faults, all the sensor nodes in the neighborhood will establish a common equilibrium time.

4.1 Time offset

The protocol applies the single message broadcast method which is used in FTSP (Mar´oti, et al., 2004) to compute the offset between two nodes. FTSP successfully eliminates major sources of uncertainties in the packet recommission (i.e., transmission time, access time, reception time, jitter of interrupt-handling and encoding/decoding time) by performing MAC-layer timestamping multiple times for every message at each byte boundary and embeds a final error-corrected and averaged timestamp into the message. The only uncertainty is the propagation time (for packets to traverse the wireless link) which is often very small and can be safely ignored. According to Mar´oti's findings (Mar´oti, et al., 2004), using only 6 timestamps per message, FTSP achieves the time stamping accuracy of 1.4 μs on the Mica2 platform. Thus, one radio broadcast is sufficient for all the neighbors to accurately calculate the time offsets between their clocks and sender's clocks, each of which is simply the difference between transmission and reception timestamps.

4.2 ICTS with Propagation Delay

Let s be the sensor node performing time-synchronization and n_s is the number of S's neighbors. T_i and T_s represent the send and receive timestamps. $p_{i,s}$ is the propagation delay when message leave node i until reach node s. Node S can then calculate the time offset between itself and node i as $\Delta_{s,i} = T_s - T_i - p_{i,s}$. After obtaining the equation for i = 1…ns from all of its neighbors, s computes its new clock value at time t or $T_s^{'}(t)$ as:

$$
\begin{aligned}
T_s^{'}(t) &= T_s(t) + \frac{1}{n_s+1}\sum_{i=1}^{n_s}\Delta_{s,i} \\
&= T_s(t) + \frac{1}{n_s+1}\sum_{i=1}^{n_s}(T_s - T_i - p_{i,s}) \\
&= T_s(t) + \frac{1}{n_s+1}\sum_{i=1}^{n_s}(T_s - T_i) - \sum_{i=1}^{n_s} p_{i,s}
\end{aligned} \tag{7}
$$

We could figure out the value of $\sum_{i=1}^{n_s} p_{i,s}$ with the strategy in Section III since the limit of $\sum_{i=1}^{n_s} p_{i,s}$ is the expected value of propagation delay.

The denominator $n_s + 1$ comes from the fact that node S's own clock is also considered for the computation of a new clock value.

Sensors terminate the initial synchronization when the local clock gets stabilized (i.e., $\left|T_s^{'}(t) - T_s(t)\right| < \varepsilon$, ε is a predefined parameter determining the synchronization accuracy). However, synchronization at a single point is insufficient, as the discrepancies in clock drift rate of different sensors will cause nodes to go out of synchronization after a short period of time. Thus, to maintain an acceptable accuracy, it is necessary to periodically execute ICTS for resynchronization, shown in Fig. 2. The appropriate resynchronization interval can be determined by the bound of the time offset and the maximum relative drift rate among sensor nodes (Sivrikaya, et al., 2004).

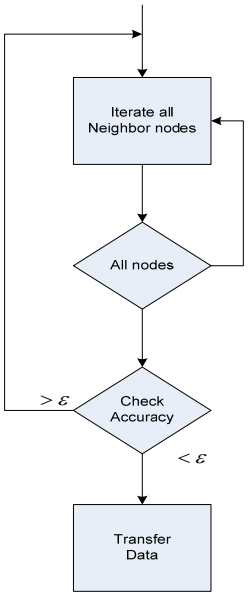


Fig. 2. Clock resynchronization

4.3 Data Packets

The neighbor sensors oscillate in the range and at the border of a node if the node is regard as a "sink point" in its territory. Limited by single acoustic channel and lower bandwidth the sink node cannot request a resynchronization with all the nodes once a neighbor sensor shifts over the acoustic range no matter a neighbor node is carried away from or brought in the sink node's territory by undercurrent or ocean wave. We could design a protocol which timestamps each data message. The timestamp will help the profile manager (introduced later) to determine if the data is confidential to be used or not. We could use the relative drift rate between the local clocks of nodes s and i which is defined as $\alpha_{s,i} = \alpha_i / \alpha_s$ to judge data confidence since there is no reference to the physical time, a node's drift rate (e.g., α_i or α_s) cannot be directly measured.

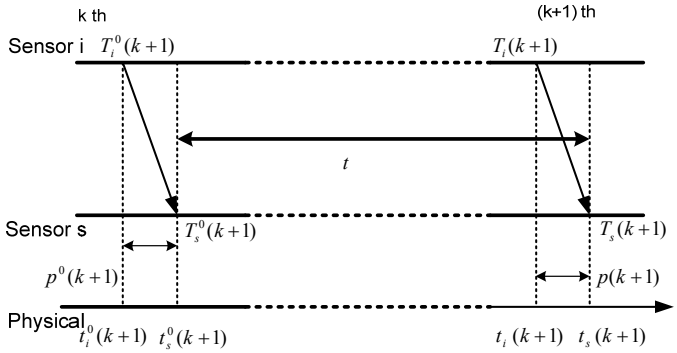


Fig. 3. ICTP with propagation

The example protocol scheme derives the relative drift rate indirectly as follows. Assume that each sensor performs synchronization periodically. In Fig. 3, $T_i(k)$ and $T_s(k)$ are the MAC-layer timestamps that record the sending time and the receive time, respectively, of the data message at the k-th iteration where k = 1, 2, … Let $t_i(k)$ and $t_s(k)$ denote the physical times corresponding to $T_i(k)$ and $T_s(k)$. Assume that sensor i and s finish their next data from $t_i(k+1)$ till $t_s(k+1)$ on physical clock. Their local clock readings of the two times are $T_i(k+1)$ and $T_s(k+1)$, respectively. The propagation delay in these two processes are $p(k)$ and $p(k+1)$ as well. t is the time between two iterations, $t = t_s(k+1) - t_s(k)$. We can express t in terms of local clocks:

$$\begin{aligned} & T_s(k+1) - T_s^0(k+1) \\ &= \alpha_s t_s(k+1) + \beta_s(k+1) - \alpha_s t_s^0(k+1) - \beta_s(k+1) \\ &= \alpha_s(t_s(k+1) - t_s^0(k+1)) \\ &= \alpha_s t \end{aligned} \tag{8}$$

and

$$
\begin{aligned}
&T_i(k+1) - T_i^0(k+1) \\
&= \alpha_i t_i(k+1) + \beta_i(k+1) - \alpha_i t_i^0(k+1) - \beta_i(k+1) \\
&= \alpha_i (t_s(k+1) - p(k+1)) + \beta_i(k+1) \\
&\quad - \alpha_i (t_s^0(k+1) - p^0(k+1)) - \beta_i(k+1) \\
&= \alpha_i (t_s(k+1) - t_s^0(k+1)) + (-p\ (k+1) + p^0(k+1)) \\
&= \alpha_i t
\end{aligned}
\tag{9}
$$

Therefore, the relative drift rate $\alpha_{s,i}$ can be derived by formula (10) with timestamps of packet inside the UWSN. We do not need to care about physical time outside.

$$
\alpha_{s,i}(k) = \frac{\alpha_i}{\alpha_s} = \frac{\alpha_i t}{\alpha_s t} = \frac{T_i(k+1) - T_i^0(k+1)}{T_s(k+1) - T_s^0(k+1)} \tag{10}
$$

4.4 Profiling Synchronization

As mentioned in the introduction, a sensor, which is brought into another sensor's territory by the undercurrent, should be examined the clock first to guarantee that data provided by this sensor has a confidential clock, that is a right relative clock drift to the existing cluster. The protocol creates a profile manager whose function is to maintain a history profile recording relative clock drift between node s and all its neighbor nodes and the nodes who have been its neighbors before. Profile manager (PM) establishes one history profile copy

$$
\left\{\alpha_{s,i}(k)\right\}_{k-q}^{k} = \left\{\alpha_{s,i}(k-q), \alpha_{s,i}(k-q+1), \cdots, \alpha_{s,i}(k)\right\}
$$

for each neighbor node i's last q relative clock drift with node s by the k-th iteration. $\left\{\alpha_{s,i}(k)\right\}_{k-q}^{k}$ exhibits a strong temporal correlation, as they represent the quality of neighbors' clocks and are updated at each iteration. Profile manager calculates a mean value μ for each profile copy with discrete or continuous probability distributions depending on the number of messages which the neighbor nodes provided. For discrete probability distributions, the protocol uses variance to compute μ, for continuous probability distributions, and we could use normal distribution to generate μ which is the location in Gaussian distribution.

With the value μ profile manager, check the timestamp of every data message provided by its neighbor. If

$$
\left(\alpha_{s,i}(k) - \mu\right)^2 < \lambda \tag{11}
$$

in discrete probability distributions and

$$
\left|\alpha_{s,i}(k) - \mu\right| < \lambda \tag{12}
$$

in continuous probability distributions, the profile manager treats the message as a confidential data message and buffers the data, if not, the data will be dropped because of untrusting. λ is a predefined accuracy value.

The profile manager (PM) will also help decide the resynchronization interval for a particular sensor cluster. As we discussed above, the confidence of data provided by neighbor nodes settle on whether the data packet could be accepted by the existing sensor cluster, a subsystem of the whole underwater network. In overall view, higher acceptance rate stands for higher utilization of censured data. If most of sampled data packets are dropped due to accuracy requirement λ, it does not reduce the utilization of censuring data but also dries out power supply since underwater is more energy consuming. The criterion of switching the node's mode from transferring data to resynchronization is determined by the data packet acceptance rate. Profile manager creates a global table called Global Confidential Table (GCT) aiming to record the accept data packet ratio. The GCT is a one dimension fixed size table which marks "1" standing for acceptance of data packet. Default value is "0" which means the packet does not meet the λ requirement. The protocol defines a threshold R as the number of acceptance data packets in GCT, shown in Fig 4. If ratio of acceptance data packets to table size is below R the profile manager will stop the node receiving data and start resynchronization until local clock accuracy reaches requirement formula (4) and (5). The upper GCT in Fig. 4 shows that the ratio is higher than the threshold and the lower one means that the cluster needs to be resynchronized.

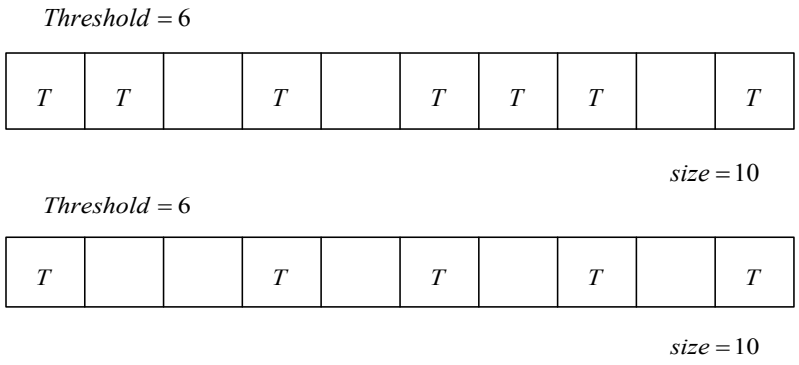


Fig. 4. Global Confidential Table

The whole process flow is shown in Fig 5.
Because there is no bidirectional neighbor relationship for every two nodes, each node maintains the relative clock drift in its own acoustic range, a cluster for profiling data. On the other hand, adjacent sensors' clusters must have overlap. The overlap plays the role to keep the whole relative clock as close as possible to a unique value. Therefore, the whole network stays in a low relative clock drift level with the help of profile manager and frequent resynchronization.

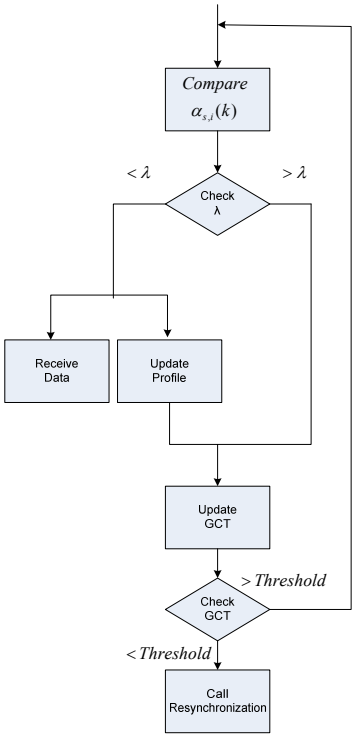


Fig. 5. Shift between sending data and time synchronization

5. The Effect of Undercurrent to Synchronization

The mobility of each node in an UWSN brings unfastened neighbor problem to a data profiling cluster. Sensors are deployed in different layers in an open space underwater. If we clip the space out from the whole by outmost sensors' furthest audio reachable range in one data profiling cluster the clipped space could be likened to a rubber balloon filled with water. The shape is easily changed when pressure comes outside. The pressure to the data profiling space in real world is undercurrent. Water moves along with many factors e.g., wind on the ocean surface, earth's rotation, etc., to unpredicted orientations. That is to say, if we research the synchronization of UWSN, we could not dismiss the high mobility even the sensors are anchored relative stable.

The second characteristic of the network underwater is that we cannot treat sensors underwater as 2 dimensions plane layout. Research on wireless sensor network above the ground usually assumes that the network is deployed onto the controlled environment without thinking too much about the latitude value. That is to say, the horizontal distance between two nodes above the ground plays more important role in research work on attributions of wireless sensor network above the ground. However, the network underwater exists in a real 3-dimension world. The vertical movement is as important as the horizontal movement when nodes are in a fluid environment. We need to use cube or sphere to describe the behavior of a node underwater instead of rectangle or circle in plane.

6. Simulations

The simulation consists by two sub phases. In the first part, we simulate the time synchronization with the traditional ICTP protocol running on our test case. Then, we simulate the example algorithm considering the effect of movement of UWSN. The profile manager (PM) took participate in this phase working abovementioned.
As the reason this chapter discussed in Section 2, the simulation use a trail deployment of sensors to measure the environmental factors. It is assumed that the real acoustic speed could be tested by professional device and calculated by. For simplicity, this simulation uses the mean value of acoustic, 1500 m/s as simulation parameter. Other parameters are shown in Table 1.

Parameter Name	**Value**
Simulation Radius	100 m
Acoustic range	35 m
Acoustic speed	1500 m/s
Sensor clock drift	± 0.3 ms/sec.
Initial clock offset	±1.0 ms
Threshold of accuracy	350 μs

Table 1. Parameters configuration

6.1 Synchronization of ICTP with propagation delay

The simulation deployed 30 sensor nodes in a cube whose side length is 100m. Every dimension of each node position is assigned randomly by a pseudo random number generator. Therefore, nodes are independent in spatial relationship. Fig 6 gives a node deployment scenario.

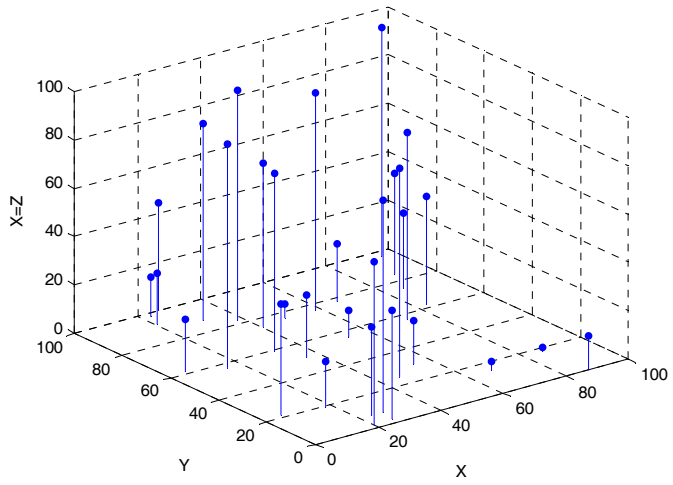


Fig. 6. Sensor nodes in 3D view

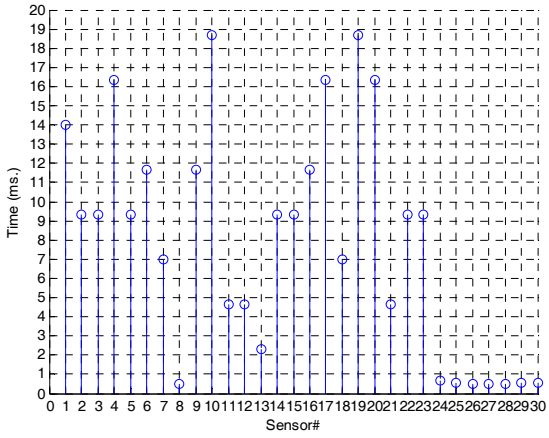


Fig.7. Time cost for each node in one UWSN

Fig 7. shows the time cost of the 30 sensors sending 100 data packets to all their neighbor nodes with ICTP synchronization method. We can find that the time cost varies due to different relative clock drift and offset of a node and its neighbor node(s).

6.2 Simulation Result of UWSN Synchronization Protocol

As it is described in previous paragraphs, the propagation delay of UWSN is 4 times bigger than transmission. Based on the observation strategy in Section 3, the simulation approximate the relationship between propagation delay and packet transmission to an integer multiple. First, we simulate the time cost that a node sends 100 data packets to all its neighbor sensor nodes when propagation delay is four times of transmission time.

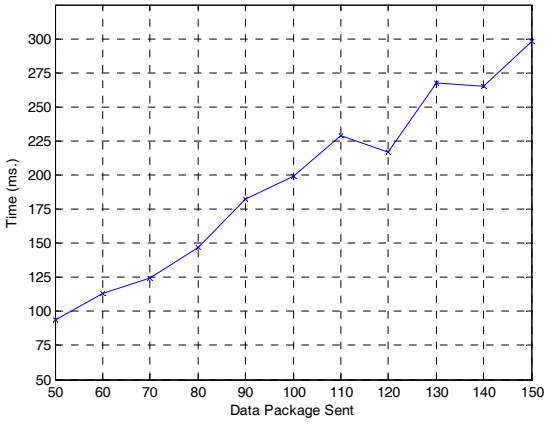


Fig. 8. 30 nodes send different number of packets when propagation delay is four times of transmission time.

Fig 8 shows the time cost curve. The total time cost goes up with total amount of data packets to be sent. Then, we add 5 more nodes to the space to structure a new network underwater.

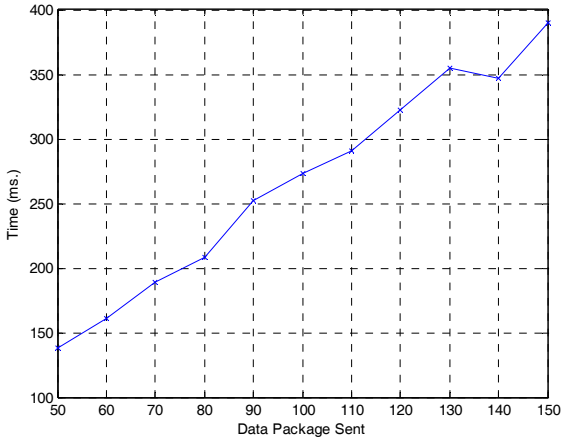


Fig. 9. 35 nodes send different number of packets when propagation delay is four times of transmission time.

Fig 9. shows the result that 35 nodes send 100 data packets to their neighbor node(s) when the propagation delay is four times of transmission time in the ICTP synchronization protocol. The time cost goes up almost the same as it goes up in the previous structure. Then the simulation deploys another five sensors into the network. There is nothing quite different but the starting point and ending point both shifted up for 50 ms in Fig 10.

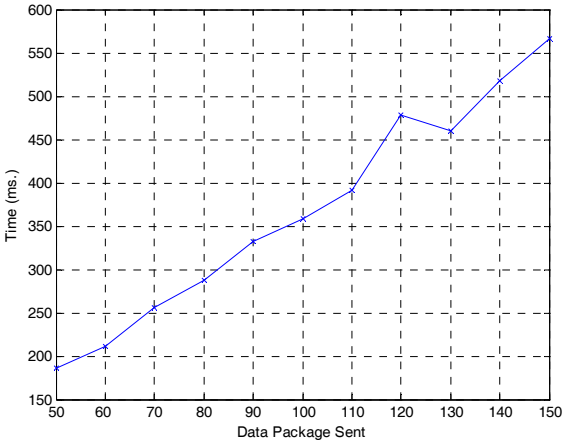


Fig. 10. 40 nodes send different number of packets when propagation delay is four times of transmission time.

To combine these three curves, result is in Fig 11.

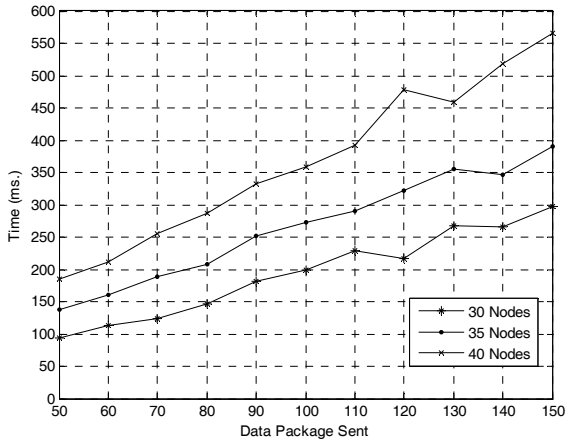


Fig. 11. 30, 35 40 nodes send different number of packets when propagation delay is four times to transmission time.

Next, simulation obtains the characteristic when propagation delay is five times to transmission time in a 30 nodes UWSN.

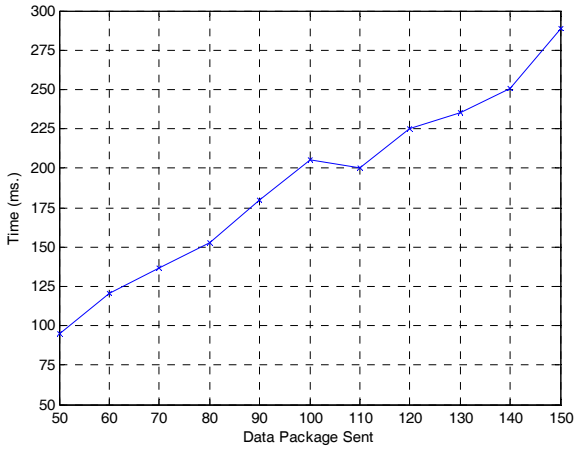


Fig. 12. 30 nodes send different number of packets when propagation delay is five times to transmission time.

In Fig 12, the total time cost increase along with the packet amount almost in the same way when the propagation delay is only four times of the transmission. Readers can compare the two curves in one chart shown in Fig 13.

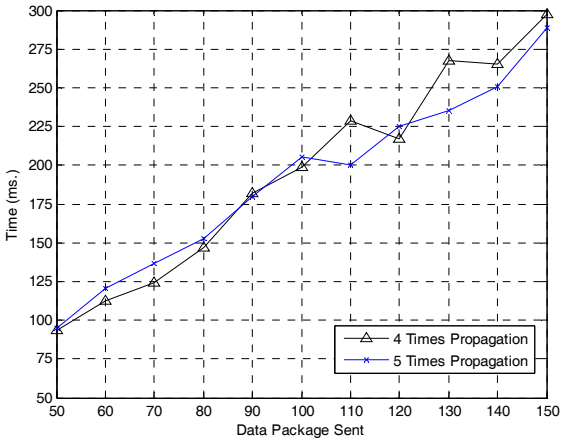


Fig. 13. 30 nodes send different number of packets when propagation delay is four or five times of transmission time.

7. Conclusion

In this chapter, we review those factors that are essential to the design of a new time synchronization protocol for an Underwater Wireless Sensor Netwrok (UWSN). We use a linear synchronization algorithm as an example to show these key points of proposing new protocols. The effect of large propagation delay of acoustic media in communication is addressed in simulating the demo prototype protocol. The simulation results demonstrate the difference of an UWSN time synchronization protocol by applying the new design pattern and by using the classical method. Simulation results also suggest the relationship between network performance and related factors.

8. References

Elson, J. E.; Girod, L. & Estrin, D. (2002). Fine-Grained Network Time Synchronization using Reference Broadcasts, *Proceedings of The Fifth Symposium on Operating Systems Design and Implementation,* pp. 147–163, ISBN 978-1-4503-0111-4, Boston, MA, USA, December 2002, New York, NY, USA

Hu, X.; Park,T. & Shin, K. G. (2008). Attack-tolerant time-synchronization in wireless sensor networks, *Proceedings of INFOCOM 2008*, pp. 41-45, ISBN 978-1-4244-2025-4, Phoenix,AZ, USA, April 2008, IEEE, Piscataway, NJ, USA

Kinsler, L.; Frey, A.; Coppens, A. & Sanders, J. (1982). *Fundamentals of Acoustics*, John Wiley & Sons, ISBN-10: 0471029335, New York, NY, USA

Kong, J.; Cui, J.; Wu, D.; & Gerla, M. (2005). Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications, *Proceedings of Military Communication Conference 2005*, pp. 1-7, ISBN 978-0-7803-9393-6, Atlantic City, NJ, USA, October 2005, IEEE, Piscataway, NJ, USA

Lamport, L. & Melliar-Smith, P. (1985). Synchronizing clocks in the presence of faults. *Journal of the Association for Computing Machinery*, Vol. 32, No. 1, (1985) 52–78, ISSN 0004-5411

Mar´oti, M.; Kusy, B.; Simon, G. & L´edeczi, A. (2004). The flooding time synchronization protocol, *Proceedings. of SenSys 2004,* pp. 39-49, ISBN 1-58113-879-2, Baltimore, MD, USA, November 2004, ACM Press, New York, NY, USA

Pompili, D.; Melodia, T. & Akyildiz, I. F. (2006). Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks, *Proceedings of The 12th Annual International Conference on Mobile Computing and Networking*, pp. 298-310, ISBN 1-59593-286-0, Los Angeles, CA, USA, September 2006, ACM Press, New York, NY, USA

Sichitiu M. L. & Veerarittiphan, C. (2003). Simple, accurate time synchronization for wireless sensor networks. *Proceeding of IEEE Wireless Communications and Networking* 2003, pp. 1266-1273 ISBN 1525-3511, New Orleans, LA, USA, March 2003, IEEE, Piscataway, NJ, USA

Sivrikaya, F. & Yener, B. (2004). Time synchronization in sensor networks: a survey, *IEEE Network Magazine's special issue on" Ad Hoc Networking: Data Communications & Topology Control,* Vol. 18, No. 4, (2004) 45-50, ISSN 0890-8044

Tang, K. & Gerla, M. (2001). Mac reliable broadcast in ad hoc networks. *Proceedings of IEEE Military Communication Conference 2001*, pp. 1008-1013, ISBN 0-7803-7225-5, Vienna, VA, USA,October 2001, Piscataway, NJ, USA

Xie, P.; Zhou, Z.; Peng, Z.; Cui, J. & Shi, Z. (2010). SDRT: a reliable data transport protocol for underwater sensor networks. *Ad Hoc Networks*, Vol. 2, No. 003, (2010) 1-15, ISSN 1570-8705

Part 4

Security

17

Security of Wireless Sensor Networks: Current Status and Key Issues

Chun-Ta Li
Department of Information Management, Tainan University of Technology
Taiwan

1. Introduction

Due to significant advances in wireless and mobile communication techniques and the broad development of potential applications, Wireless Sensor Networks (WSNs) have attracted great attention in recent years. Nevertheless, WSNs are formed dynamically by a number of power-limited sensor nodes and the manager node with long-lasting power. WSNs are self-organized and autonomous systems consisting of common sensors, manager nodes and back-end data center. Firstly, the common sensors are responsible for transmitting the real-time sensor data of specific monitoring environment to the intermediate collection nodes called manager node. Finally, the back-end data center will receive the sensed data from manager nodes to do further process and analysis. Undoubtedly, all communication between nodes are through the wireless transmission techniques. Furthermore, due to the property of self-organized, without support from the fixed infrastructure and the topology of wireless sensor network changes dynamically, therefore, broadcasting is the general way for communications in WSNs.
Wireless sensor network has been widely used in practical applications, such as monitoring of forest fire, detection of military purpose, medical or science areas and even in our home life. However, WSNs are easily compromised by attackers due to wireless communications use a broadcast transmission medium and their lack of tamper resistance. Therefore, an attacker can eavesdrop on all traffic, inject malicious packets, replay older messages, or compromise a sensor node. Generally, sensor nodes are most worried about two major security issues, which are privacy preserving and node authentication. Privacy means the data confidentiality is achieved under security mechanism, and hence it allows network communications between sensor nodes and the manager station to proceed securely. In addition, a well-structured authentication mechanism can ensure that no unauthorized node is able to fraudulently participate and get sensitive information from WSNs. As a result, several schemes have been proposed to secure communications in WSNs. In this chapter, we classify them into three classifications based on the cryptographic techniques: symmetric keys, asymmetric keys and one-way hashing functions.
The rest of this chapter is organized as follows: In Section 2, we introduce the characteristics and consideration of WSNs. In Section 3, we review some security threats and requirements in WSNs. Section 4 is for the security countermeasure schemes and its classification. Finally, we conclude some future works for the secure networking in WSNs.

2. Wireless Sensor Network

Compared with the traditional communication networks, some characteristics and considerations for wireless sensor networks are discussed and addressed in the design of WSNs. These are briefly reviewed in this section.

2.1 Characteristics of Wireless Sensor Network

- **Non-centralized architecture**: In WSNs, the status of every node is identical and no one is responsible for providing normal services. It is lack of a central administration and every node can join or disjoin the network any time. Besides, it does not affect the whole sensor network if some node failed and is reliable for applications with high stable requirement.
- **Self-organized**: Because WSNs are characterized as infrastructure-less networks and lack of fixed infrastructure. Thus, the sensor network is fully constructed by themselves when it is begin working with some pre-defined layering protocols and distributed algorithms. Once sensor networks are constructed completely, the sensor data would be collect and send to back-end system for further processing through the networks they built.
- **Multi-hop routing**: The sensor range of nodes in the WSNs is assumed to be limited, so if a node A would like to communicate with node D, which is out of communication range of node A. The node B would be a intermediate node and is responsible for transmitting the communication data to each other between node A and node B. The multi-hops is illustrated as Figure 1.
- **Dynamic topology**: In most of sensor network architecture assume that sensor nodes are deployed randomly and the network topology would be changed dynamically since the sensor node might be shut down, crash, recovery or utilize mobile sensors.

2.2 Consideration of Wireless Sensor Networks

- **Hardware constraints**: This part is related to physical property and many constraints on these areas have been proposed. For example, limited energy. In addition, due to the influence of limited volume of the sensor, some sensor can only provide limited storage, limited bandwidth, limited energy and limited computation ability.
- **Communication**: The existing communicating schemes show that there are three main types of communications in WSNs; including direct, clustering-based, and multi-hops communication. In direct communication, every sensor node transmits its sensor data to a manager node and the manager node is responsible for collecting these data to back-end data center for further processing. In clustering communication, all sensor nodes are divided into several groups and each cluster head node is responsible for collecting data within its group. Multi-hops communication is used because the communication range of a sensor is assumed to be limited and the neighboring sensor nodes maybe used for transmitting the communication packets to each other on their path between the source node and the destination node.
- **Scalability**: Another consideration is the scalability of sensor networks. In this case, networking must keep on working whatever the number of sensor nodes are placed will not be affected.

- **Fault tolerance**: Due to the influence of applied environment on sensors, many exceptions have been addressed in sensor networks. For example, sensors may crash, power failure or shut down etc. Such problems need to be avoided by the strategies of fault tolerance to keep on networking.
- **Power saving**: When the sensors are distributed to monitor some environments of interest, these sensors may work over a long span of several weeks even for months. Therefore, how to provide a mechanism of power saving to extend its lifespan is highly important. In general, there's too great a consumption of power during the transmitting message phase.
- **Cost**: Depending on the application of sensor network, a large number sensors might be scattered randomly over an environment, such as weather monitoring. If the overall cost was appropriate for sensor networks and it will be more acceptable and successful to users which need careful consideration.
- **Mobility**: In clustered (hierarchical) WSNs, sensor nodes are typically organized into many clusters, with cluster controllers collecting sense data from ordinary sensor nodes in the managed cluster to the back-end data center. Furthermore, compared to mobile ad hoc networks, when sensor nodes are randomly deployed in a designated area, they only infrequently move from one cluster to another, and thus mobility is not a critical issue in WSNs.
- **Sleep pattern**: The sleep pattern is highly necessary in WSNs to extend the availability of the networks. For example, the manager node can set fresh bootstrapping times for live sensors while other sensor nodes can shut down to save power. Different sensor nodes are operated according to the bootstrapping times to which they belong and the lifetime of WSNs is therefore extended in a differentiated way (23).
- **Security**: One of the challenges in WSNs is to provide high-security requirements with constrained resources. The security requirements in WSNs are comprised of node authentication, data confidentiality, anti-compromise and resilience against traffic analysis. To identify both trustworthy and unreliable nodes from a security standpoints, the deployment sensors must pass an node authentication examination by their corresponding manager nodes or cluster heads and unauthorized nodes can be isolated from WSNs during the node authentication procedure. Similarly, all the packets transmitted between a sensor and the manager node must be kept secret so that eavesdroppers cannot intercept, modify and analyze, and discover valuable information in WSNs.

3. Security Threats and Requirements in Wireless Sensor Networks

In addition to the characteristics and considerations mentioned above, security threats and requirements are also critical for a variety of sensor network applications. In recent years, there are several security issues in WSNs have been proposed. In this section, we will introduce some security threats and requirements in WSNs.

1. Passive attacks : In passive attacks (such as eavesdropping attacks), eavesdroppers can unintrusively monitor on the communication channel between two communicating nodes to collect and discover valuable information without disturbing the communication (22; 24; 25).

2. **Active attacks** : active attacks (such as node replication attacks, sybil attacks, wormhole attacks, and compromised node attacks) can be further classified into two categories: external attacks and internal attacks. In external attacks (such as sybil attacks and wormhole attacks), a node does not belong to a sensor network and it can first eavesdrop on packets sent or received by normal participating nodes for the eventual purpose of malicious tempering, interfering, guessing, or spamming, and then injects invalid packets to disrupt the network functionalities.

    - For sybil attacks, a sensor node can illegitimately claim multiple IDs by either directly forging false IDs, or else impersonating legal IDs. This harmful attack may lead to serious threats to distributed storage, routing algorithm and data aggregation.
    - For wormhole attacks, the malicious node may be located within transmission range of legitimate nodes while legitimate nodes are not themselves within transmission range of each other. Thus, the malicious node can tunnel control traffic between legitimate nodes and nonexistent links which in fact are controlled by the malicious node. Finally, the malicious node can drop tunnelled packet or carry out attacks on routing protocols.

    Internal attacks (such as node replication attacks and node compromised attacks) are usually caused by compromised members who are belong to the sensor network in question, and hence internal attacks are more difficult to safeguard against than external attacks.

    - For node replication attacks, when a sensor node is compromised by attackers, they can directly place many replicas of this compromised node at different areas within the networks. Thus, attackers may use these compromised nodes to subvert the network functionalities, for example by injecting false sense data.
    - For compromised attacks, due to the lack of tamper resistance in sensor nodes, attackers may compromise a sensor node and use it to establish communication channels with non-compromised sensors to launch other more serious attacks within the sensor network.

According to the above description of the security threats, we can infer that a secure sensor network corresponds with the following requirements.

1. **Node authentication** : For this requirement, a deployed sensor node proves its validity to its neighboring sensors and the manger node. Thus, an invalid outsider would be unable to send malevolent data into the networks and the manager node can confirm that received sensed data has come from a valid sensor node, not from malicious outsiders. This also implies that a sensor node joined in WSNs has been authenticated and it has the right to access the sensor network.
2. **Availability** : The availability of the network should not be affected even if sensors can only provide limited storage, limited power, and limited computational ability. Therefore, a mechanism regulating of sleep patterns is necessary for a sensor to extend its lifetime.

3. **Location awareness** : The damage cannot be spread from the victimized area to the entire network by security attacks even if the sensor node is compromised. A secure communication scheme must limit the damage's scope caused by the intruders; the mechanism of location awareness is used for this purpose.
4. **Key establishment** For sensor-to-sensor key establishment, a shared key is established by two communication nodes to protect communications. Thus, all sensed data transmitted between participants could be verified and protected even if an attacker eavesdrops on the communications between nodes or injects illegal sensed data into networks, this requirement still provides an adequate level of security.
5. **No verification table** : The verification tables are not required to be stored inside the manager nodes to prevent stolen-verifier attacks.
6. **Confidentiality** : Path-key establishment in every session must be secure against malicious intruders even if those attackers collect transmission packets.
7. **Perfect forward secrecy** : In a two-party path-key establishment, a scheme is said to have perfect forward secrecy if revealing of the secret key to an intruder cannot help him/her derive the session keys of past sessions.
8. **Key revocation** : When the back-end system or the manager node decides to terminate a sensor utilizing task, or when a sensor is lost, the sensor must not be allowed to make use of the credential which it stores to connect to networks.
9. **Re-keying** : By introducing a re-keying mechanism, a manager node can conveniently update a sensor's credential without the intervention of back-end system for the purpose of reducing the communication interactions and management burden on that back-end system.

4. Literature Classifications

There are many researches about the application with key management proposed in the past. In this chapter, we classify wireless sensor network schemes into different classifications based on the application scenarios, including: deployment, organization, re-keying, cryptography and authentication. We then divide each classification into several subclassifications based on key management and node authentication. WSNs have a vast field of applications, including deployment and organization in both military and civilian aspects, from the battlefield surveillance, environment monitoring, medical sensing, traffic control and so on. Thus, the adoptions of security countermeasures are important issues and key management mechanisms are the core of the secure communications. Table 1 is showed the literature classification on secure communication schemes.

4.1 Deployment and Organization of WSNs

Depending on its applications, a sensor deployment manner can be classified in two types: scattered deployment and deployment in designated area. For scattered deployment, in order to achieve large scale of deployment, sensor nodes can be deployed via aerial scattering and the immediate neighboring nodes of any sensor node are unknown in advance. On the other hand, due to the unattended nature of WSNs, an attacker may launch various security threats such as node compromised attacks, the damage might be spread from the compromised area to the entire network. Therefore, many schemes deploy sensors in designated area in order to minimize and localize its impact to a small region.

Classification	Characteristic	Papers
Deployment	Scattered deployment	(1–3; 7; 9; 14; 15; 18; 33; 36; 37; 41–44)
	Designated area	(5; 6; 16; 21; 23; 26; 32; 37; 41; 42)
Organization	Distributed WSN	(1–3; 7; 9; 16; 18; 26; 36; 37; 41–44)
	Hierarchical WSN	(5; 6; 14; 15; 21; 23; 32; 33; 37; 41; 42)
Re-keying	Periodical update	(18; 23; 32; 37; 41; 42; 44)
	Node revocation/attachment	(1; 3; 6; 15; 17; 18; 23; 26; 33; 37; 41–43)
Cryptography	Symmetric key	(1; 3; 5–7; 14–16; 21; 23; 26; 32; 33; 36; 37; 41; 42; 44)
	Asymmetric key	(2; 14; 17; 23; 33; 37; 41–43)
	Hashing function	(3; 7; 15; 17; 18; 23; 32; 35–37; 41; 42; 44)
Authentication	Pair-wise authentication	(1; 3; 5–7; 15–17; 23; 26; 32; 33; 36; 37; 41–44)
	Group-wise authentication	(2; 14; 18; 21; 32; 33; 37; 41; 42; 44)

Table 1. The classification of secure communication schemes

In Figure 1, two general organizations for distributed and hierarchical WSNs are illustrated. A distributed/hierarchical structure of WSN consists of three types of participants, namely, a powerful back-end data center, manager nodes and sensor nodes. Each manager node is responsible for collecting and forwarding all sensed data of its managed area to the back-end data center for further processing from sensor nodes under the area for which it is responsible. In distributed WSNs, a number of sensors are uniformly distributed into sense field and there are no specific roles for each deployment sensor node. In hierarchical WSNs, there are two types of roles for deployment sensors, namely: cluster head and sensor node. Based on geographical and deployment knowledge, a manager node groups all sensors into multiple logical groups and the grouping function is conducted through the selection of cluster head for each group. The main objective of cluster heads are acting as aggregation nodes and fusing the sense data collected from their nearby sensor nodes before routing the resultant data to a manager node. Therefore, cluster heads are much more computational and communication ability than normal sensor nodes in hierarchical WSNs.

4.2 Authentication Scenarios

For authentication in WSNs, three types of scenarios for pair-wise and group-wise authentication are illustrated in Figure 2. For example, in Figure 2(a), a pair-wise authentication is accomplished between node x and node y. For group-wise authentication, we divided it into two scenarios: cluster-based authentication and global-based authentication. In Figure 2(b), a cluster authentication is used by a cluster head and all its neighboring sensor nodes, and it

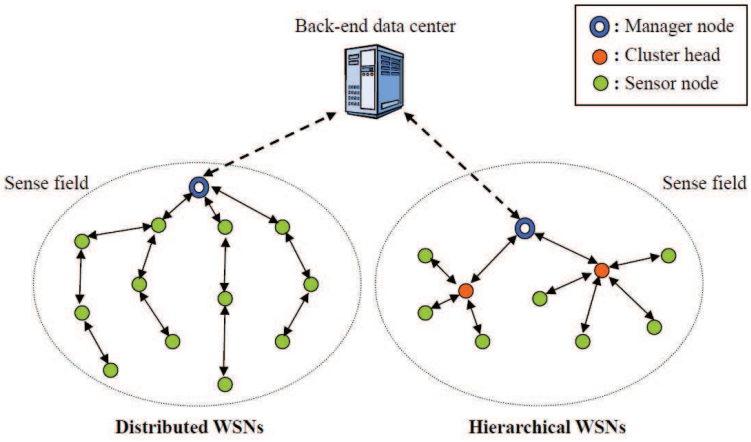


Fig. 1. Organization of WSNs

is used for securing clustered broadcast messages. Finally, in Figure 2(c), this is a node authentication verified by the manager node and all sensor nodes in the sense field. A global authentication is done by the manager node for securing communications that are broadcast to the entire network and prevent illegal sensor nodes from participating the sensor networks.

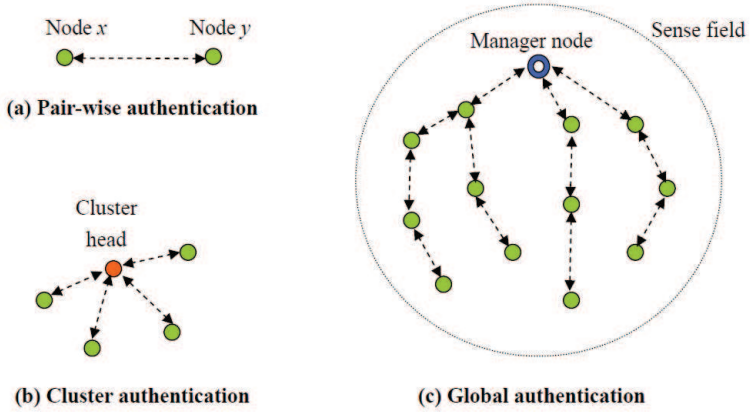


Fig. 2. Authentication scenarios

4.3 Cryptographic Approaches

In order to protect privacy and secure communications, participating nodes joined in WSNs should be authenticated and shared keys should be established between deployed sensors and their neighboring nodes. For example, in two-party communications, a deployed node establishes a pair-wise key with each of its neighboring nodes. Similarly, in broadcast communications, a group-wise key should be shared by all nodes in the network. We classify the

security of node authentication and key establishment schemes into three types of cryptography: symmetric keys, asymmetric keys and one-way hashing functions.

4.3.1 Symmetric Keys

Recently, many schemes (1; 3; 5–7; 14–16; 21; 23; 26; 32; 33; 36; 37; 41; 42; 44) were proposed to secure communications in WSNs and one of secure communication schemes is based on symmetric key cryptography. A simple solution to ensure privacy would be store a single master key MK into all deployed sensors prior to their deployment. Thus, a legal node N_A can use this master key to establish a pair-wise key $K = F(MK||N_A||N_B)$ with its neighboring node N_B for securing communications that require privacy or node authentication, where F is a pseudo-random function. However, this solution fails to prevent security breaches and thus is impracticable for WSNs for whose sensors lack tamper resistance and are easy for attackers to compromise, leaving all the secret in those networks known to attackers. As a result, during initial deployment phase, we suggest that there should be a security mechanism for erasing master key. For example, the manager node sets a timer with reasonable time interval T for a deployed sensor to discover its neighboring nodes. When a timer expires after T, deployed sensor node erases MK and attackers cannot inject illegal sensed data into networks without knowing MK.

The other extreme solution is to store a set of $n-1$ key pairs in each sensor node before deployment in such a way that it shares a unique key pair with all other nodes in the networks, where n is the number of sensor nodes in WSNs. However, this solution is only suitable for small networks due to it requires large memory to store keys and becomes a serious problem when the network needs to be expanded. Therefore, many probabilistic key pre-deployed schemes were proposed to overcome these shortages. A large pool of P keys and their identifiers are generated and d distinct keys are randomly drawn from P and pre-loaded into each sensor's key ring, where $P \gg d$. This solution ensures that only a few keys need to be stored in each sensor's memory and two nodes share at least one key, based on a selected probability. An extension to the basic probabilistic scheme is proposed by Liu and Ning, called polynomial pool-based key pre-distribution scheme (26). This scheme randomly selects polynomials from a polynomial pool and stores them to each sensor instead of randomly choosing keys from a key pool. A detailed survey on symmetric keying schemes could be found in (37; 41; 42)

4.3.2 Asymmetric Keys

As sensors have constrained resources and are expensive to install, computational and communication overhead must be kept at a minimum. Hence the traditional asymmetric cryptosystems such as RSA (34) and ElGamal (10) are not suitable to use in WSNs and most key management and establishment schemes for WSNs are based upon symmetric key cryptography. However, many security solutions based on symmetric keys are usually subject to various attacks and they are unable to achieve sufficient scalability (2). On the contrary, asymmetric key cryptography provide better scalability and security strength and allow for flexible key management as it does not require pre-distribution of keys. Therefore, several solutions based on asymmetric key algorithms have been proposed in the literature (2; 14; 17; 23; 33; 37; 41–43). Gura et al. (12) showed that both RSA and Elliptic Curve Cryptography (ECC) (20) public key cryptography (PKC) is applicable on two 8-bit CPUs without hardware acceleration and ECC is widely being adopted to provide PKC support so that the existing PKC-based solutions can be exploited. TinyECC(27), a software package, is being investigated to provide ECC-based PKC operations that can be flexibly configured and integrated into limited-resource sensor

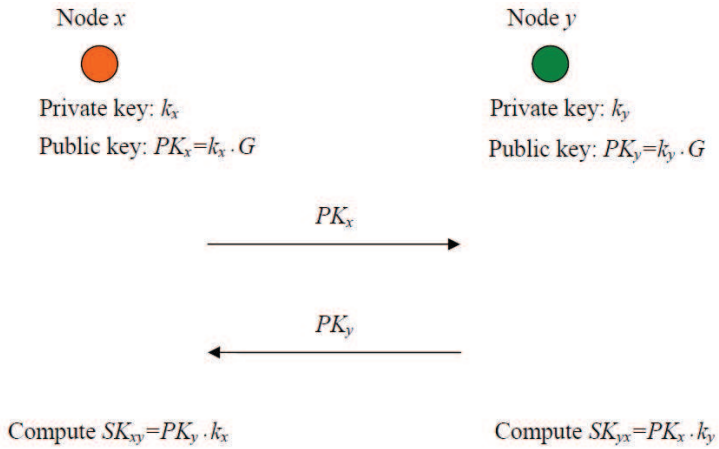


Fig. 3. Key exchange of a agreed pair-wise key under Diffie-Hellman based on ECDLP

devices. Targeted at security of TinyECC, it provides PKC-based schemes that have proven to be secure; ECC-160 and ECC-224 have the same security level as RSA-1024 and RSA-2048, respectively. Moreover, at the beginning of the node deployment, two nodes establish the permanent pair-wise key using a computationally less expensive variant of the Diffie-Hellman key exchange scheme (8) based on the Elliptic Curve Discrete Logarithm Problem (ECDLP) (11; 28), as shown in Figure 3. Each node is pre-loaded with private/public key pair and only two rounds of handshake are required. Private keys are denoted as k_x and k_y for node x and node y, respectively as well as the public keys $PK_x = k_x \cdot G$ and $PK_y = k_y \cdot G$. After receiving neighboring node's public key, each node will compute agreed pair-wise key as $SK_{xy} = PK_y \cdot k_x = k_x \cdot k_y \cdot G = PK_x \cdot k_y = SK_{yx}$.

In 2007, Zhou et al. design an access control scheme (43) based on ECC for sensor networks and their scheme accomplishes node authentication and key establishment to prevent malicious nodes from joining sensor networks. In 2009, Huang proposed an improved version (17) of Zhou et al.'s scheme to reduce large amounts of computations and communications between two nodes. In (14), Hsieh et al. proposed a dynamic authentication protocol to authentication a new node-joining sensor network, establishment of secure links and broadcast authentication between neighboring nodes in cluster-based sensor networks. In (2), Cao et al. proposed an ID-based multi-user broadcast authentication scheme based on ECC for providing strong security, sound scalability and performance efficiency simultaneously.

4.3.3 One-way Hashing Functions

One-way hashing functions (such as MD5 and SHA-1) are important tools in the field of cryptographic applications due to their efficiency with regard to computational costs and are suitable for resource-constrained devices. In general, the security of an one-way hashing function $h(.)$ is based on the hardness of inverting the inputs from the outputs; that is, given a and $h(.)$, it is easy to compute $h(a) = b$. However, only given b, it is hard to find a, satisfying $h(a) = b$. Figure 4 shows the construction of an one-way hash chain. Participating nodes generate an initial value $h^1(k) = h(k)$, where k is the initial key and $h^1(k)$ represents the initial key k has been hashed once. Thus, h^n can be regarded as the key k which has been hashed n times such

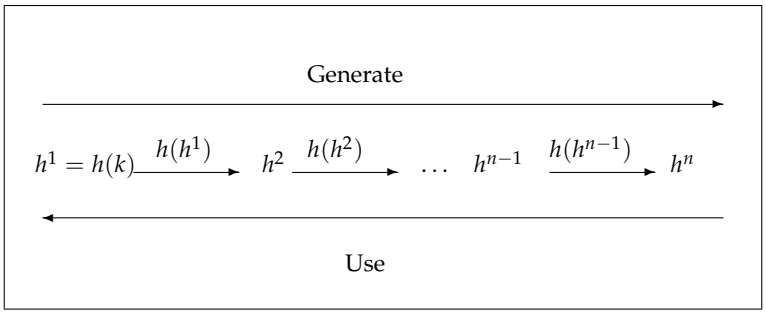


Fig. 4. Construction of an one-way hash chain

that $h^n(k) = h(h^{n-1}(k))$, where $n = 2, 3, 4, \ldots$. Due to the one-way property, the hash chain can be used in reverse order of generation for authentication; that is, $h^{n-1}(k)$ can be proven to be authentic if $h^n(k)$ has been proven to be authentic. For example, we assume that the lifetime of a sensor network is divided into n intervals and each time interval T_m has its own master key $K_m = h(K_{m-1})$, where $1 \leq m \leq n$, $K_1 = h(k)$ and k is an initial key. Figure 5 illustrates the mapping between master keys and time intervals.

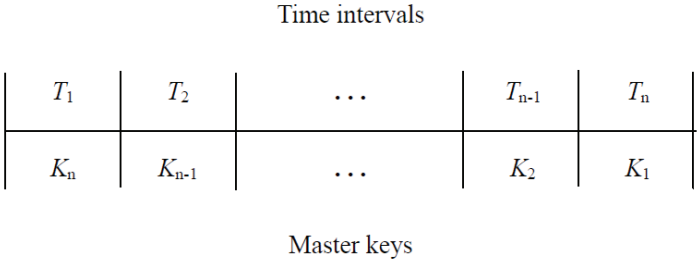


Fig. 5. The mapping between master keys and time intervals by using one-way hash chain

In addition, the Message Authentication Code (MAC) which is generated by node x and it would be verified by node y and is defined by $MAC = h(k; m)$, where m denotes the message under the protection key of k. Several solutions based on one-way hashing functions have been proposed in the literature (3; 7; 15; 17; 18; 23; 32; 35–37; 41; 42; 44). In (35), Shan and Liu proposed the hashed random key pre-distribution, if nodes x and y are deployed in WSNs, respectively, with $K_x = h^a(k)$ and $K_y = h^b(k)$, then node y can easily derive $K_x = h^{a-b}(K_y)$, where k is randomly selected from the key pool and $a > b$. In Li et al.'s scheme (23), the concepts of MAC and one-way hash chain are widely be used to authenticate the validity of transmission messages and participating nodes.

5. Conclusions and Future Works

We argue that no single security scheme is ideal to all the applications where sensor networks are used and the cryptographic techniques adopted must depend upon the scenarios of ap-

plied architectures and concerns of security requirements in WSNs. There are some future research issues should be considered for wireless sensor networks in this chapter. There are also the critical success factors of wireless sensor networks. We briefly describe them as follows.

- **Soft message encryption**: In order to achieve performance efficiency and reduce resource requirements, a soft message encryption mechanism is used in which a message is divided into different parts and each part of the message is involved in encrypting the whole message itself. This technique has less strength than the sophisticated encryption algorithms. However, it eliminates the need of key distribution centers and key establishment (29). For soft message encryption (13), we assume that a $3m$-bits message is divided into three parts of m bits each and we define these parts by x, y and z. Then, parts x, y and z are encrypted by the following conditions:

  $$x' = x \oplus z$$
  $$y' = y \oplus x$$
  $$z' = z \oplus x \oplus y$$

  Now, the parts x', y' and z' are now transmitted instead of x, y and z. Finally, at the back-end data center, the message parts can be decrypted using the following equations:

  $$x = x' \oplus y' \oplus z'$$
  $$y = x' \oplus z'$$
  $$z = y' \oplus z'$$

- **Multiple communication paths**: For pair-wise key establishment in single communication path, it is vulnerable to stop forwarding attack if an intermediate node is compromised along the path. Moreover, it cannot prevent Byzantine attacks that attackers may use the compromised nodes to alter, inject, spoof, or sniff messages. A secret key may be exposed if any intermediate node along the path is compromised and a secret key established between the source node and destination node by multiple communication paths can decrease the risk of path key exposure problem. Therefore, multi-path key establishment solutions are resilient to resist stop forwarding, ensure network availability from connective failure and prevent compromised sensors from knowing the secret in WSNs (30; 36; 38). In Figure 6, we use the above-mentioned soft encryption with multiple communication paths as an example.
- **Efficient data aggregation**: The main objective of data aggregation technique is to combine the sensed data receiving from deployed sensor nodes at certain cluster heads to minimize the total amount of data transmission before forwarding sensed data to the external manager node. An efficient and secure data aggregation is essential for cluster-based WSNs in which data aggregation is eliminating data redundancy to reduce energy consumption to extend the network lifetime (4; 19; 31; 39; 40). An example of data aggregation is presented in Figure 7 where a group of data aggregators collect the data from their neighboring nodes, aggregate them and send the aggregated data to the manager node.
- **Malicious node detection**: In order to ensure a secure networking, it should design a security mechanism to detect malicious nodes and false messages by legitimate nodes

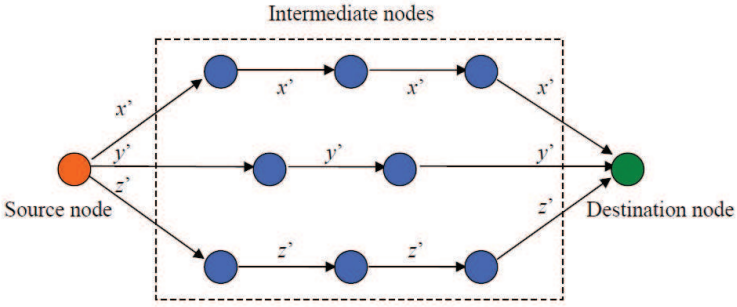


Fig. 6. An example of multiple communication paths

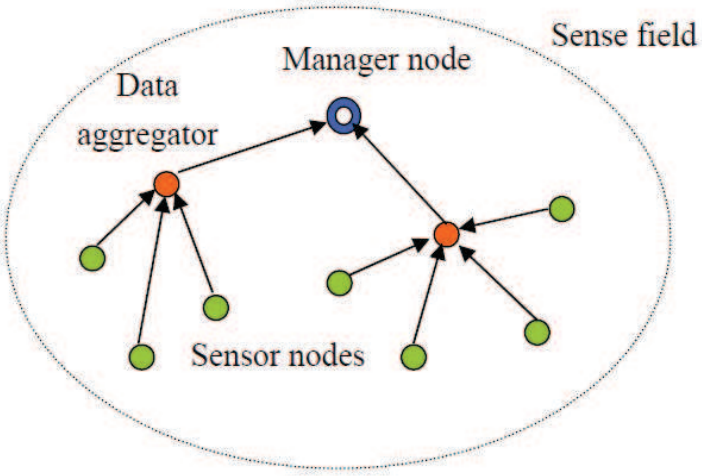


Fig. 7. Data aggregation in cluster-based wireless sensor networks

and the basic idea of detection of malicious behavior node is to provide a hop-by-hop authentication in WSNs.

- **Node revocation-awareness**: Unlike the addition of a sensor node to WSNs, the revocation of a node is much more complicated. When a sensor node is compromised or it exhausts its power, it must not be allowed to make use of the key information stored in local memory to connect to the sensor networks and it requires many keys to be revoked. As a result, it is important to design a node revocation-awareness mechanism without bring serious impacts on the network efficiency and connectivity.

6. References

[1] Sasikanth Avancha, Jeffrey Undercoffer, Anupam Joshi and John Pinkston, "Secure sensor networks for perimeter protection", *Computer Networks*, vol. 43, no. 4, pp. 421-435, 2003.

[2] Xuefei Cao, Weidong Kou, Lanjun Dang and Bin Zhao, "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks", *Computer Communications*, vol. 31, no. 4, pp. 659-667, 2008.
[3] Haowen Chan, Virgil D. Gligor, Adrian Perrig and Gautam Muralidharan, "On the Distribution and Revocation of Cryptographic Keys in Sensor Networks", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 233-247, 2005.
[4] Huifang Chen, Hiroshi Mineno and, Tadanori Mizuno, "Adaptive data aggregation scheme in clustered wireless sensor networks", *Computer Communications*, vol. 31, no. 15, pp. 3579-3585, 2008.
[5] Yi Cheng and Dharma P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 35-48, 2007.
[6] Michael Chorzempa, Jung-Min Park and Mohamed Eltoweissy, "Key management for long-lived sensor networks in hostile environments", *Computer Communications*, vol. 30, no. 9, pp. 1964-1979, 2007.
[7] Mauro Conti, Roberto Di Pietro and Luigi V. Mancini, "ECCE: Enhanced cooperative channel establishment for secure pair-wise communication in wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 49-62, 2007.
[8] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.
[9] E. Ekici, S. Vural, J. McNair and D. Al-Abri, "Secure probabilistic location verification in randomly deployed wireless sensor networks", *Ad Hoc Networks*, vol. 6, no. 2, pp. 195-209, 2008.
[10] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, 1985.
[11] A. Riskiran and R. Lee, "Workload characterization of elliptic curve cryptography and other network security for constrzined environments", *in Proceedings of IEEE International Workshop on Workload Characterization*, pp. 127-137, 2002.
[12] N. Gura, A. Patel, A. Wander, H. Eberle and S. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit cpus", *In 2004 Workshop on Cryptographic Hardware and Embedded Systems*, August 2004.
[13] T. Haniotakis, S. Tragoudas and C. Kalapodas, "Security enhancement trhough multiple path transmission in ad hoc networks", *in IEEE International Conference on Communiations*, pp. 4187-4191, 2004.
[14] Meng-Yen Hsieh, Yueh-Min Huang and Han-Chieh Chao, "Adaptive security design with malicious node detection in cluster-based sensor networks", *Computer Communications*, vol. 30, no. 11-12, pp. 2385-2400, 2007.
[15] Fei Hu, Waqaas Siddiqui and Krishna Sankar, "Scalable security in Wireless Sensor and Actuator Networks (WSANs): Integration re-keying with routing", *Computer Networks*, vol. 51, no. 17, pp. 285-308, 2007.
[16] D. Huang and D. Medhi, "Secure pairwise key establishment in large-scale sensor networks: an area partitioning and multigroup key predistribution approach", *ACM Transactions on Sensor Networks*, vol. 3, no. 3, article 16, 2007.
[17] H. F. Huang, "A novel access control protocol for secure sensor networks", *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 272-276, 2009.

[18] Yixin Jiang, Chuang Lin, Minghui Shi and Xuemin (Sherman) Shen, "Self-healing group key distribution with time-limited node revocation for wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 14-23, 2007.

[19] Jamal N. Al-Karaki, Raza Ul-Mustafa and Ahmed E. Kamal, "Data aggregation and routing in Wireless Sensor Networks: Optimal and heuristic algorithms", *Computer Networks*, vol. 53, no. 7, pp. 945-960, 2009.

[20] K. Koblitz, "Elliptic Curve Cryptosystems", *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[21] Jason H. Li, Bobby Bhattacharjee, Miao Yu and Renato Levy, "A scalable key management and clustering scheme for wireless ad hoc and sensor networks", *Future Generation Computer Systems*, vol. 24, no. 8, pp. 860-869, 2008.

[22] Chun-Ta Li, Min-Shiang Hwang and Yen-Ping Chu, "Improving the Security of A Secure Anonymous Routing Protocol with Authenticated Key Exchange for Ad Hoc Networks", *International Journal of Computer Systems Science and Engineering*, vol. 23, no. 3, pp. 227-234, 2008.

[23] Chun-Ta Li, Min-Shiang Hwang and Yen-Ping Chu, "An Efficient Sensor-To-Sensor Authenticated Path-Key Establishment Scheme for Secure Communications in Wireless Sensor Networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, 2009.

[24] Chun-Ta Li and Yen-Ping Chu, "Cryptanalysis of Threshold Password Authentication Against Guessing Attacks in Ad Hoc Networks", *International Journal of Network Security*, vol. 8, no. 2, pp. 166-168, 2009.

[25] Chun-Ta Li, Min-Shiang Hwang and Yen-Ping Chu, "A Secure Event Update Protocol for Peer-To-Peer Massively Multiplayer Online Games Against Masquerade Attacks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12(A), pp. 4715-4723, 2009.

[26] D. Liu and P. Ning, "Improving Key Predistribution with Deployment Knowledge in Static Sensor Networks", *ACM Transactions on Sensor Networks*, vol. 1, no. 2, pp. 204-239, 2005.

[27] An Liu and Peng Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks", *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.

[28] D. Malan, M. Welsh and M. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography", *in Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 71-80, 2004.

[29] Prayag Narula, Sanjay Kumar Dhurandher, Sudip Misra and Isaac Woungang, "Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing", *Computer Communications*, vol. 31, no. 4, pp. 760-769, 2008.

[30] Nidal Nasser and Yunfeng Chen, "SEEM: Secure and energy-efficient multipath routing protocol for wireless sensor networks", *Computer Communications*, vol. 30, no. 11-12, pp. 2401-2412, 2007.

[31] Suat Ozdemir and Yang Xiao, "Secure data aggregation in wireless sensor networks: A comprehensive overview", *Computer Networks*, vol. 53, no. 12, pp. 2022-2037, 2009.

[32] Biswajit Panja, Sanjay Kumar Madria and Bharat Bhargava, "A role-based access in a hierarchical sensor network architecture to provide multilevel security", *Computer Communications*, vol. 31, no. 4, pp. 793-806, 2008.

[33] Rabia Riaz, Ayesha Naureen, Attiya Akram, Ali Hammad Akbar, Ki-Hyung Kim and H. Farooq Ahmed, "A unified security framework with three key management schemes for wireless sensor networks", *Computer Communications*, vol. 31, no. 18, pp. 4269-4280, 2008.

[34] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[35] T. Shan and C. Liu, "Enhancing the key pre-distribution scheme on wireless sensor networks", *in IEEE Asia-Pacific Conference on Service Computing*, IEEE CS, pp. 1127-1131, 2008.

[36] Jang-Ping Sheu and Jui-Che Cheng, "Pair-wise path key establishment in wireless sensor networks", *Computer Communications*, vol. 30, no. 11-12, pp. 2365-2374, 2007.

[37] Marcos A. Simplicio Jr., Paulo S.L.M. Barreto, Cintia B. Margi and Tereza C.M.B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks", *Computer Networks*, article in press, 2010.

[38] Eliana Stavrou and Andreas Pitsillides, "A survey on secure multipath routing protocols in WSNs", *Computer Networks*, vol. 54, no. 13, pp. 2215-2238, 2010.

[39] S. Upadhyayula and S. K. S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 5, pp. 626-648, 2007.

[40] Kui Wu, Dennis Dreef, Bo Sun and Yang Xiao, "Secure data aggregation without persistent cryptographic operations in wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 100-111, 2007.

[41] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fei Hu and Michael Galloway, "A survey of key management schemes in wireless sensor networks", *Computer Communications*, vol. 30, no. 11-12, pp. 2314-2341, 2007.

[42] Junqi Zhang and Vijay Varadharajan, "Wireless sensor network key management survey and taxonomy", *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63-75, 2010.

[43] Yun Zhou, Yanchao Zhang and Yuguang Fang, "Access control in wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 3-13, 2007.

[44] S. Zhu, S. Setia and S. Jajodia, "LEAP+: Efficient seurity mechanisms for large-scale distributed sensor networks", *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500-528, 2006.

A Compromise-resilient Pair-wise Rekeying Protocol in Hierarchical Wireless Sensor Networks

Song Guo
School of Computer Science and Engineering The University of Aizu, Japan
sguo@u-aizu.ac.jp

Zhuzhong Qian
State Key Laboratory for Novel Software Technology Nanjing University, China
qzz@nju.edu.cn

1. Introduction

Wireless sensor networks (WSNs) have been envisioned to be very useful for a broad spectrum of emerging civil and military applications (Akyildiz et al., 2002). However, sensor networks are also confronted with many security threats such as node compromise, routing disruption and false data injection, because they are normally operated in an unattended, harsh or hostile environment. Among all these threats, the WSNs are particularly vulnerable to the node compromise because sensor nodes are not tamper-proof devices. When a sensor network is deployed in unattended and hostile environments such as battlefield, the adversaries may capture and reprogram some sensor nodes, or inject some malicious sensor nodes into the network and make the network accept them as legitimate nodes. After getting control of a few nodes, the adversary can mount various attacks from inside the network (Zhang et al., 2008). Therefore, it is desirable to design key distribution protocols to support secure and robust pair-wise communication among any pair of sensors.

This is a challenging task in sensor networks because they have scarce resources in energy, computation and communication. As a result, the conventional asymmetric key cryptosystem, such as RSA (Rivest et al., 1978) and Diffie-Hellman (Diffie & Hellman, 1976), can not be implemented in sensor nodes due to their very limited capacities and only lightweight energy efficient key distribution mechanisms are affordable. Furthermore, sensor nodes are low-cost and they cannot afford tamper-resistance hardware. Recent advances in physical attack show that even memory chips with built-in tamper-resistance mechanisms are subject to various memory read-out attacks. Thus, an adversary might easily capture the sensor devices to acquire their sensitive data and keys and then abuse them to further compromise the communication between other non-captured nodes. In order to conquer such *node capture attack* (NCA) problem, it is desirable to design protocols to support secure and robust pair-wise communication among any pair of sensors.

To defend against such attack, the security mechanisms in WSNs are required. Most of existing key management schemes focus on the efficiency of bootstrapping session keys which has been intensively studied in the literature of WSNs (Cheng & Agrawal, 2005; Du et al., 2003; Eschenauer & Gligor, 2002). Traditionally, once such key system is adopted, the whole security system is established and fixed. However, when the WSN runs for a long time using a

fixed key, it enhances the probability for the adversaries to decrypt the key by analyzing the adequate messages eavesdropped or capturing some nodes. Under this circumstance, the entire network security might be threatened. Thus, it is necessary to update this key with a new key periodically to maintain backward secrecy (Mishra, 2002). The idea is to prevent a node with the new key from going backwards in time to decipher previous content encrypted with prior keys. Likewise, when a node leaves, it is necessary to update the key to maintain forward secrecy (Mishra, 2002). The idea is to prevent a node from using an old key to continue to decrypt new content.

WSNs can be broadly classified into flat WSNs and hierarchical WSNs. It has been shown in (Cheng & Agrawal, 2007) that a hierarchical architecture can provide better performance, in terms of communication overhead, than a flat architecture in such networks. This is the marjor reason why most recent lightweight energy efficient rekeying mechanisms are proposed for hierarchical WSNs. In a flat WSN, all senor nodes have the same computational and communication capacities. In a hierarchical WSN, however, some special sensor devices, called Cluster Head (CH), have much higher capacities than other sensor nodes. By applying some clustering algorithms like (Heinzelman et al., 2002), the whole set of sensor devices could be partitioned into several distinct clusters such that each cluster has at least one CH. Under this arrangement, each sensor node forwards the generated packets to its local CH by short-range transmissions, and the CH then performs a pre-processing for the raw data received from all other senor nodes in the cluster and finally forwards the aggregated data to the sink node, or Base Station (BS), by long-range transmissions.

Most existing polynomial-based rekeying schemes suffer the node capture attack. Let us examine Chadha's rekeying protocol proposed in (Chadha et al., 2005) as an example to show its vulnerability to NCA. The basic idea is that the rekeying message from a CH can disallow the compromised nodes to renew their pair-wise keys. In the pre-loading phase, each sensor node S_i is pre-loaded the secret values $h(S_i)$ obtained from a $2t$-degree masking polynomial $h(x)$. This scheme assumes that each CH has the intrusion detection capacity. In the rekeying phase, the CH generates a t-degree secrecy polynomial $f(x)$ and constructs $w(x)$ as $w(x) = g(x)f(x) + h(x)$, where $g(x)$ is constructed using the Ids of all compromised nodes. Once $g(x)$ is evaluated at the Id of any malicious node, the result will be equal to 0. The CH then broadcasts $w(x)$ and the Id list of all detected compromised nodes throughout the whole group members. Upon receiving the message, any non-revoked node S_i can compute the new pair-wise key $f(S_i)$ between sensor node S_i as follows: $f(S_i) = (w(S_i) - h(S_i))/g(S_i)$. We observe that if there are $(2t + 1)$ nodes are compromised in an arbitrary rekeying phase, the $2t$-degree polynomial $h(x)$ can be derived. Recalling that the polynomials $w(x)$ and $g(x)$ are public, we conclude that $f(x)$ can be derived as well and used to calculate the pair-wise key in any given rekeying phase. In addition, their vulnerability to the node capture attack disables them from supporting both forward and backward secrecy. This motivates us to design a new compromise-resilient pair-wise rekeying scheme with strong resistance to such attack.

The rest of this paper is organized as follows. Section 2 presents our system model and gives an overview of background knowledge. Section 3 describes a perturbation based pair-wise rekeying protocol. Sections 4 and 5 evaluate the security and the performance of our proposal, respectively. Section 6 summarizes our findings.

2. Preliminaries

2.1 Network Model

As in other hierarchical models of sensor network (Cheng & Agrawal, 2007; Zhang et al., 2005), our system also assumes that a sensor network is divided into clusters, which are the

minimum unit for detecting events. A cluster head coordinates all the actions inside a cluster and each pair of cluster heads in their transmission range can communicate directly with each other. Each low-cost sensor node (SN) has low data processing capability, limited memory storage and battery power supplies, and short radio transmission range. The CHs are equipped with richer resources (*e.g.*, higher power batteries, large memory storages, powerful antenna, *etc*.) and higher data processing capacities, and thus can execute relatively complicated numerical operations. Moreover, we assume a single base station (BS) or an access point (AP) in the network and works as the network controller to collect event data. The information collected by cluster heads from all its sensor nodes is retrieved by a BS or a AP periodically. During the information retrieval operation, the BS/AP broadcasts a beacon to activate cluster heads in its coverage area. Activated cluster heads then transmit their data to the BS/AP through a common wireless channel. As the most powerful node in a WSN, the BS/AP has virtually unlimited memory storage capacity and sufficiently large radio transmission range to reach all other devices in a network.

Under such model, we say the link (v, u), corresponding to the wireless communication channel between nodes v and u, is secure if they share a secret pairwise key $K_{v,u}$. Due to the constrained resources, computationally expensive and energy-intensive operations for pairwise key establishment are not favorable for such systems. In addition, each sensor node is not tamper-resistant. Once a sensor node is captured, the adversary can read its memory to get all information stored there. Schemes for key predistribution enable nodes in a large network to agree on pairwise secret keys. The sensor network is administrated by an offline authority, which is responsible for node initialization and deployment. Before deploying a node, the authority assigns the node a unique identity (ID) from a set of legitimate IDs and some secret information that will be used to allow any two nodes v and u to agree on a shared key $K_{v,u}$.

2.2 Symmetric Polynomial Function

As the basis of our pair-wise rekeying protocol for any wireless link between a CH and a SN, the polynomial-based key predistribution scheme originally proposed in (Blundo et al., 1993) works as follows.

Let F_q be a finite field, in which q is the maximum prime number satisfying $q < 2^\ell$ that can accommodate a cryptographic key with ℓ bits. The elements of F_q can be used as pairwise keys. To achieve t-resilience using the Blundo's scheme (Blundo et al., 1993), the authority chooses a random symmetric bivariate polynomial $f \in F_q[x, y]$ of degree t in each variable as the master secret polynomial:

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j. \tag{1}$$

The coefficients a_{ij} ($a_{ij} = a_{ji}$) are randomly chosen from F_q. A node with Id $u \in F_q$ is preloaded the univariate polynomial:

$$g_u(y) = f(u, y). \tag{2}$$

The shared key $K_{v,u}$ between nodes v and u is

$$g_v(u) = f(v, u) = g_u(v), \tag{3}$$

which both parties can compute using the fact that $f(x, y)$ is symmetric. The security proof in (Blundo et al., 1993) ensures that this scheme is unconditionally secure and t-collusion resistant; *i.e.*, a coalition of no more than t compromised nodes cannot know anything about

the key shared by any two non-compromised nodes. However, an attacker who compromises $t+1$ nodes can use interpolation to recover the master polynomial $f(x,y)$.
By applying the symmetric property, a secure link can be easily built up by just exchanging the IDs of transmission nodes. On the other hand, a t-degree bivariate polynomial key scheme can only keep secure against coalitions of up to t compromised sensors. Although increasing the value of t can improve the security property of bivariate polynomial key scheme, it is not suitable for wireless sensor networks due to the limited memory size of sensors.

2.3 Perturbation Polynomial Function

Our proposed pair-wise rekeying protocol exploits the characteristic of the perturbation polynomial, which was originally introduced in (Zhang et al., 2007). Given a finite field F_q, a positive integer r ($r < \ell$), and a set of node Ids S ($S \subset \{0, \cdots, q-1\}$), a polynomial set Φ is a set of perturbation polynomials regarding r and S if any polynomial $\phi(\cdot) \in \Phi$ has the following *limited infection* property:

$$\forall x \in S, \phi(x) \in \{0, \cdots, 2^r - 1\}. \tag{4}$$

According to the above definition, the value of a perturbation polynomial will not be larger than $(2^r - 1)$, *i.e.*, it has at most r bits. This property is used to design perturbation-based scheme. If let an r-bit number add to a ℓ-bit number, only the least significant r-bit of the ℓ-bit numer will be directly affected. Wheather the most significant $(\ell - r)$ bits are changed or not will hinge on if a carry is generated from the least significant r bits in the addition process. For example, we assume $\ell = 6$ and $r = 4$. The addition $(101001)_2 + (0101)_2 = (101110)_2$ changes the least significant 4-bits but not the most $\ell - r = 2$ significant bits of the first operand, but $(101001)_2 + (1100)_2 = (110101)_2$ not only changes least significant 4-bits but also the most significant 2 bits, because a carry is generated from the least significant 4-bits.

3. A Pair-wise Rekeying Protocol

In general, the design of a light-weight compromise-recilient rekeying scheme in WSNs is difficult because of the vulnerability of sensor nodes and the constrained system resources. Due to these challenges, a practical pair-wise rekeying scheme for WSNs should be resilient to large number of node compromises, be efficient in computation, communication, and storage, and allow both full and direct key establishment. In this section, we present a perturbation-based pair-wise rekeying protocol that can achieve all these goals.
In the basic polynomial-based scheme (Blundo et al., 1993), where any two nodes (with IDs u and v) are given shares ($f(u,y)$ and $f(v,y)$) of a symmetric polynomial $f(x,y)$, they can always find a match $f(u,v)$ to be used as the shared key of size ℓ bits. Different from this, our rekeying scheme does not use shares generated from symmetric polynomial but perturbation polynomials such that (1) a match can still be achived and (2) the shared key is difficult to crack by large-scale NCAs. To further explain the above basic idea, we now introduce the three major steps of the rekeying scheme: system initialization, pre-distribution of perturbed polynomials, and key establishment and rekeying. In order to present it in a formal way, we list the notations used in our protocol descriptions in Table 1 for convenience to the readers.

3.1 System Initialization

We assume that there are n sensor nodes to be deployed in the network. The node deployment can be done by only once, or several times in order to extend the lifetime of the network with

Notation	Description
CH_a	The Id of cluster head a
CS_k	The Id of compromised sensor node k
$E(\texttt{data}, K)$	An encryption function using K as a key
$f(x,y)$	a symmetric polynomial
F_q	a finite field with any element that can be represented by ℓ bits
$g_u(y)$	the univariate polynomial for node u obtained by $g_u(y) = f(u,y)$
$\bar{g}_u(y)$	the perturbed polynomial preloaded to node u
$H^k(x)$	the hashed value based on the most significant k bits of x
$K_{a,b}$	the shared pairwise key between nodes a and b
ℓ	the minimal integer satisfying $2^\ell > q$
n	the total number of sensor nodes to be deployed, $n < q$
n_a	the number of sensor nodes in a cluster
n_c	the number of compromised sensor nodes in a cluster
m	the total number of perturbation polynamials, $m = \|\Phi\|$
$p_u(y)$	a randomly generated univariate rekeying polynomial at node u
q	a large prime number
r	a positive integer such that $2^r < q$
S	a set of legitimate IDs for sensor nodes, $S \subset \{0, \cdots, q-1\}$
SN_i	The Id of sensor node i
t	the degree of both variables x and y in the symmetric polynomial $f(x,y)$
$\phi_u(y)$	a perturbation polynamial assigned for node u
Φ	a set of perturbation polynamials satisfying the limited infection property regarding r and S

Table 1. Notations

the renewed nodes. Based on the number n, a large prime number q is chosen such that $n < q$ and let ℓ be the minimal integer satisfying $2^\ell > q$.
The offline authority arbituary constructs a bivariate symmetric polynomial $f(x,y) \in F_q[x,y]$, where the degrees of x and y are both t, and for any $x, y \in F_q$, $f(x,y) = f(y,x)$. It then applies the method in (Zhang et al., 2007) to construct the legitimate ID set S for sensor nodes and the perturbation polynamial set Φ, which satisfies the limited infection property regarding r and S with m ($m \geq 2$) number of bivariate symmetric polynomials. Finally, we note that the desired number of bits for any pairwise key is $\ell - r$.

3.2 Pre-distribution of Perturbed Polynomials

Before sensor devices are deployed into usage, some secret information should be pre-assigned as follows. Each cluster head a needs to be preloaded with a unique Id $CH_a \in S$ and a perturbed polynomial $\overline{g}_{CH_a}(y)$:

$$\overline{g}_{CH_a}(y) = f(CH_a, y) + \phi_{CH_a}(y) = g_{CH_a}(y) + \phi_{CH_a}(y). \tag{5}$$

Similarly, for each sensor node i, the security server preloads it with a unique Id $SN_i \in S$ and a perturbed polynomial $\overline{g}_{SN_i}(y)$:

$$\overline{g}_{SN_i}(y) = f(SN_i, y) + \phi_{SN_i}(y) = g_{SN_i}(y) + \phi_{SN_i}(y). \tag{6}$$

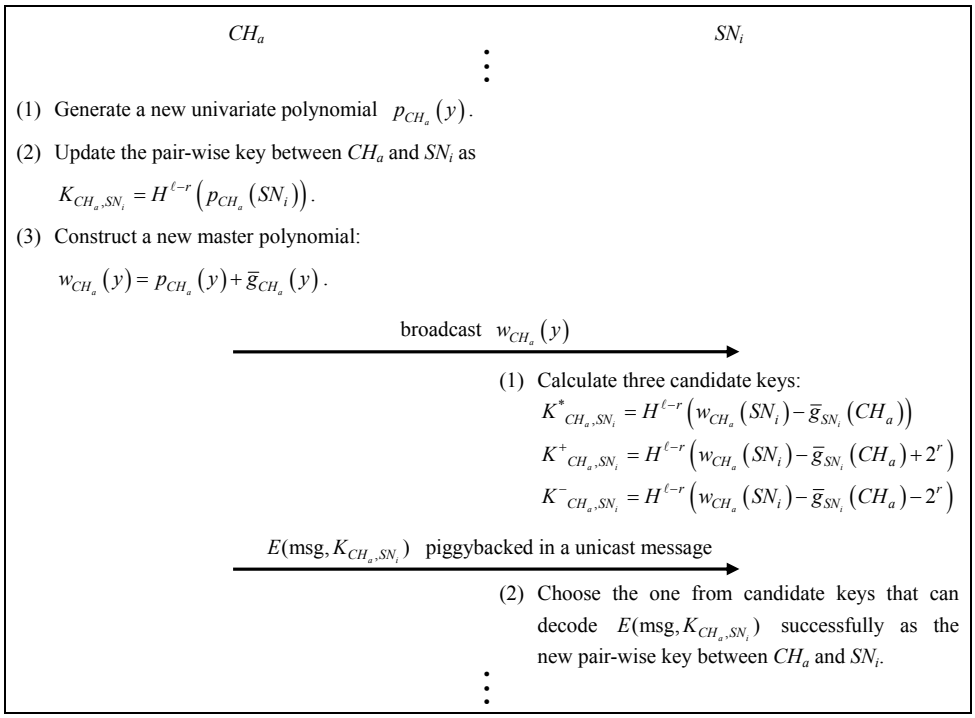


Fig. 1. The protocol for pair-wise key establishment and rekeying

Note that the security authority only preloads each sensor device u (a CH or SN) the coefficients of $\overline{g}_u(y)$. Hence, each sensor device cannot extract from $\overline{g}_u(y)$ the coefficients of the original polynomial shares of either $f(x,y)$, $f_u(y)$, or $\phi_u(y)$ ($\phi_u(\cdot) \in \Phi$). Furthermore, each sensor device is equipped with the same one-way hash function $H^k(x)$, which returns the hashed value based on the most significant k bits of x.

3.3 Pair-wise Key Establishment and Rekeying

After the key pre-assignment phase, wireless sensors are randomly distributed in a given area, and later on, some clustering algorithm, *e.g.*, (Heinzelman et al., 2002), shall organize the network into a hierarchical structure. The following intra-cluster protocol, as illustrated in Figure 1, is to establish the new pair-wise key between a cluster head a and one of its member sensor nodes i in a new round of rekeying phase, in which the orignal pair-wise key establishment is treated the same as the subsequent rekeyings. The inter-cluster rekeying protocol for CH-CH links works in a similar manner and thus is omitted here.

- Step 1: At the beginning of each rekeying phase, CH_a randomly generates a new t-degree univariate rekeying polynomial function $p_{CH_a}(y)$. For each of its sensor node SN_i, CH_a updates the corresponding pair-wise key K_{CH_a,SN_i} as

$$K_{CH_a,SN_i} = H^{\ell-r}(p_{CH_a}(SN_i)). \tag{7}$$

- Step 2: CH_a uses $p_{CH_a}(y)$ and the preloaded polynomial $\overline{g}_{CH_a}(y)$ to construct a master polynomial $w_{CH_a}(y)$:
$$w_{CH_a}(y) = p_{CH_a}(y) + \overline{g}_{CH_a}(y) \tag{8}$$
and broadcasts its ID CH_a and this polynomial $w_{CH_a}(y)$ to all its sensor nodes by a single transmission.
- Step 3: Upon receiving the broadcast message, each SN_i evaluates the preloaded polynomial $\overline{g}_{SN_i}(y)$ at $y = CH_a$ and evaluates the receieved master polynomial $w_{CH_a}(y)$ at $y = SN_i$. After that, three candidate keys $K^*_{CH_a,SN_i}$, $K^+_{CH_a,SN_i}$ and $K^-_{CH_a,SN_i}$ will be calculated as follows, respectively.
$$K^*_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a)\right) \tag{9}$$
$$K^+_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) + 2^r\right) \tag{10}$$
$$K^-_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) - 2^r\right) \tag{11}$$
- Step 4: At a later time, a encoded information $E(\text{msg}, K_{CH_a,SN_i})$ will be piggybacked in a normal unicast message sent from CH_a to SN_i. The exact new pair-wise key is determined by SN_i once such message can be decoded successfully using one of the candidate keys.

Note that due to the characteristic of the perturbation polynomial (Zhang et al., 2007), only one of the candidate keys (9) - (11) will be validated as the new pair-wise key between SN_i and CH_a, *i.e.*,
$$K_{CH_a,SN_i} \in \left\{K^*_{CH_a,SN_i}, K^+_{CH_a,SN_i}, K^-_{CH_a,SN_i}\right\}. \tag{12}$$
The unicast message can be also sent from SN_i to CH_a. Under this circumstance, the new pair-wise key will be calculated at SN_i as $K_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a)\right)$, while three candidate keys will be evaluated at CH_a as $K^*_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i)\right)$, $K^+_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i) + 2^r\right)$, and $K^-_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i) - 2^r\right)$. All remaining rekeying processes are the same and conclusion in (12) will be also made.

3.4 Examples

To help understand the details of our rekeying protocol, we provide the following simplified example with $CH_a = 3$ and $SN_i = 2$. In system initialization, we set $q = 127$, $t = 2$, $\ell = 7$, and $r = 3$. All arithmetic operations are over finite field F_{127}. The bivariate symmetric polynomial is $f(x,y) = xy^2 + x^2y + 2xy + 5$ and the corresponding univariate polynomials for CH_a and SN_i are $g_3(y) = f(3,y) = 3y^2 + 15y + 5$ and $g_2(y) = f(2,y) = 2y^2 + 8y + 5$, respectively. Now, we consider the following cases in a rekeying phase, in which CH_a generates a new univariate polynomial function $p_3(y) = 3y^2 + 15y + 9$ under different preloaded perturbed polynomials.

Case 1: Suppose the perturbation polynomials for CH_a and SN_i are $\phi_3(y) = y^2 - 3y + 5$ and $\phi_2(y) = y^2 - 4y + 5$, respectively. Note that both polynomials satisfy the limited infection property: $\phi_3(2) = 3 \in \{0,1,\cdots,7\}$ and $\phi_2(3) = 2 \in \{0,1,\cdots,7\}$. Their preloaded polynomials are therefore $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 12y + 10$ and $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + 4y + 10$, respectively, as illustrated in Figure 2. In rekeying, CH_a calculates the new pair-wise key as $K_{3,2} = H^4(p_3(2)) = H^4(51) = H^4(0110011)$ and

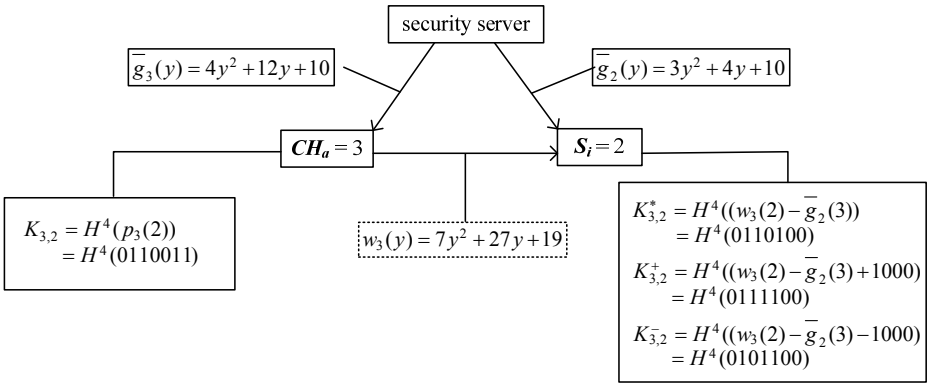


Fig. 2. Example of $K_{CH_a,SN_i} = K^*_{CH_a,SN_i}$

sends the master polynomials $w_3(y) = p_3(y) + \overline{g}_3(y) = 7y^2 + 27y + 19$ to SN_i. At SN_i side, it then calculates three candidate keys: $K^*_{3,2} = H^4(w_3(2) - \overline{g}_2(3)) = H^4(52) = H^4(0110100)$, $K^+_{3,2} = H^4(60) = H^4(0111100)$, and $K^-_{3,2} = H^4(44) = H^4(0101100)$. We observe that $K_{CH_a,SN_i} = K^*_{CH_a,SN_i}$ ($H^4(0110011) = H^4(0110100)$) is achieved.

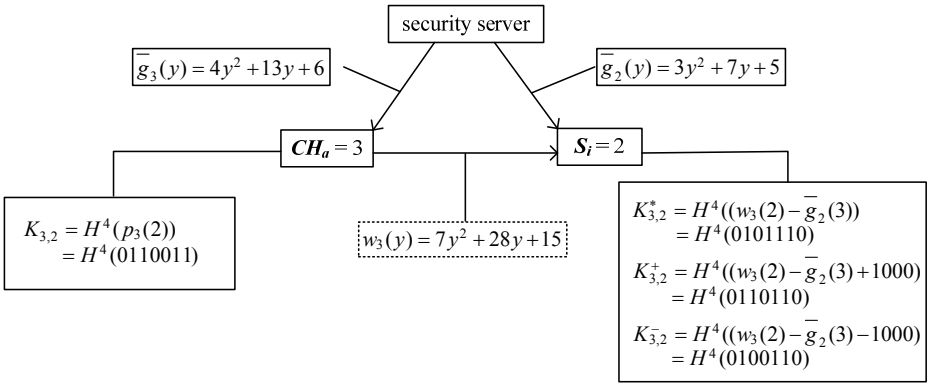


Fig. 3. Example of $K_{CH_a,SN_i} = K^+_{CH_a,SN_i}$

Case 2: Under different perturbation polynomials $\phi_3(y) = y^2 - 2y + 1$ ($\phi_3(2) = 1$) for CH_a and $\phi_2(y) = y^2 - y$ ($\phi_2(3) = 6$) for SN_i, we can obtain $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 13y + 6$, $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + 7y + 5$, and $w_3(y) = p_3(y) + \overline{g}_3(y) = 7y^2 + 28y + 15$. Eventually, we observe $K_{CH_a,SN_i} = K^+_{CH_a,SN_i}$ ($H^4(0110011) = H^4(0110110)$) as shown in Figure 3.

Case 3: Similarly, the perturbation polynomials $\phi_3(y) = y^2 - 6y + 14$ ($\phi_3(2) = 6$) and $\phi_2(y) = y^2 - 7y + 13$ ($\phi_2(3) = 1$) are for CH_a and SN_i, respectively. We then obtain $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 9y + 19$, $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + y + 18$, and $w_3(y) = p_3(y) + \overline{g}_3(y) =$

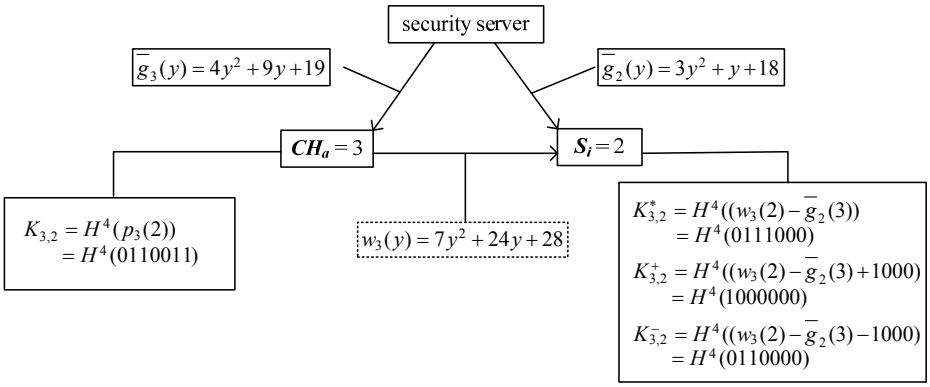


Fig. 4. Example of $K_{CH_a,SN_i} = K^-_{CH_a,SN_i}$

$7y^2 + 24y + 28$. The final case $K_{CH_a,SN_i} = K^-_{CH_a,SN_i}$ ($H^4(0110011) = H^4(0110000)$) is shown in Figure 4.

4. Security Analysis

In this section, we give a security analysis for our proposed rekeying scheme and compare it to other proposals in terms of robustness to the node capture attack.

4.1 Breaking Rekeying Polynomial $p_{CH_a}(y)$

We assume that an adversary has compromised n_c sensor nodes in cluster a, denoted as CS_k ($k = 1, \cdots, n_c > t$), and has obtained all their preloaded information.

To derive the polynomial $p_{CH_a}(y)$ that is used to generate the new pair-wise key as shown in (7), the adversary needs to break $\overline{g}_{CH_a}(y)$ because $p_{CH_a}(y) = w_{CH_a}(y) - \overline{g}_{CH_a}(y)$, in which $w_{CH_a}(y)$ is the public information broadcasted by CH_a. Furthermore, for any sensor node y of CH_a, the corresponding pair-wise key $K_{CH_a,y}$ satisfies:

$$\begin{aligned} K_{CH_a,y} &= H^{\ell-r}\left(w_{CH_a}(y) - \overline{g}_{CH_a}(y)\right) \\ &= H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y) - \phi_{CH_a}(y)\right) \\ &= \begin{cases} H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y)\right), & \text{or} \\ H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y) - 2^r\right). \end{cases} \end{aligned}$$

The above equation shows that to break $\overline{g}_{CH_a}(y)$ is equivalent to break $g_{CH_a}(y)$ or $f(CH_a, y)$. This can be done by collecting a number of polynomials $\overline{g}_{CS_k}(y)$ stored in the compromised sensor nodes, which satisfy

$$\overline{g}_{CS_k}(y) = f(CS_k, y) + \phi_{CS_k}(y). \tag{13}$$

It can be formulated as a linear equation system as follows.

$$\sum_{i=0}^{t} a_{ij} \cdot (CS_k)^i + b_{kj} = d_{kj}, 0 \le j \le t, 1 \le k \le n_c \tag{14}$$

Note that a_{ij} and b_{kj} are the variables of this linear equation system, which are defined by (1) and the following equation

$$\phi_{CS_k}(y) = \sum_{j=0}^{t} b_{kj} \cdot y^j, 1 \leq k \leq n_c, \tag{15}$$

respectively. On the other hand, the values of d_{kj} are known to the adversary:

$$\overline{g}_{CS_k}(y) = \sum_{j=0}^{t} d_{kj} \cdot y^j, 1 \leq k \leq n_c. \tag{16}$$

By applying a similar reasoning technique in (Zhang et al., 2007), we can derive that the probabilities to find the solution of the linear equation system (14) in one attempt is $m^{-(t+1)}$, in which m is the total number of perturbation polynamials, *i.e.*, $m = |\Phi| \geq 2$. In other words, to break $f(x, y)$, or $g_{CH_a}(y) = f(CH_a, y)$, in one attempt is $m^{-(t+1)}$. Finally, we can conclude that the computational complexity for breaking $p_{CH_a}(y)$ under the condition of $t + 1$ compromised nodes is $\Omega\left(m^{t+1}\right)$.

4.2 Node Capture Attack

After deployment, each cluster head and each sensor node can be captured and compromised by attackers due to the unattended deployment environments and their lack of tamper-resistance. The adversary can read out all information stored in the node to get all secret information. In addition, the attackers may collect the secrets owned by compromised nodes, and attempt to derive the secrets held by innocent nodes (and therefore can cheat these innocent nodes or impersonate as them). This is the well-known node capture attack.

In the Chadha's scheme (Chadha et al., 2005), each sensor node SN_i is pre-loaded a $2t$-degree masking polynomial $h(x)$ in its storage. After $2t$ sensor nodes are compromised, the whole network will crash. In our proposed pair-wise rekeying protocol, in order to derive the rekeying polynomial $p_{CH_a}(y)$ of cluster head a, the adversary needs to break the original symmetric polynomial $f(x, y)$ with extremely low probability.

Assume that the degree of polynomial function is $t = 80$, the NCA-robustness comparison of these two protocols are illustrated in Figure 5. As we observe that after a number of sensor nodes are compromised, Chadha's schemes will disclose the polynomials that can generate any group key in the past or future. On the contrary, our proposed scheme can achieve both forward and backward secrecy because such polynomials are extremely hard to be broken in our approach.

5. Performance Analysis

In this section, we evaluate the performance of our proposal by comparing with Chadha's scheme (Chadha et al., 2005). The performance metrics include the computational complexity, communication overhead, and storage overhead. Table 2 summarizes the performance results. In the Chadha's scheme, each cluster head first constructs $w(x) = g(x)f(x) + h(x)$ and calculates $n_a - n_c$ pair-wise keys for all innocent nodes, in which n_a and n_c are number of all sensor nodes and compromised sensor nodes, respectively, in a cluster. It needs $O(n_c^2 + n_c t + (n_a - n_c)t) = O(n_c^2 + n_a t)$ multiplications. Upon receiving $w(x)$, each sensor node needs to derive its personal key using $O(t)$ multiplications. In our proposed pair-wise

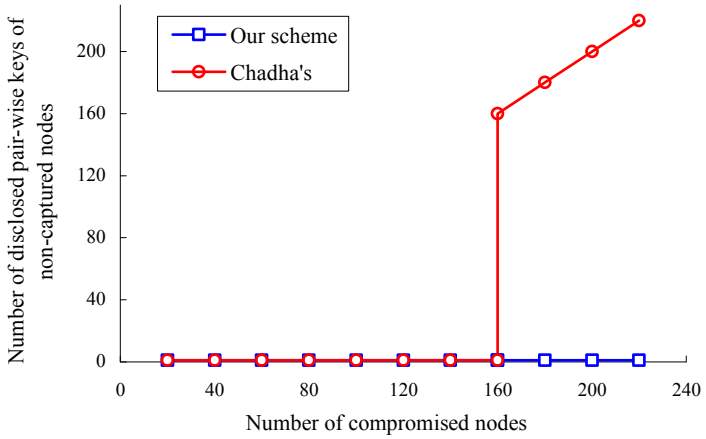


Fig. 5. NCA robustness comparison ($t = 80$)

		Chadha's	Our Scheme
Computation	Cluster head	$O(n_c^2 + n_a t)$ mul.	$O((n_a - n_c)t)$ mul. $n_a - n_c$ hash fun.
	Sensor node	$O(t)$ mul.	$O(t)$ mul. 3 hash fun.
Communication	Cluster head	$(2t + n_c + 1) \cdot \ell$	$(t+1) \cdot \ell$
	Sensor node	0	0
Storage	Cluster head	$(2t+1) \cdot \ell$	$(t+1) \cdot \ell$
	Sensor node	ℓ	$(t+1) \cdot \ell$

Table 2. Performance analysis

rekeying scheme, each cluster head needs to recalculate $n_a - n_c$ pair-wise keys using the rekeying polynomial with $O((n_a - n_c)t)$ multiplications. Each key generation involves a hash function operation as well. For each sensor node, it needs to calculate three candidate keys, which takes $O(t)$ multiplications and 3 hash function operations.

In the Chadha's scheme, each cluster head broadcasts a new $2t$-degree polynomial $w(x)$ and n_c Ids of detected compromised nodes to all the sensor nodes in the cluster. Such broadcast message has $(2t + n_c + 1) \cdot \ell$ bits. No message transmission at sensoe node side. The only communication overhead in our proposed scheme is the broadcast message for sending the t-degree master polynomial with $(t+1) \cdot \ell$ bits. Note that, the overhead of the piggybacked short message for key agreement are considered as normal traffic and not included in Table 2.

In the evaluation of storage overhead, we consider the space requirement of the preloaded information in each sensor node and cluster head for the rekeying schemes. In Chadha's scheme, each cluster head is pro-loaded a $2t$-degree masking polynomial function $h(x)$. All coefficients for the polynomial require $(2t+1) \cdot \ell$ bits. Each sensor node S_i needs to store one secret values $h(S_i)$ with ℓ bits. In our scheme, each sensor device (both cluster head and sensor node) is preloaded one t-degree perturbed polynomial taking $(t+1) \cdot \ell$ bits.

6. Conclusion

The traditional polynomial based pair-wise rekeying protocol suffers the large-scale node capture attack. Once $t+1$ nodes are compromised, all previous and future keys for any pair of nodes will be disclosed. We present a compromise-resilient pair-wise rekeying scheme based on a three-tier WSN. It can significantly improve the security level by reducing this probability from 1 down to $m^{-(t+1)}$ ($m \geq 2$). Our proposed scheme also achieves both forward and backward secrecy.

7. References

Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor Networks: A Survey, *Journal of Computer Networks*, Vol. 38, No. 4, 393–422.

Blundo, C.; De Santis, A.; Herzberg, A.; Kutten, S.; Vaccaro, U. & Yung, M. (1993). Perfectly-secure key sistribution for dynamic conferences, *LNCS*, Vol. 740, 471–486.

Chadha, A.; Liu, Y. & Das, S. (2005). Group key distribution via local collaboration in wireless sensor, *IEEE SECON*, pp. 46–54, July 2005.

Cheng, Y. & Agrawal, D. P. (2005). Efficient pairwise key establishment and management in static wireless sensor networks, *IEEE MASS*, November 2005.

Cheng, Y. & Agrawal, D. P. (2007). A improved key distribution mechanism for large-scale hierarchical wireless sensor networks, *Journal of Ad Hoc Networks*, Vol. 5, No. 1, 35–48.

Diffie, W. & Hellman, M. E. (1976). New direction in cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 644–654.

Du, W. L.; Deng, J.; Han, Y.& Varshney, P. K. (2003). A pairwise key pre-distribution scheme for wireless sensor network, *ACM Conference on Computer and Communications Security*, pp. 42–51, October 2003.

Eschenauer, L. & Gligor, V. (2002). A key-management scheme for distributed sensor networks, *ACM CCS*, pp. 41–47, November 2002.

Heinzelman, W. R.; Chandrakasan, A. P. & Balakrishnan, H. (2002). An application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, 660–670.

Mishra, S. (2002). Key management in large group multicast, *Technical Report CU-CS-970-02*, University of Colorado.

Rivest, R.; Shamir, A. & Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems, *Communications of ACM*, Vol. 21, No. 2, 120–126.

Zhang, W.; Song, H.; Zhu, S. & Cao, G. (2005). Least privilege and privilege deprivation: Towards to tolerating mobile sink compromises in wireless sensor networks, *ACM MobiHoc*, pp. 378–389, May 2005.

Zhang, W.; Tran, M.; Zhu, S. & Cao, G. (2005). A random perturbation-based scheme for pairwise key establishment in sensor networks, *ACM MobiHoc*, pp. 90–99, September 2007.

Zhang, W.; Subramanian, N.; Zhu, S. & Wang, G. (2005). Lightweight and compromise-resilient message authentication in sensor networks, *IEEE INFOCOM*, pp. 1418–1426, April 2008.

Security architecture, trust management model with risk evaluation and node selection algorithm for WSN

Bin Ma[1,2] and Xianzhong Xie[1,2]
[1] *School of computer science and technology, Chongqing University of Posts and Telecommunications*
[2] *Institute of Personal Communications, Chongqing University of Posts and Telecommunications*
P.R. China

1. Introduction

Wireless sensor networks are ideal candidates to monitor the environment in a variety of applications such as military surveillance, forest fire monitoring, etc. In such a network, a large number of sensor nodes are deployed over a vast terrain to detect events of interest (e.g., enemy vehicles, forest fires), and deliver data reports over multihop wireless paths to the user. Security is essential for these mission-critical applications to work in an adverse or hostile environment.

Wireless Sensor networks are typically characterized by limited power supplies, low bandwidth, small memory sizes and limited energy. This leads to a very demanding environment to provide security. Public-key cryptography is too expensive to be usable, and even fast symmetric-key ciphers must be used sparingly. Communication bandwidth is extremely dear: each bit transmitted consumes about as much power as executing 800–1000 instructions(J. Hill et al 2000), and as a consequence, any message expansion caused by security mechanisms comes at significant cost.

Wireless sensor networks consist of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. In addition to one or more sensor nodes, each node in wireless sensor networks is typically equipped with a radio transceiver or other wireless communication devices, a microcontroller, and an energy source, usually a battery.

Wireless sensor networks are the connection between physical world and mankind, which cannot be simply regarded as communication networks. It should mainly concentrate on sensory information processing and services. Wireless sensor networks should be developed as an integrated information infrastructure, in which information aggregation and collaborative processing are key issues.

And so, all nodes share common sensing tasks in wireless sensor networks. This implies that not all sensors are required to perform the sensing task during the whole system lifetime. Turning off some nodes does not affect the overall system function as long as there are enough working nodes to assure it. Therefore, if we can schedule sensors to work alternatively, the system lifetime can be prolonged by exploiting redundancy. In this chapter,we present a cross-layer trust management model based on cloud model; and using the trust model, we innovate an algorithm of node selection in Wireless sensor networks.
The rest of the chapter is structured as follows. In the beginning we introduce wireless sensor networks. Furthermore, A discussion of related work for security architecture and trust management model. Thereafter, we provide a unique security requirements of WSNs and present a security architecture for wireless sensor networks that addresses most of the problems above, also describe the technical aspects of our security architecture. Subsequently, we utilizes lightweight trust management model that allow for easy access control between the mobile sensor nodes and secure the communication inside the network. Furthermore, it minimizes the effects of compromised sensor nodes.

2. Related Works

2.1 security architecture

Security in sensor networks has been studied by several other researchers. Perrig et al(2001). developed the security architecture SPINS, which is based on the two protocols SNEP, a protocol for data confidentiality, two-party data authentication, and data freshness and μTESLA, a broadcast authentication protocol.Their architecture relies on the concept, that every node shares a secret key with a trusted base station, which is at all times able to communicate with every node in the network.
Furthermore, several key management schemes have been put forward for sensor networks: Basagni et al(2001). proposed a solution to periodically update a symmetric key which is shared by all nodes in the network. Their solution is based on the assumption that all nodes are constructed tamper-proof, which is not always the case. Carman et al(2000). studied several key management protocols in sensor networks with respect to performance on different hardware platforms. Zhu et al(2003). proposed the Localized Encryption and Authentication Protocol(LEAP) which utilizes four types of keys for each node. These are used for different purposes and range from the individual key that is shared with the base station, up to a group key that is shared with all nodes in the network. Eschenauer and Gligor(2002) presented a pool-based random key predistribution system, which Chan et al.(2003) extended by presenting three new mechanisms for key establishment.
Wood and Stankovic(2002,2003) identified several DoS attacks in sensor networks and presented a protocol, which allows to map regions that are subject to DoS by radio jamming.

2.2 trust management model

The traditional trust management systems are suitable for wired and wireless ad hoc network, but cannot satisfy the security requirements of wireless sensor network. Because they need very large resources consumption which is wireless sensor network lacked.
The trust management system may be the centralism or the distribution, but they both do not suit sensor network, the central system needs enough energy to satisfy the extra route need, but in the distributional system, each node needs enough storage space and strong computing

power. But in the sensor network, all node joint operation as if is more realistic. Therefore, the mix low consumption trust management system can satisfy the demand of sensor network.
Since Marsh(1994) introduced the research of trust to the computer domain, trust mechanism has gradually obtained more and more researcher's(Blaze M 1996, Adrian Perrig 2001, Sasha Slijepcevic 2002, and so on) values for its flexibility and extendibility. The people proposed the numerous trust models in distribution network, pervasive computing, peer-to-peer computing, ad hoc network and so on. In these models, trust is usually quantified as a definite real number. However, because the node trust has much subjectivity, natural insufficiency has existed by using the definite value to describe trust. For example, if node A trusts node B, it is very difficult to determine that the trust value should be 0.9 is 0.8. Therefore, uncertainty is considered to be the important attribute of trust, namely trust among the node is fuzziness and randomness; especially among strange node. Therefore, uncertainty must be considered when trust model build. Based on this, a cross-layer wireless sensor network trust model based on cloud model is proposed. This model unifies the description of trust degree and uncertainty of trust relationship among the nodes with trust cloud forms, and gives algorithms of trust cloud transmission and merge.
The cloud model by Deyi Li et al(2000,2004) has first proposed as the qualitative description and the quota expressed of one kind of terminology. It unifies the fuzziness and randomness, thus describing the uncertainty well. Now, the cloud model has already applied in numerous domains, like data mining, automatic control, quantitative evaluation and so on.

3. Security architecture

3.1 The security requirement of wireless sensor networks

Wireless sensor networks are composed of massive sensor nodes. These nodes are small, cheap, battery power supply, and have the ability of wireless communication and monitor. All the nodes are deployed densely in the monitored region to monitor the Physical world.
Because the sensor nodes mostly are deployed in the enemy or nobody region, sensor network security problem is prominent especially. Lacking effective safety mechanism already becomes the chief obstacle of the sensor network application.
Wireless sensor network's own characteristic (the limitation of computation, communication and memory, lacks of the apriority to nodes deploying, unreliable Physical security of deployed region as well as dynamic change of network topology and so on) enables the sensor network except to have the traditional network security requirements, but also has some specific security property.

Data Confidentiality

The sensor network should not reveal the information to the neighbor network. In many applications, the node transmits the highly confidential data. The standard method to protect data confidentiality is enciphered data with the key, the receiver can decipher data, therefore achieves confidentiality, establish the security channel among the nodes according to the communication mode.

Data Authentication

In the sensor network, message authentication is important to many applications. When the network is constructed, authentication to the management task is necessary. At the same

time, the enemy is very easy to insert information ,so the receivers need to determine the reliability of message's origin. The data authentication permit data confirmation that the receivers is the sender who declared sends out.
In two nodes communication, the data authentication may be achieved through the symmetrical mechanism: Sender and receiver share one key to calculate the messages authentication code (MAC) of all communication data. When the message arrived with the correct MAC, the receiver can be sure that the message indeed is the real sender sends out.

Data Integrity
In the communication, the data integrity guarantee all the data that receivers receive in transmission process not be changed by enemy. The data integrity may achieve through the data authentication.

Data Freshness
All data survey of sensor network is related with the time, cannot guarantee the confidentiality and the authentication sufficiently, but must certainly guarantee that each message is fresh. The data freshness implied the data is recent, and guaranteed that the enemy have not replay the information before. There are two types of freshness: The weak freshness provides the partial information order, but does not carry any delay information; the strong freshness provides complete order of the request/response, and permit delay forecast. The sensation survey needs the weak freshness, but in the network time synchronism needs the strong freshness.

Key management
In order to realize, satisfy the above security requirements, the encryption key needs to be managed. As a result of the energy and the computing limit, wireless sensor networks needs to maintain balanced between the security rank and these limits. Key management should include the key allocation, the initialization stage, the node increase, the key abolishment, the key renewal.
All in all, The security requirement of wireless sensor networks is main list:

1) As the key feature of wireless sensor network applications, the diversity of sensors, data flow and QoS requires the system architecture be of compatibility, universality and scalability to meet the various requirements.
2) The prevailing studies on wireless sensor networks focus on the solution of low data rate, short packet burst, low network traffic and low device energy issues. Many standardization organizations have been working on the standards of PHY/MAC layers, network protocol, identifier and sensor interfaces, however the completed security solutions on various layers have not been found out.
3) In wireless sensor network applications, such as anti-intrusion, public security, and environment monitoring, various sensors have to work cooperatively, while the current solution cannot meet the requirements.
4) The main purposes of wireless sensor networks are information sensing and processing. Thus, the security of information cooperative processing scheme in wireless sensor networks must be considered in the architecture design.

3.2 Security issues of each layers in wireless sensor networks

The network protocol stack of wireless sensor networks is composed of physical layer, data link layer, network layer, transmission layer and application layer.

Each function as follows:
Physical layer is responsible for the frequency selection, the carrier frequency production, the signal detection and the data encryption, the layer include modulation, transmission, receive and data encryption technology.
Data link layer is used for establishing communication link of reliable point-to-point or point to multipoint.
Network layer is primary responsible for route production and routing.
Transmission layer is used to establish end-to-end link between wireless sensor network and Internet or other exterior networks.
Application layer has provided kinds of practical applications of wireless sensor network.

Security problem of each layer:
Security of physical layer is how to establish the effective data encryption mechanism. Due to the property of sensor network, low expenses cryptography algorithm is still a hot spot in sensor network security research.
Data link layer or medium access control (MAC) layer provides the reliable correspondence channel for the neighbor node which is easy to come under the DOS attack. The solution is regulating the MAC admittance control, and the network neglects excessively requests automatically.
Network layer is easy to come under the attack, because each node is the latent route node, security routing algorithm immediate influence security and usability of wireless sensor network. Application layer's research mainly concentrates in providing the safe support for the entire wireless sensor network, is also the key management and the security multicast research.
Overall approach of sensor network security ensure that all layers' security, this solution could be the best option than a single security for a single layer.

3.3 Stereoscopic security architecture of wireless sensor networks

Wireless sensor network is easy to come under each kind of attack, and has many hidden security problems. At present the quite general sensor network security architecture divides the sensor network protocol stack into hardware layer, operating system layer, middleware layer and application layer. Its security module has divided into 3 layers: security primitive, security service and security application. This security architecture divided the security problem into three levels, it have the advantages of succinct question description, agreement distinctive nuance merit, but there are some general security problem among them, it could not place some security protocols in some layer to solve forcefully; And this architecture can not solve deceit of evil intention node, it have enormous hidden security problems.
With deep research on the sensor network security demand and each layer's security problem's, as well as experiences of our topic-based group, and linking the original wireless sensor network architecture, we proposed stereoscopic wireless sensor network security architecture as shown in Fig.1. This network security architecture is composed of hierarchical network communication and security protocol and the wireless sensor network support technology. The hierarchical network communication and security protocol structure is similar to the TCP/IP protocol architecture; the wireless sensor network support technology is mainly to sensor node own management as well as the user to the wireless sensor's management; two partial protocols and the technology has overlapping and the union, and have formed a cubic structural model.

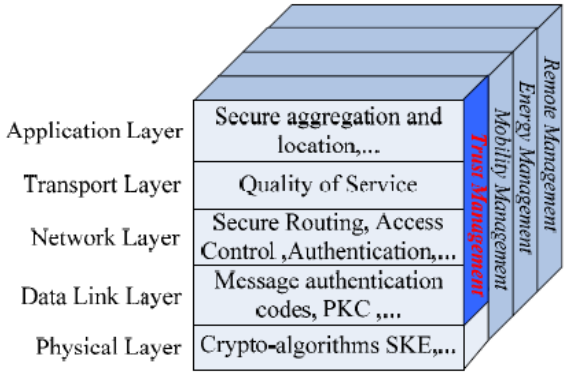


Fig. 1. security architecture of wireless sensor networks

4. Trust management model with risk evaluation

The traditional trust management systems are suitable for wired and wireless ad-hoc network, but cannot satisfy the security requirements of wireless sensor network. Because they need very large resources consumption which is wireless sensor network lacked.

The trust management system may be the centralism or the distribution, but they both do not suit sensor network, the central system needs enough energy to satisfy the extra route need, but in the distributional system, each node needs enough storage space and strong computing power. But in the sensor network, all node joint operation as if is more realistic. Therefore, the mix low consumption trust management system can satisfy the demand of sensor network.

Since Marsh introduced the research of trust to the computer domain, trust mechanism has gradually obtained more and more researcher's values for its flexibility and extendibility. The people proposed the numerous trust models in distribution network, pervasive computing, peer-to-peer computing, ad hoc network and so on. In these models, trust is usually quantified as a definite real number. However, because the node trust has much subjectivity, natural insufficiency has existed by using the definite value to describe trust. For example, if node A trusts node B, it is very difficult to determine that the trust value should be 0.9 is 0.8. Therefore, uncertainty is considered to be the important attribute of trust, namely trust among the node is fuzziness and randomness; especially among strange node. Therefore, uncertainty must be considered when trust model build. Based on this, a cross-layer wireless sensor network trust model based on cloud model is proposed. This model unifies the description of trust degree and uncertainty of trust relationship among the nodes with trust cloud forms, and gives algorithms of trust cloud transmission and merge.

The cloud model has first proposed as the qualitative description and the quota expressed of one kind of terminology. It unifies the fuzziness and randomness, thus describing the uncertainty well. Now, the cloud model has already applied in numerous domains, like data mining, automatic control, quantitative evaluation and so on.

This part of chapter uses the concept of cloud model to estimate dynamic context and consequently presents the definition of risk signal, and a trust management model based on risk evaluation for wireless sensor networks is proposed. The risk is evaluated using cloud model, quantified using risk and trust uncertainty degree are presented in a uniform form. The simulation results show that the proposed trust model based on risk evaluation can

efficiently expressed uncertainty of risk and trust, and decreased trust risk of nodes. And so this trust model also can evidently taked from the rate of trust risk, and enhanced successful cooperation ratio of WSN's system.

4.1 Cloud model

Cloud model was firstly proposed as a model of the uncertainty transition between a linguistic term of a qualitative concept and its numerical representation. In short, it is the model of the uncertainty transition between qualitative concept and quantitative description. In the discourse universe, the cloud mainly reflects two uncertainties: the fuzziness (the boundary character of both this and that) and the randomness (occurrence probability). The cloud model completely integrates the fuzziness and randomness, researches the uncertain rules which have contained by basic linguistic term(or linguistic atom) in natural language, that not only is possible to obtain the scope and distribution rule of quantitative data, but also may effectively transform precise number to qualitative linguistic term.

Formally, a cloud can be defined as follows.

Defines 1: Let U be the set as the universe of discourse, μ is a random function with a stable tendency $\mu : U \to [0,1]$,and g is also a random function with a stable tendency $g : U \to U$,He is an uncertain factor and 0…He, and

1) $u' = g(u, He), u \in U$
2) $y = \mu(u', He)$

then (U, g, μ, He) is a cloud, and (u', y) is a cloud drop.

The bell-shaped clouds, called normal clouds are most fundamental and useful in representing linguistic terms, see Fig. 2. A normal cloud is described with only three digital characteristics, expected value(Ex), entropy(En) and hyper entropy(He).

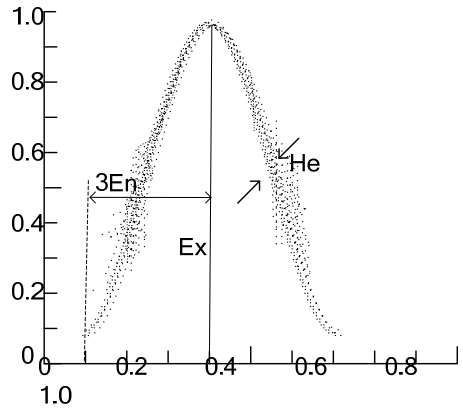


Fig. 2. Normal Cloud with digital characteristic

The expected value Ex of a cloud is the position at the universe of discourse, corresponding to the center of gravity of the cloud. In other words, the element Ex in the universe of discourse fully belongs to the linguistic term represented by the cloud model. The entropy, En, is a

measure of the fuzziness of the concept over the universe of discourse showing how many elements in the universe of discourse could be accepted to the linguistic term. It should be noticed that the entropy defined here is a generic notion, and it need not be probabilistic. The hyper entropy, He, is a measure of the uncertainty of the entropy En. Close to the waist of the cloud, corresponding to the center of gravity, cloud drops are most dispersed, while at the top and bottom the focusing is much better. The discrete degree of cloud drops depends on He.
Given three digital characteristics Ex, En, and He, to represent a linguistic term, a set of cloud drops may be generated by the following algorithm:

Algorithm 1: Forward Cloud Generator Algorithm
Input: *the expected value of cloud Ex,*
the entropy of cloud En,
the hyper entropy of cloud He,
the number of drops N.
Output: *a normal cloud with digital characteristics Ex, En, and He.*

1) *Produce a random value x which satisfies with the normal distribution probability of mean=Ex, and standard error = En;*
2) *Produce a random value En' which satisfies with the normal distribution probability of mean = En, and standard error = He;*
3) *Calculate*

$$y = \exp\left[-\left(x_i - Ex\right)^2 \Big/ 2\left(En'\right)^2 \right] \tag{1}$$

4) *Let (x, y) be a cloud drop in the universe of discourse;*
5) *Repeat 1-4 until the number of drops required all generated.*

The idea of using only three digital characteristics to generate a cloud is creative. The generator could produce as many drops of the cloud as you like (Fig. 2). This kind of generators is called a forward cloud generator. All the drops obey the properties described above. Cloud-drops may also be generated upon conditions. It is easy to set up a half-up or half-down normal cloud generator with the similar strategy, if there is a need to represent such a linguistic term. It is natural to think about the generator mechanism in an inverse way. Given a number of drops, as samples of a normal cloud, the three digital characteristics Ex, En, and He could be obtained to represent the corresponding linguistic term. This kind of cloud generators may be called backward cloud generators. Since the cloud model represents linguistic terms, the forward and backward cloud generators can be served interchangeably to bridge the gap between quantitative and qualitative knowledge.
Backward cloud gennerators are the uncertainty transformation model realizing the transformation between a numeric value and it's linguistic value, in other words, the mapping between quantitative and qualitative representation. It effectively converts a certain number of accurate data to the concept indicated by appropriate qualitative linguistic values(Ex,En,He) which represent the character of the whole drops.

In this chapter, backward cloud algorithm without certainty is adopted. The steps are presented as follows:

Algorithm 2: Backward Cloud Generator Algorithm

Input : $x_i(i=1,2,3...,n)$;

Output : *(Ex,En,He)* ;

1) *Calculate the mean value of* x_i *,*V、 *the first order absolute central moment* M_1 *, and the variance of* x_i *,* M_2 ;

2) *Compute the expectation of* x_i *,* $Ex = V$;

3) *Compute the entropy of* x_i *,* $En = M_1 \times \sqrt{\frac{\pi}{2}}$;

4) *Compute the entropy of* En *,* $He = \sqrt{M_2 - En^2}$ 。

4.2 Trust definition

4.2.1 Risk evaluation based on cloud model

In wireless sensor network environment, entity could observe dynamic variation of context information, then feel risk. It was series approve transmit, thereof function curve too COMPare intricacy, inconvenience to with derivative 'formal description that even by surveillant dynamic context information sometimes nope series derivable, even if.Whereas uncertainty of risk, This chapter based on cloud model describe dynamic variation of context information .At known context normal state,using backward cloud algorithm without certainty protract context normal cloud, and got the digital characteristics.Compute is kept watch on the belonging to of context information sample value of time degree, if the context information that this at that time engraves samples a value to belong to normal appearance cloud and thinks to have no risk creation, whereas, think risk signal creation.The description like this even has general.

Defines 2: **context information cloud:** $Cloud = (I, t, Ex, En, He, \delta)$

Here : $I = \{S, E, C, R, U, \cdots\}$: *I* means aggregate of context information by watching.

t : Context information of sample partition time.

Ex : Sample point that have already known is regarded as cloud drop, we adopt the expectation value of context information cloud with the backward cloud generator. This expectation value is named the gravity of cloud. In this place, context information is accepted normal. $Ex \approx \hat{E}x = \overline{M} = \frac{1}{n}\sum_{i=1}^{n} m_i$.

En : The principle is above, $En \approx \hat{E}n = \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n} \left| m_i - \hat{E}x \right|$.

He : The principle is above, $He \approx \hat{H}e = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} \left(m_i - \overline{M}\right)^2 - \hat{En}^2}$.

δ : membership grade valve.

Defines 3: membership grade function definition of context information cloud, assume *m* is sampling value of context information *S* at hours *T*, and *m* that is computed by formula (2) could be known as normal degree of certainty μ :

$$\mu = e^{-\frac{(m-Ex)^2}{2(En)^2}} \tag{2}$$

$\mu > \delta$: The context information value of T time belongs to normal scope and have no risk signal creation;

$\mu < \delta$: The context information value of T time doesn't belongs to normal scope and have risk signal creation.

Above all of the risk signals is according to single context information, only with a single context information creation of the risk signal is not enough to predicate risk of occurrence in whole system. And so, we need to synthesize various risk signals of context informations to synthesize judgment.This chapter gives the evaluation method of risk.

Defines 4: definition of " Risk" : $Risk = (I', Q, \varphi)$

Here : I' is meaning that I aggregate of context information correspond with risk information

$Q = \{q_1, q_2, \cdots q_n | 0 < q_i < 1\}$: respectively representation each proportion of context information risk signal in whole sysytem.

φ : risk vavle

Risk= $S' \times q_1 + E' \times q_2 + C' \times q_3 + R' \times q_4 + \cdots > \varphi$: have risk occurrence ;

Risk= $S' \times q_1 + E' \times q_2 + C' \times q_3 + R' \times q_4 + \cdots \leq \varphi$: the context is normal and have no risk occurrence。

4.2.2 Trust cloud

Trust cloud is the core concept of the model. Based on the formalized definition of the cloud, its formalized definition is given as follows:

Defines 5: The trust cloud is the description of trust relationship among nodes with One-Dimensional Normal Cloud forms, it indicates is:

$$\left.\begin{array}{c} tc_{AB} = nc(\mathrm{Ex}, \mathrm{En}, \mathrm{He}, \mathrm{Risk}) \\ 0 \leq \mathrm{Ex} \leq 1, 0 \leq \mathrm{En} \leq 1 \\ 0 \leq \mathrm{He} \leq 1, 0 \leq Risk \leq 1 \end{array}\right\} \tag{3}$$

That is, trust is a normal cloud among the nodes, Ex is trusts expectation., it indicates basic trust value of node A to the B; En is trust entropy, it reflects uncertainty of the trust relationship; He is trust ultra entropy, it reflects uncertainty of the trust entropy; and Risk is trust risk, it reflects degree of trust risk.

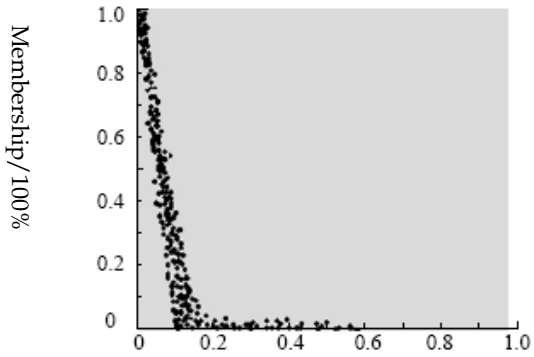
It must point out that when En is close to 0 and He=0, trust relationship among the nodes is fuzzy, but its ambiguity is definite. When En=0 and He=0, trust relationship among the nodes is definite and have no uncertainty. For example, the node is interior node of the system or definite trust relationship in the identical management system. In the chart 4-1, several different shapes of trust cloud have been given, and they have represented different trust value and uncertainty separately. Discovered from the chart that Ex is bigger, the trust cloud is closer to the biggest trust value, namely 1; En is bigger, the trust cloud's scope is wider; He is bigger, the trust cloud's cloud drop dispersion is bigger.

4.2.3 Differences between distrust and unknown trust

In the trust model, distrust and unknown trust has the difference. If node A does not trust node B, it represents A know B, and cannot trust it. However, if node A unknown trust node B, it represents A not know whether should trust B. The tradition method is using different trust value to distinguish distrust and unknown trust. For example: - 1 describes unknown trust and 0 describes distrust. However, this cannot reflect two concepts truthfully, especially unknown trust.

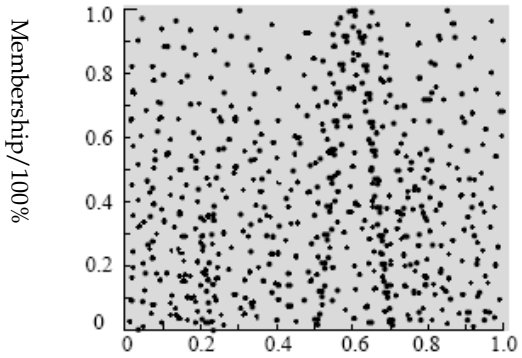
In the view of cloud model, distrust describe trust relationship among the nodes from the trust value angle, might use Ex=0 to describe. The unknown trust describe trust relationship among the nodes from trust uncertainty angle, may use En=1 and He=1 to describe.

However, these two kinds of trust have the possibility to coexist in the identical trust relationship. For example: If node A is strange to node B, therefore B unknown trusts A. Suppose B's trust threshold is small, A will be trusted under the certain extent. On the contrary, Suppose B's trust threshold is big, B will not trust A. In this case, the existing trust model could not describe. Based on the cloud model trust model distrust can be described by establishment expectation Ex =0 and unknown trust can be described by establishes ultra entropy En. From Fig. 3 (a) ~ (d), it can be seen that distrust and unknown trust have the differences of definiteness and the uncertainty as well as have the possibility to overlap.



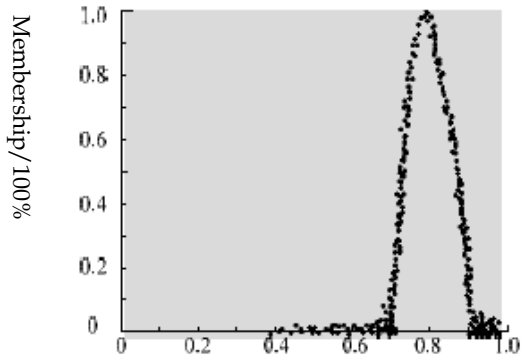
tc(0,0.05,0.001)

(a) Somewhat definite distrust



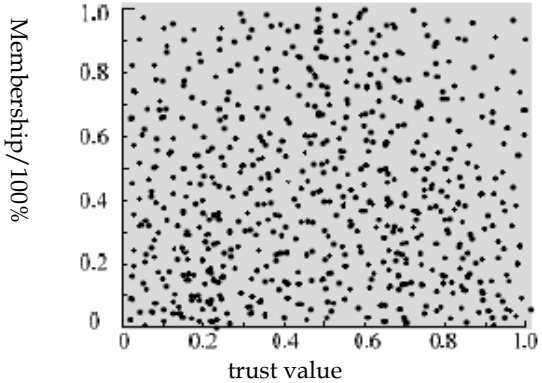
tc(0.6,0.1,0.2)

(b) Somewhat uncertainty unknown trust



tc(0.8,0.1,0.001)

(c) Somewhat indefinite distrust



tc(0.01,0.1,0.3)

(d) unknown trust, also distrust

Fig. 3. distrust and unknown trust cloud chart

4.3 Trust Propagation

In the wireless sensor network, the node cannot always directly obtain the recommendation trust value of the strange node from the neighbor node, therefore trust propagation is introduced. Supposed there are m nodes as E1, E2, E3, …Em, the node Ei, Ei+1($0 \le i \le m-1$) have the trust cloud $tc_i(Ex_i, En_i, He_i)$,and then computing the cloud trust $tc(Ex, En, He)$ is needed by this.

Because the trust cloud of E1 to Em is transmitted by the middle nodes, this is called trust cloud's propagation, and its computation algorithm is as follows:

$$\left.\begin{aligned} tc\left(\mathrm{Ex},\mathrm{En},\mathrm{He},\mathrm{Risk}\right) &= tc_1 \otimes tc_2 \otimes \cdots \otimes tc_m \\ &= \prod_{i=1}^{m} tc_i\left(\mathrm{Ex_i},\mathrm{En_i},\mathrm{He_i},Risk_i\right) \\ \mathrm{Ex} = \prod_{i=1}^{m} \mathrm{Ex_i},\ \mathrm{En} &= \min\left(\sqrt{\sum_{i=1}^{m} \mathrm{En_i^2}},1\right) \\ \mathrm{He} = \min\left(\sum_{i=1}^{m} \mathrm{He_i},1\right),\ Risk &= \min\left(\sum_{i=1}^{m} Risk_i,1\right) \end{aligned}\right\} \tag{4}$$

Here $\otimes$ is called as trust cloud logical multiplication operator. Analyze the parameter's significance, the trust cloud expectation more draws close to 0, the ultra entropy that the cloud drop dispersion increases, obviously after propagation, trust cloud's trust degree reduces with the uncertainty increases, this in accordance with the actual situation.

4.4 Trust mergence

In the wireless sensor network, the trust relationship during the numerous nodes constituted a trust network, there are many trust ways between two nodes. Thus, according to different trust ways, when calculating the trust relationships between two nodes it will obtain many trust clouds. By now, these clouds need to merge a trust cloud.

Supposed there are m nodes as $tc_1, tc_2, tc_3, \ldots, tc_m$, the nodes may merge into a trust cloud by the algorithm as follows:

$$\left.\begin{aligned} tc\left(\mathrm{Ex},\mathrm{En},\mathrm{He},\mathrm{Risk}\right) &= tc_1 \oplus tc_2 \oplus \cdots \oplus tc_m \\ &= \sum_{i=1}^{m} nc_i\left(\mathrm{Ex_i},\mathrm{En_i},\mathrm{He_i},Risk_i\right) \\ \mathrm{Ex} = \frac{1}{m}\prod_{i=1}^{m} \mathrm{Ex_i},\ \mathrm{En} &= \min\left(\frac{1}{m}\sum_{i=1}^{\mathrm{m}} \mathrm{En_i},1\right) \\ \mathrm{He} = \min\left(\frac{1}{m}\sum_{i=1}^{m} \mathrm{He_i},1\right),\ Risk &= \min\left(\frac{1}{m}\sum_{i=1}^{m} Risk_i,1\right) \end{aligned}\right\} \tag{5}$$

Here $\oplus$ is called as the trust cloud logical add operator. Analyze the parameter's significance, the cloud trust degree and the uncertainty of the merged cloud must surpass the first two kind of trust cloud.

5. Node selection algorithm for WSN

In this trust model, trust is not indicated with any definite value, but uses the trust cloud to express. The trust cloud is described with three digital eigenvalue, for it's very difficult to apply the trust cloud directly. Therefore, when selects node, using a definite trust value is quite important. In this model, a trust factor is defined. The trust factor can be calculated by using trust cloud and node can be chose with the trust factor.

5.1 calculates trust factor

Because this trust model describes trust with cloud, it not only described the trust degree moreover to describe trust indefiniteness, the definition algorithm of computation trust factor has also manifested these two characteristics. Therefore algorithm of trusted factor computation has defined as follows:

Supposed a trust cloud $tc(Ex, En, He)$ and N cloud drops, the trust factor can be calculated as the following steps:

- generate N cloud drop according to the forward cloud generator algorithm
- Calculates the trust factor with the formula

$$tg = \frac{1}{N}\sum_{i=1}^{N} x_i \times y_i \tag{6}$$

As the above algorithm shown, influenced by normal random number of the forward cloud generator algorithm, the calculated trust factor can not be the same by many times, this has also manifested the trust uncertainty. However, there will still be a trust expectation. If the trust cloud using En=0 and He=0 to describes a definite trust, the factor will present the same value every time when calculated it, namely Ex.

5.2 node selection algorithm

The processing flow of wireless sensor node selection algorithm as follows.

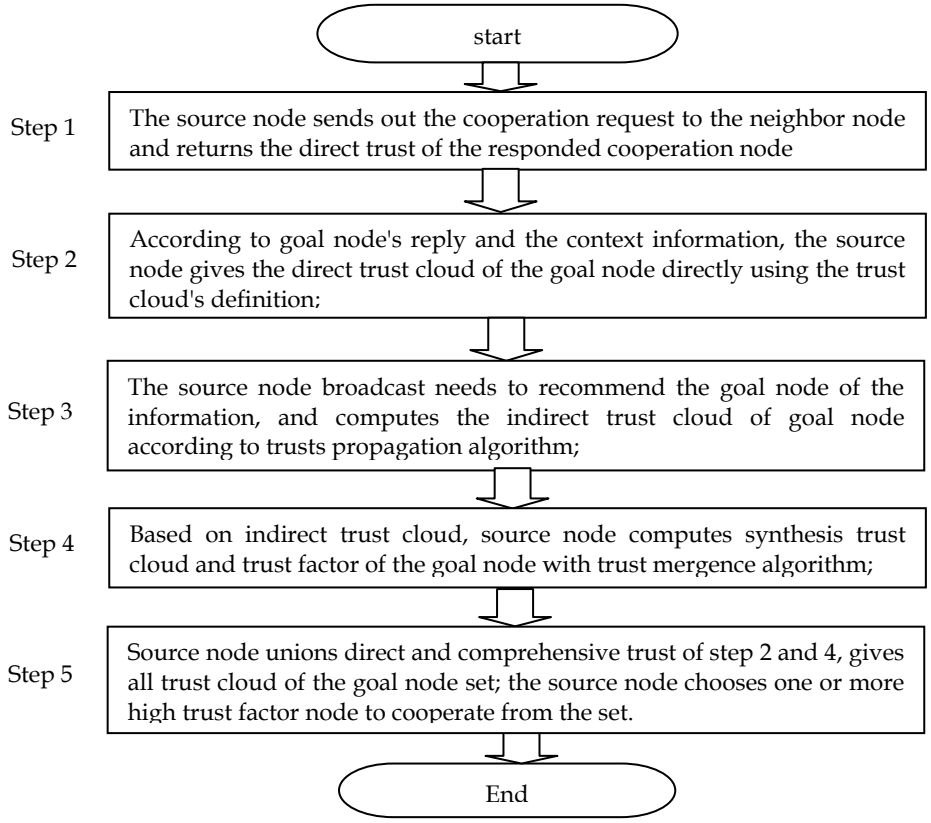


Fig. 4. wireless sensor node selection algorithm

6. Conclusion and Further Research

In this chapter, we have proposed a security architecture that provides confidentiality, integrity, and authentication with trust management for a wireless sensor network. For this purpose, we present a security architecture for wireless sensor networks that addresses most of the security requirements. It utilizes lightweight trust model algorithms that allow for easy acess control between the mobile sensor nodes and secure the communication inside the network. Furthermore, it minimizes the effects of compromised sensor nodes.Finally, we propose a cross-layer wireless sensor network trust model based on cloud model. This model unifies the description of trust degree and uncertainty of trust relationship among the nodes with trust cloud forms, and gives algorithms of trust cloud transmission and merge. By using the trust model and algorithm, a Node selection algorithm based on trust cloud is proposed.

7. Acknowledgements

This research work has been partially supported by the National Natural Sciemce Foundation of China under grant No.60872037, the Science & Technology Research Program of the Municipal Education Commission of Chongqing of China under Grant No. KJ090506, and the Natural Science Foundation of Chongqing of China under Grant No. CSTC 2010BB2218, CSTC 2008BB2411.

8. References

A.Cerpa, J.Elson, D.Estrin, L.Girod,M.Hamilton and J. Zhao.(2002)Habitat Monitoring: Application Driver for Wireless Communications Technology,*UCLA Computer Science Technical Report*.

Anderson, R., and Kuhn, M.(1996). Tamper Resistance – A Cautionary Note. *Proceedings of the 2nd Usenix Workshop on Electronic ommerce*, pp.1-11, USENIX Association, Oakland.

Basagni, S., Herrin, C., Bruschi, D., and Rosti, E.(2001). Secure Pebblenets. *Proceedings of the 2nd International Symposium on Mobile Ad Hoc Networking & Computing*.pp. 156 – 163, ACM Press, Washington DC.

Bin Ma.(2009).A Novel Stereoscopic Security Architecture with Trust Management for Wireless Sensor Networks.*Proceedings of the ICCSN '09*. pp.797-800, IEEE Computer Society Press,Maoco.

Bin Ma.(2009).Cross-Layer Trust Model and Algorithm of Node Selection in Wireless Sensor Networks.*Proceedings of the ICCSN '09*. pp.812-815, IEEE Computer Society Press,Maoco.

Blaze M, Feigenbaum J, Lacy J.(1996). Decentralized trust management. *Proc. of the 17th Symp. on Security and Privacy*. pp.164−173.Oakland: IEEE Computer Society Press.

Carman, D.W., Kruus, P.S., and Matt, B.J.(2000). *Constraints an Approaches for Distributed Sensor Network Security*. Technical Report, NAI Labs.

Chan, H., Perrig, A., and Song, D.(2003).Predistribution Schemes for Sensor Networks. *Proceedings of the IEEE Security and Privacy Symposium*.pp. 197 – 213. IEEE Computer Society Press, Los Alamos.

Deyi Li,Changyu Liu,Yi Du,Xu Han.(2004). Artificial Intelligence with Uncertainty. *Journal of Software*, vol. 15(11),pp 1583-1594.

Deyi Li.(2000). Uncertainty in Knowledge Representation. *Engineering Science*, vol. 2(10),pp 73-79.

Eschenauer, L., and Gligor, V.D.(2002).A Key-Management Scheme for Distributed Sensor Networks. *Proceedings of the Conference on Computer and Communications Security '02*. pp. 41 – 47.Washington DC .

I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci.(2002). Wireless Sensor Networks: A Survey, *Computer Networks*,Vol. 38, No. 8, August 2002, pp. 398-422.

J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister. (2000). System architecture directions for networked sensors. *Proceedings of ACM ASPLOS-IX*,pp.21-28, Cambridge,MA,USA.

J. Kahn, R. Katz, and K. Pister.(1999).Next Century Challenges: Mobile Networking for Smart Dust, *Proc. of ACM MobiCom'99*,pp. 271-278.ACM Press, Washington DC.

Ma Bin.(2008). Coordinated Trust Model in Pervasive Computing Based on Cloud Theory. *Computer Engineering*, vol. 34(9),pp 162-163,166.

Ma Bin,Xie Xian-zhong.(2009). A novel intelligent risk based access control planning. *Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition)*, vol. 21(4),pp 523-527.

Ma Bin,Xie Xian-zhong.(2010). Cloud Trust Model for Wireless Sensor Networks. *Computer Science*, vol. 37(3),pp 128-132.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J.D.(2001). SPINS: Security Protocols for Sensor Networks. *Proceedings of the 7th International Conference on mobile Computing and Networks*.,pp.189 –199,ACM Press, Washington DC.

S. Marsh.(1994) Formalising Trust as a Computational Concept, *Departmet of Computer Science and Mathematics of PhD: University of Stirling*.

Wood, A.D., and Stankovic, J.A.(2002). Denial of Service in Sensor Networks.*IEEE Computer*, Vol. 35, No. 10, October 2002, pp. 54 – 62.

Wood, A.D., Stankovic, J.A, and Son, S.H.(2003). JAM: A Jammed-Area Mapping Service for Sensor Networks. *Proceedings of the 24th Real-Time Systems Symposium*.pp. 286 – 297. IEEE Computer Society Press, Los Alamos .

Zhu, S., Setia, S., and Jajodia, S.(2003).LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks.*Proceedings of the Conference on Computer and Communications Security '03*.pp. 62 – 72, ACM Press, Washington DC, 2003.

20

Distributed Detection of Node Capture Attacks in Wireless Sensor Networks

Jun-Won Ho
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, USA

Abstract

Wireless sensor networks are vulnerable to node capture attacks because sensor nodes are usually deployed in unattended manner. Once attacker captures sensor nodes, he can compromise them and launch various types of attacks with those compromised nodes. Therefore, node capture attacks are hazardous and should be detected as soon as possible to reduce the harm incurred by them. To meet this need, we propose a node capture detection scheme in wireless sensor networks. Our scheme detects the captured sensor nodes by using the sequential analysis. We analytically show that our scheme detects node capture attacks in robust and efficient manner.

1. Introduction

Wireless sensor networks have recently gained much attention in the sense that they can be readily deployed for many different types of missions. In particular, they are useful for the missions that are difficult for humans to carry out. For example, they are suitable for sensing dangerous natural phenomenon such as volcano eruption, biohazard monitoring, and forest fire detection. In addition to these hazardous applications, sensor networks can also be deployed for battle field surveillance, border monitoring, nuclear and chemical attack detection, intrusion detection, flood detection, weather forecasting, traffic surveillance and patient monitoring (Akyildiz et al., 2002).

To carry out a variety of missions, the network operator deploys the base station and a set of small sensor devices in the network field. Specifically, sensor devices form ad-hoc networks, collaborate with each other to sense the phenomenon associated with the assigned missions and then send the sensory data to the base station. The network operator obtains the mission related information by analyzing the data collected at the base station. To help sensor nodes carry out the missions efficiently and effectively, many researchers proposed a variety of the network service and communication protocols (Yick et al., 2008). Specifically, localization, coverage, compression and aggregation protocols have been proposed for the network services. Various network protocols from physical layer to transport layer have been proposed for the communication.

Since sensor networks are often deployed in an unattended manner, most of these protocols are exposed to a variety of attacks such as denial of service attacks, routing disruption and

false data injection attacks, network service disruption attacks (Du & Xiao, 2008; Karlof & Wagner, 2003; Wood & Stankovic, 2002). To defend the wireless sensor networks against these various attacks, many schemes have been developed in the literature. For instance, secure routing schemes have been proposed to mitigate routing disruption attacks (Karlof & Wagner, 2003; Parno et al., 2006). False data injection attacks can be mitigated by using the authentication schemes (Ye et al., 2004; Yu & Li, 2009; Zhu et al., 2004). Secure data aggregation protocols are used to prevent attacker from disrupting aggregation (Chan et al., 2006; Deng et al., 2003; Przydatek et al., 2003; Yang et al., 2006). Many schemes have also been proposed to protect localization and time synchronization protocols from the threat (Capkun & Hubaux, 2006; Ganeriwal et al., 2005; Hu et al., 2008; Li et al., 2005; Liu et al., 2005; Song et al., 2007; KSun et al., 2006).

However, most of them focus on making the protocols be attack-resilient rather than removing the source of attacks. Although attack-resiliency approach mitigates the threats on the network services and communication protocols, this approach requires substantial time and effort to continuously enhance the robustness of the protocols in accordance with the emergence of new types of attacks. Moreover, since it is hard to predict new types of attacks, the protocols will likely have resiliency only after being damaged by new types of attacks. Thus, we need to detect and revoke the sources of attacks as soon as possible to substantially reduce the costs and damages incurred by employing attack-resilience approach. The principle sources of various attacks are compromised sensor nodes in the sense that attacker can compromise sensor nodes by exploiting the unattended nature of wireless sensor networks and thus do any malicious activities with them.

A straightforward strategy for sensor node compromise is to launch *node capture attack* in which adversary physically captures sensor nodes, removes them from the network, compromises and redeploys them in the network. After redeploying compromised nodes, he can mount a variety of attacks with compromised nodes. For example, he can simply monitor a significant fraction of the network traffic that would pass through these compromised nodes. Alternatively, he could jam legitimate signals from benign nodes or inject falsified data to corrupt monitoring operation of the sensors. A more aggressive attacker could undermine common sensor network protocols, including cluster formation, routing, and data aggregation, thereby causing continual disruption to the network operations. Hence, node capture attacks are dangerous and thus should be detected as quickly as possible to minimize the damage incurred by them.

To meet this need, we propose a node capture attack detection scheme in wireless sensor networks. We use the fact that the physically captured nodes are not present in the network during the period from the captured time to redeployed time. Accordingly, captured nodes would not participate in any network operations during that period. By leveraging this intuition, we detect captured nodes by using the Sequential Probability Ratio Test (SPRT) Wald (2004). The main advantage of our scheme is to quickly detect captured nodes with the aid of the SPRT.

The rest of paper is organized as follows. Section 2 describes the network and attacker models. Section 3 describes our node capture attack detection scheme. Section 4 presents the security analysis of our proposed scheme. Section 5 presents the performance analysis of our proposed scheme. Section 6 presents the related work. Finally, Section 7 concludes the paper.

2. Models

In this section, we present the network models and attacker models for our proposed scheme.

2.1 Network Models

We first assume a static sensor network in which the locations of sensor nodes do not change after deployment. We also assume that every sensor node works in promiscuous mode and is able to identify the sources of all messages originating from its neighbors. We believe that this assumption does not incur substantial overhead because each node inspects only the source IDs of the messages from its neighbors rather than the entire contents of the messages.

2.2 Attacker Models

We assume that an attacker can physically capture sensor nodes to compromise them. However, we place limits on the number of sensor nodes that he can physically capture in each target region. This is reasonable from the perspective that an increase in the number of the captured sensor nodes will lead to a rise in the likelihood that attacker is detected by intruder detection mechanisms. Therefore, a rationale attacker will want to physically capture the limited number of sensor nodes in each target region while not being detected by intruder detection mechanisms. Moreover, we assume that it takes a certain amount of time from capturing nodes to redeploying them in the network. This is reasonable in the sense that an attacker needs some time to compromise captured sensor nodes.

3. Node Capture Attack Detection Using the Sequential Probability Ratio Test

In this section, we present the details of node capture detection scheme.

A straightforward approach for node capture detection is to leverage the intuition that a captured node is not present in the network from being captured to being redeployed. Specifically, we first measure the absence time period of a sensor node and then compare it to a pre-defined threshold. If it is more than threshold value, we decide the sensor node as a captured nodes. This simple approach achieves efficient node capture detection capability as long as a threshold value is properly configured. However, it is not easy to configure a proper a threshold value to detect captured nodes. If we set threshold to a high value, it is likely that captured nodes bypass the detection. On the contrary, if we set threshold to a low value, it is likely that benign nodes can be detected as captured nodes. To minimize these false positives and negatives, we need to set up threshold in such a way that it is dynamically changed in accordance with the measured absence time duration for a node. To meet this need, we use the Sequential Probability Ratio Test (SPRT) (Wald, 2004), which is a statistical decision process and is regarded as a dynamic threshold scheme (Jung et al., 2004). We can take advantage of using the SPRT from the perspective that the SPRT reaches a decision with few pieces of samples while achieving low false positive and false negative rates (Wald, 2004). Specifically, we apply the SPRT to node capture detection problem as follows. For each time slot, every sensor node measures the number of messages sent by its neighbors. Each time the number of messages sent by a neighbor is above (resp. equal to) zero, it will expedite the test process to accept the null (resp. alternate) hypothesis that the neighbor is present (resp. absent) in the network. Once a node accepts alternate hypothesis, it decides that the neighbor has been captured and disconnects the communication with the neighbor.

After deployment, every sensor node u discovers its neighboring nodes. The entire time domain of node u is divided into a series of time slots. For each neighbor node v, node u measures the number of messages sent by v every time slot. We denote the number of messages whose originator is v during the ith time slot by N_i. Let V_i be denote a Bernoulli random

variable that is defined as:

$$V_i = \begin{cases} 1 & \text{if } N_i = 0 \\ 0 & \text{if } N_i > 0 \end{cases} \quad (1)$$

where $i \geq 1$. The success probability δ of Bernoulli distribution is defined as

$$Pr(V_i = 1) = 1 - Pr(V_i = 0) = \delta. \quad (2)$$

If δ is smaller than or equal to a preset threshold δ', it is likely that node v is present in the network and is accordingly not captured by attacker. On the contrary, if $\delta > \delta'$, it is likely that node v is absent in the network and is accordingly captured by attacker. The problem of deciding whether v is captured or not can be formulated as a hypothesis testing problem with null and alternate hypotheses of $\delta \leq \delta'$ and $\delta > \delta'$, respectively. In this problem, we need to devise an appropriate sampling strategy in order to prevent hypothesis testing from leading to a wrong decision. In particular, we should specify the maximum possibilities of wrong decisions that we want to tolerate for a good sampling strategy. To do this, we reformulate the above hypothesis testing problem as one with null and alternate hypotheses of $\delta \leq \delta_0$ and $\delta \geq \delta_1$, respectively, such that $\delta_0 < \delta_1$. In this reformulated problem, the acceptance of the alternate hypothesis is regarded as a false positive error when $\delta \leq \delta_0$, and the acceptance of the null hypothesis is regarded as false negative error when $\delta \geq \delta_1$. To prevent the decision process from making these two types of errors, we define a user-configured false positive α' and false negative β' in such a way that the false positive and negative should not exceed α' and β', respectively.

Now we present how node u performs the SPRT to make a decision of v with the n observed samples, where N_i is treated as a sample. Let us define H_0 as the null hypothesis that v is present in the network and is not captured by attacker, H_1 as the alternate hypothesis that v is not present in the network and is captured by attacker. We then define L_n as the log-probability ratio on n samples, given as:

$$L_n = \ln \frac{\Pr(V_1, \ldots, V_n | H_1)}{\Pr(V_1, \ldots, V_n | H_0)}$$

Assume that V_i is independent and identically distributed. Then L_n can be rewritten as:

$$L_n = \ln \frac{\prod_{i=1}^{n} \Pr(V_i | H_1)}{\prod_{i=1}^{n} \Pr(V_i | H_0)} = \sum_{i=1}^{n} \ln \frac{\Pr(V_i | H_1)}{\Pr(V_i | H_0)} \quad (3)$$

Let y_n denote the number of times that $V_i = 1$ in the n samples. Then we have $L_n = y_n \ln \frac{\delta_1}{\delta_0} + (n - y_n) \ln \frac{1-\delta_1}{1-\delta_0}$ where $\delta_0 = \Pr(V_i = 1 | H_0)$, $\delta_1 = \Pr(V_i = 1 | H_1)$. The rationale behind the configuration of δ_0 and δ_1 is as follows. δ_0 should be configured in accordance with the likelihood of the occurrence that a benign node is determined to be absent in the network during a time slot. δ_1 should be configured to consider the likelihood of the occurrence that a captured node is determined to be absent in the network during a time slot. On the basis of the log-probability ratio L_n, the SPRT for H_0 against H_1 is given as follows:

- $L_n \leq \ln \frac{\beta'}{1-\alpha'}$: accept H_0 and terminate the test.
- $L_n \geq \ln \frac{1-\beta'}{\alpha'}$: accept H_1 and terminate the test.
- $\ln \frac{\beta'}{1-\alpha'} < L_n < \ln \frac{1-\beta'}{\alpha'}$: continue the test process with another observation.

This SPRT can be written as:

- $y_n \leq s_0(n)$: accept H_0 and terminate the test.
- $y_n \geq s_1(n)$: accept H_1 and terminate the test
- $s_0(n) < y_n < s_1(n)$: continue the test process with another observation.

Where

$$s_0(n) = \frac{\ln \frac{\beta'}{1-\alpha'} + n \ln \frac{1-\delta_0}{1-\delta_1}}{\ln \frac{\delta_1}{\delta_0} - \ln \frac{1-\delta_1}{1-\delta_0}}, \quad s_1(n) = \frac{\ln \frac{1-\beta'}{\alpha'} + n \ln \frac{1-\delta_0}{1-\delta_1}}{\ln \frac{\delta_1}{\delta_0} - \ln \frac{1-\delta_1}{1-\delta_0}}$$

,α' and β' are the user-configured false positive and false negative rates, respectively.
If the SPRT terminates in acceptance of H_0, node u restarts the SPRT with newly received messages from v. However, if the SPRT accepts H_1, u terminates the SPRT on v, decides v as a captured node, and disconnects the communication with v.
The pseudocode for the SPRT is presented as Algorithm 1.

Algorithm 1 SPRT for replica detection

```
INITIALIZATION: t = 1, y = 0
INPUT: N_t
OUTPUT: accept the hypothesis H_0 or H_1
compute s_0(t) and s_1(t)
if N_t == 0 then
   y = y + 1
end if
if y >= s_1(t) then
   accept the alternate hypothesis H_1 and terminate the test
end if
if y <= s_0(t) then
   accept the null hypothesis H_0 and initialize t to 1 and y to 0
   return;
end if
t = t + 1
```

4. Security Analysis

In this section, we first present the detection capability of our scheme and then discuss about the limitations of node capture attacks under the presence of our scheme and countermeasures against some possible attack strategies against our scheme.
In the SPRT, the following types of errors are defined.

- α : error probability that the SPRT leads to accepting H_1 when H_0 is true.
- β : error probability that the SPRT leads to accepting H_0 when H_1 is true.

Since H_0 is the hypothesis that a node u has not been captured, α and β are the false positive and false negative probabilities of the SPRT, respectively. According to Wald's theory (Wald, 2004), the upper bounds of α and β are:

$$\alpha \leq \frac{\alpha'}{1-\beta'}, \qquad \beta \leq \frac{\beta'}{1-\alpha'} \tag{4}$$

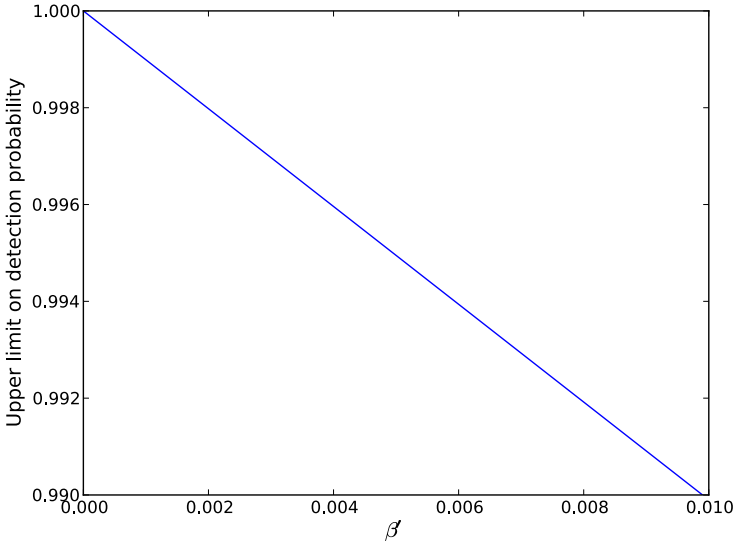


Fig. 1. Upper limit on detection probability vs. β' when $\alpha' = 0.01$.

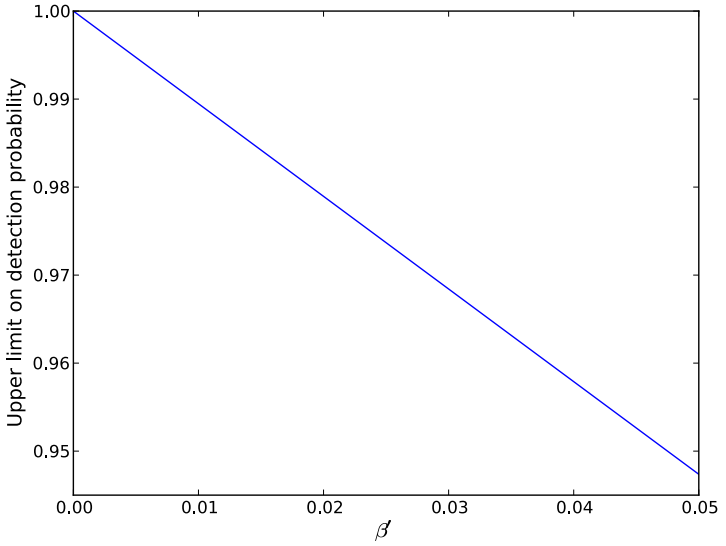


Fig. 2. Upper limit on detection probability vs. β' when $\alpha' = 0.05$.

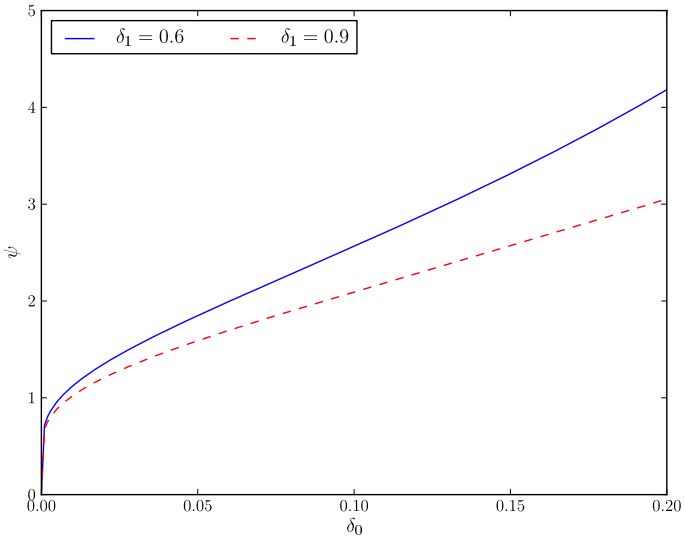


Fig. 3. ψ vs. δ_0 when $\alpha' = \beta' = 0.01$.

Furthermore, Wald proved that the sum of the false positive and negative probabilities of the SPRT are limited by the sum of user-configured false positive and negative probabilities. Namely, the following inequality holds:

$$\alpha + \beta \leq \alpha' + \beta' \tag{5}$$

Since β is the false negative probability, $(1 - \beta)$ is the node capture detection probability. Accordingly, the lower bound on the node catpure detection probability will be:

$$(1 - \beta) \geq \frac{1 - \alpha' - \beta'}{1 - \alpha'} \tag{6}$$

From Equations 4 and 6, we can see that low user-configured false positive and negative probabilities will lead to a low false negative probability for the sequential test process. Hence, it will result in high detection rates.

As shown in Figures 1 and 2, we study how α' and β' affect the upper limit of node capture detection probability $(1 - \beta)$. Specifically, the upper limit decreases as the rise in β' when the user configures α' to 0.01 and 0.05. However, we see that the upper limit is bounded from below 0.99 (resp., 0.945) when $\alpha' = 0.01$ (resp., 0.05) as long as β' is configured to at most 0.01 (resp., 0.05). Hence, the node capture detection capability is guaranteed with at least probability of 0.945 when both α' and β' are set to at most 0.05.

Now we derive the limitation of the time period from when a node is captured and removed in location L to when it is redeployed in the same location L. Suppose that the entire n time slots are taken from the removal to redeployment of captured node. Since the captured node

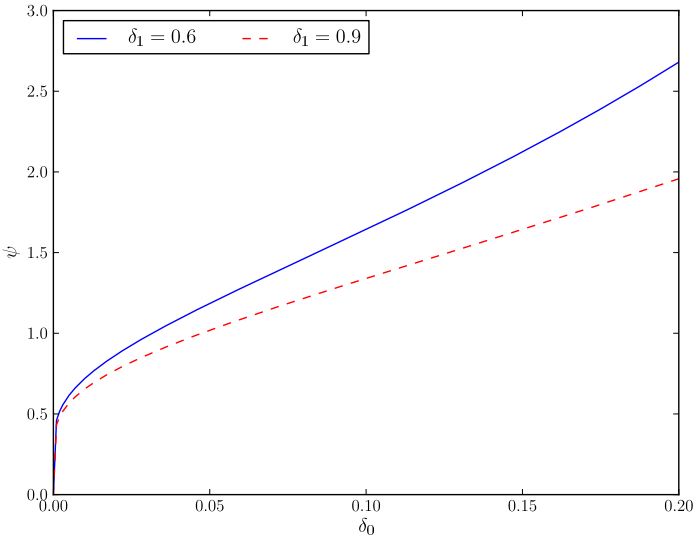


Fig. 4. ψ vs. δ_0 when $\alpha' = \beta' = 0.05$.

will not be present in the network for n time slots and a time slot corresponds to a sample in the SPRT, $y_n = n$ holds. Accordingly, $y_n = n < s_1(n)$ should hold for captured node to avoid being detected. In other words, the following Inequality should hold to bypass the detection:

$$n < \psi = \frac{\ln \frac{1-\beta'}{\alpha'}}{\ln \frac{\delta_1}{\delta_0}} \tag{7}$$

As shown in Figures 3 and 4, we study how the values of δ_0 and δ_1 affect ψ when $\alpha' = 0.01, \beta' = 0.01$ and $\alpha' = 0.05, \beta' = 0.05$. Specifically, ψ increases as δ_0 rises when δ_1 is configured to 0.6 and 0.9, but it decreases as δ_1 rises when δ_0 is fixed. We see from this that small and large values of δ_0 and δ_1 lead to the small value of ψ. We also observe that n is less than 5 and 3 in the case of $\alpha' = \beta' = 0.01$ and $\alpha' = \beta' = 0.05$, respectively. This means that attacker should finish compromising and redeploying the captured node within at most five time slots in order to prevent them from being detected. Hence, our scheme will substantially limit the time duration for captured node not to be detected.

However, if a captured node is not redeployed in its initial location L but in different location L', even though it cannot be accepted as legitimate neighbors by the nodes around L, it can still be accepted as legitimate neighbors by the nodes around L' and thus have an impact on these nodes. To defend the network against this attack, we propose a countermeasure based on the group deployment strategy. This involves three important assumptions.

First, we assume that sensor nodes are deployed in group-by-group. More specifically, sensor nodes are grouped together by the network operator and programmed with the corresponding group information before deployment, with each group of nodes being deployed towards the same location, called the *group deployment point*. After deployment, the group members exhibit similar geographic relations. We argue that this is reasonable for sensor network in

which nodes are spread over a field, such as being dropped from an airplane or spread out by hand. A simple way to do this would be to keep the groups of nodes in bags marked with the group IDs and use a marked map with the group IDs on it. All that is needed is a map of the territory and a way to pre-determine the deployment points, such as assigning a point on a grid to each group. This argument is further supported by the fact that the group deployment strategy has been used for various applications in sensor networks such as key distribution (Du et al., 2004), detection of anomalies in localization (Du et al., 2005), and public key authentication (Du et al., 2005).

The deployment follows a particular probability density function (pdf), say f, which describes the likelihood of a node being a certain distance from its group deployment point. For simplicity, we use a two-dimensional Gaussian distribution to model f, as in (Du et al., 2005). Let (x_g, y_g) be the group deployment point for a group g. A sensor node in group g is placed in a location (x, y) in accordance with the following model:

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_g)^2+(y-y_g)^2}{2\sigma^2}} \tag{8}$$

where (x, y) is group deployment point and σ is the standard deviation of the two-dimensional Gaussian distribution. According to Equation 8, 68% and 99% of nodes in a group are placed within a circle whose center is the group deployment point and radius is σ and 3σ, respectively.

Second, we assume that it takes some time for an attacker to capture and compromise a sensor node. This need not be a long time, but we assume that there is a minimum amount of time that it takes to compromise a node once it has been deployed. [1] Third, we assume that the clocks of all nodes are loosely synchronized with a maximum error of ϵ. This can be achieved by the use of secure time synchronization protocols as proposed in (Ganeriwal et al., 2005; Hu et al., 2008; Song et al., 2007; KSun et al., 2006).

Under these assumptions, the main idea of the proposed countermeasure is to pre-announce the deployment time of each group, and have nodes treat as captured and redeployed any node that initiates communications after a long time of its expected deployment. More specifically, when a group G_u of nodes are deployed, they will be pre-loaded with a time stamp T_u that is digitally signed by a trusted server. This time stamp indicates that the sensor nodes in G_u should finish neighbor discovery before time T_u. If they try to setup neighbor connections with other nodes after time T_u, they are considered to be captured and redeployed nodes. The time stamp T_u should be a function of the deployment time T, the time T_r needed for capturing, compromising, and redeploying a node, and the maximum time synchronization error ϵ. Specifically, the network operator should set $T + T_d + \epsilon < T_u < T + T_d + T_r - \epsilon$, where T_d is the neighbor discovery time, such that no nodes should have clocks too fast to accept the new node, but no new node could be compromised and accepted in time. This means that $\epsilon < 0.5T_c$ determines the maximum amount of allowable error.

5. Performance Analysis

This section describes how many observations are required on average for each node to decide whether its neighboring node has been captured or not.

Let n denote the number of samples to terminate the SPRT. Since n is changed with the types of samples, it is treated as a random variable with an expected value $E[n]$. According to (Wald,

[1] According to (Hartung et al., 2005), it took approximately one minute to compromise a node.

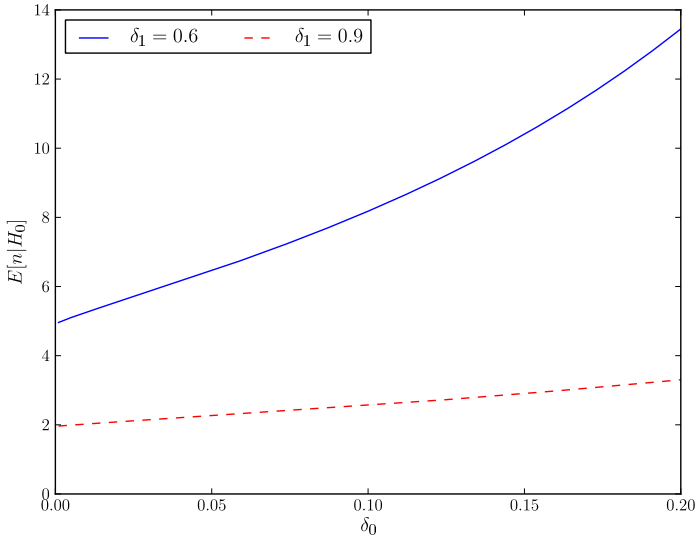


Fig. 5. $E[n|H_0]$ vs. δ_0 when $\alpha' = \beta' = 0.01$.

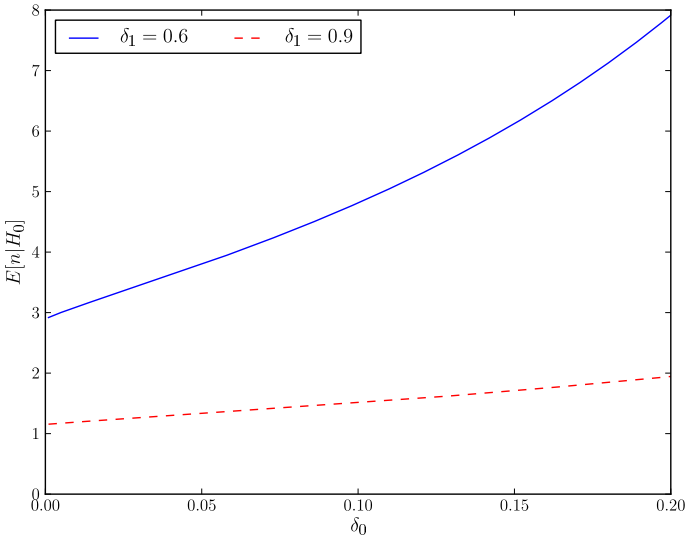


Fig. 6. $E[n|H_0]$ vs. δ_0 when $\alpha' = \beta' = 0.05$.

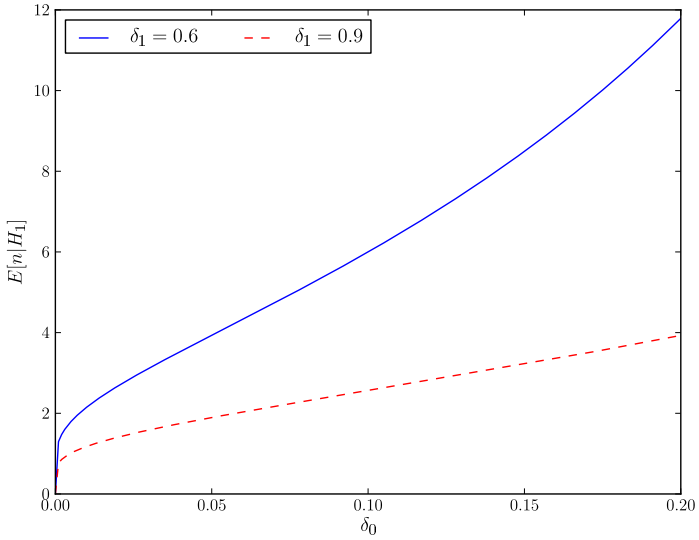


Fig. 7. $E[n|H_1]$ vs. δ_0 when $\alpha' = \beta' = 0.01$.

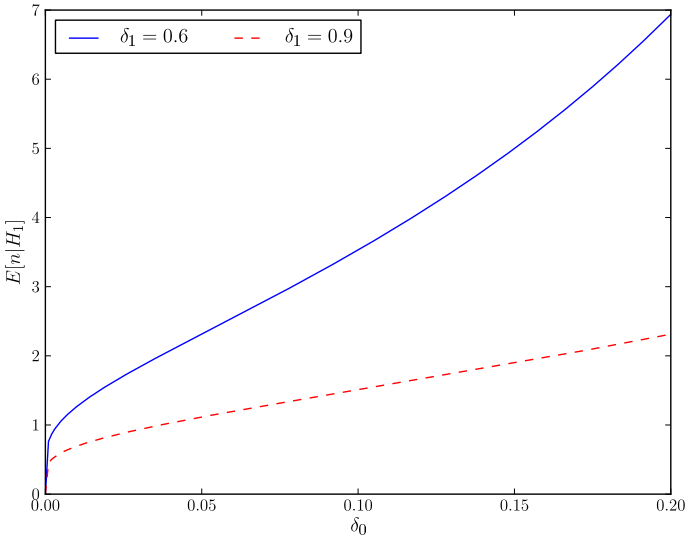


Fig. 8. $E[n|H_1]$ vs. δ_0 when $\alpha' = \beta' = 0.05$.

2004), $E[n]$ is given by:

$$E[n] = \frac{E[L_n]}{E\left[\ln \frac{\Pr(V_i|H_1)}{\Pr(V_i|H_0)}\right]} \tag{9}$$

From Equation 9, we compute the expected values of n conditioned on hypotheses H_0 and H_1 as follows:

$$E[n|H_0] = \frac{(1-\alpha')\ln\frac{\beta'}{1-\alpha'} + \alpha'\ln\frac{1-\beta'}{\alpha'}}{\delta_0 \ln\frac{\delta_1}{\delta_0} + (1-\delta_0)\ln\frac{1-\delta_1}{1-\delta_0}}$$

$$E[n|H_1] = \frac{\beta'\ln\frac{\beta'}{1-\alpha'} + (1-\beta')\ln\frac{1-\beta'}{\alpha'}}{\delta_1 \ln\frac{\delta_1}{\delta_0} + (1-\delta_1)\ln\frac{1-\delta_1}{1-\delta_0}} \tag{10}$$

As shown in Figures 5, 6, 7, and 8, we study how the values of δ_0 and δ_1 affect $E[n|H_0]$ and $E[n|H_1]$ when $\alpha' = \beta' = 0.01$ and $\alpha' = \beta' = 0.05$. Specifically, $E[n|H_1]$ increases as the rise of δ_0 for a given value of δ_1. This means that captured nodes are detected with a small number of samples when δ_0 is small. For a given value of δ_0, $E[n|H_1]$ decreases as the increase of δ_1. This means that large values of δ_1 reduce the number of samples required for node capture detection. Similarly, the small value of δ_0 and the large value of δ_1 contribute to decrease of $E[n|H_0]$, leading to the small number of samples required for deciding that benign node is not captured.

6. Related Work

In this section, we describe a number of research works that are related to node capture detection in wireless sensor networks.

In (Tague & Poovendran, 2008), node capture attacks are modeled in wireless sensor networks. However, this work did not propose detection schemes against node capture attacks. In (Conti et al., 2008), node capture attack detection scheme was proposed in mobile sensor networks. They leverage the intuition that a mobile node is regarded as being captured if it is not contacted by other mobile nodes during a certain period of time. However, this scheme will not work in static sensor networks where sensor nodes do not move after deployment.

Software-attestation based schemes have been proposed to detect the subverted software modules of sensor nodes (Park & Shin, 2005; Seshadri et al., 2004; Shaneck et al., 2005; Yang et al., 2007). Specifically, the base station checks whether the flash image codes have been maliciously altered by performing attestation randomly chosen portions of image codes or the entire codes in (Park & Shin, 2005; Seshadri et al., 2004; Shaneck et al., 2005). In (Yang et al., 2007), a sensor node's image codes are attested by its neighbors. However, all these schemes require each sensor to be periodically attested and thus incur a large overhead in terms of communication and computation.

Reputation-based trust management schemes have been proposed to manage individual node's trust in accordance with its actions (Ganeriwal & Srivastava, 2004; Li at al., 2007; YSun et al., 2006). Specifically, a reputation-based trust management scheme was proposed in (Ganeriwal & Srivastava, 2004). The main idea of the scheme is to use a Bayesian formulation in order to compute an individual node's trust. In (YSun et al., 2006) information theoretic frameworks for trust evaluation were proposed. Specifically, entropy-based and probability-based schemes have been proposed to compute an individual node's trust. In (Li at al., 2007), node mobility is leveraged to reduce an uncertainty in trust computation and speed up the trust convergence. However, these trust management schemes do not revoke compromised

nodes and thus compromised nodes can keep performing malicious activities in the network. ID traceback schemes have been proposed to locate the malicious source of false data (Ye et al., 2007; Zhang et al., 2006). However, they only trace a source of the data sent to the base station and thus they do not locate the malicious sources that send false data or control messages to other benign nodes in the network.

After physically capturing and compromising a few sensor nodes, attacker can generate many replica nodes with the same ID and secret keying materials as the compromised nodes, and mount a variety of attacks with replica nodes. Randomized and line-selected multicast schemes were proposed to detect replicas in wireless sensor networks (Parno et al., 2005). In the randomized multicast scheme, every node is required to multicast a signed location claim to randomly chosen witness nodes. A witness node that receives two conflicting location claims for a node concludes that the node has been replicated and initiates a process to revoke the node. The line-selected multicast scheme reduces the communication overhead of the randomized multicast scheme by having every claim-relaying node participate in the replica detection and revocation process.

A Randomized, Efficient, and Distributed (RED) protocol was proposed to enhance the line-selected multicast scheme of (Parno et al., 2005) in terms of replica detection probability, storage and computation overheads (Conti et al., 2007). However, RED still has the same communication overhead as the line-selected multicast scheme of (Parno et al., 2005). More significantly, their protocol requires repeated location claims over time, meaning that the cost of the scheme needs to be multiplied by the number of runs during the total deployment time. Localized multicast schemes based on the grid cell topology detect replicas by letting location claim be multicasted to a single cell or multiple cells (Zhu et al., 2007). The main strength of (Zhu et al., 2007) is that it achieves higher detection rates than the best scheme of (Parno et al., 2005). However, (Zhu et al., 2007) has similar communication overheads as (Parno et al., 2005).

A clone detection scheme was proposed in sensor networks (Choi et al., 2007). In this scheme, the network is considered to be a set of non-overlapping subregions. An exclusive subset is formed in each subregion. If the intersection of subsets is not empty, it implies that replicas are included in those subsets. Fingerprint-based replica node detection scheme was proposed in sensor networks (Xing et al., 2008). In this scheme, nodes report fingerprints, which identify a set of their neighbors, to the base station. The base station performs replica detection by using the property that fingerprints of replicas conflict each other.

7. Conclusion

In this paper, we proposed a node capture attack detection scheme using the Sequential Probability Ratio Test (SPRT). We showed the limitations of the benefits that attacker can take from launching node capture attacks when our scheme is employed. We also analytically showed that our scheme detects node capture attacks with a few number of samples while sustaining the false positive and false negative rates below 1%.

8. References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks : a survey. *Computer Networks* 38(4):393–422, March 2002.

Boneh, D. & Franklin, M.K. (2001). Identity-based encryption from the weil pairing. In *CRYPTO*, pages:213-229, August 2001.

Capkun, S. & Hubaux, J.P. (2006). Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, February 2006.

Chan, H., Perrig, A., & Song, D. (2003). Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages:197-213 , May 2003.

Chan, H., Perrig, A., & Song, D. (2006). Secure hierarchical in-network aggregation in sensor networks . In *ACM CCS*, pages:278-287, October 2006.

Cocks, C. (2001). An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages:360-363, December 2001.

Choi, H., Zhu, S., & La Porta, T.F. (2007). {SET}: detecting node clones in sensor networks. In *IEEE/CreateNet SecureComm*, pages:341-350, September 2007.

Conti, M., Pietro, R.D., Mancini, L.V., & Mei, A. (2007). A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In *ACM Mobihoc*, pages:80-89, September 2007.

Conti, M., Pietro, R., Mancini, L., & Mei, A. (2008). Emergent Properties: Detection of the Node-capture Attack in Mobile Wireless Sensor Networks. In *ACM WiSec*, April 2008.

Delgosha, F. & Fekri, F. (2006). Threshold key-establishment in distributed sensor networks using a multivariate scheme. In *IEEE INFOCOM*, pages:1-12, April 2006.

Deng, J., Han, R., & Mishra, S. (2003). Security support for in-network processing in wireless sensor networks. In *ACM SASN*, pages:83-93, October 2003.

Du, W., Deng, J., Han, Y. S., & Varshney, P. (2003). A pairwise key pre-distribution scheme for wireless sensor networks. In *ACM CCS*, pages 42–51, October 2003.

Du, W., Deng, J., Han, Y. S., Chen, S., & Varshney, P. (2004). A key management scheme for wireless sensor networks using deployment knowledge. In *IEEE INFOCOM*, pages:586-597, March 2004.

Du, W., Fang, L., & Ning, P. (2005). {LAD}: localization anomaly detection for wireless sensor networks. In *IEEE IPDPS*, pages:874-886, April 2005.

Du, W., Wang, R., & Ning, P. (2005). An efficient scheme for authenticating public keys in sensor networks. In *ACM MobiHoc*, pages:58-67, May 2005.

Du, X. & Xiao, Y. (2008). Chapter 17: A survey on sensor network security *Springer Wireless Sensor Networks and Applications*, 2008

Eschenauer, L. & Gligor, V. (2002). A key-management scheme for distributed sensor networks. In *ACM CCS*, pages:41-47, November 2002.

Ganeriwal, S.& Srivastava, M. (2004). Reputation-based framework for high integrity sensor networks. In *ACM SASN*, pages:66-77, October 2004.

Ganeriwal, S., Čapkun, S., Han, C.C., & Srivastava, M.B. (2005). Secure time synchronization service for sensor networks. In *ACM WiSe*, pages:97-106, September 2005.

Gupta, V., Millard, M., Fung, S., Zhu, Y., Gura, N., and Eberle, S., & Chang, H. (2005). Sizzle: a standards-based end-to-end security architecture for the embedded internet. In *IEEE PerCom*, pages:247-256, March 2005.

Hartung, C., Balasalle, J., & Han, R. (2005). Node compromise in sensor networks: the need for secure systems. In *Technical Report CU-CS-990-05, Department of Computer Science, University of Colorado at Boulder*, January 2005.

Hu, L. & Evans, D. (2003). Using directional antennas to prevent wormhole attacks. In *Proceedings of the 11th Network and Distributed System Security Symposium*, pages 131–141, February 2003.

Hu, Y.C., Perrig, A., & Johnson, D.B. (2003). Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM 2003*, April 2003.

Hu, X., Park, T., & Shin, K. G. (2008). Attack-tolerant time-synchronization in wireless sensor networks. In *IEEE INFOCOM*, pages:41-45, April 2008.

Jung, J., Paxon, V., Berger, A.W. & Balakrishnan, H. (2004). Fast port scan detection using sequential hypothesis testing. In *IEEE Symposium on Security and Privacy*, pages:211-225, May 2004.

Karlof, C. & Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks Journal*, 1(2-3):293-315, September 2003.

Li, Z., Trappe, W., Zhang, Y., & Nath, B. (2005). Robust statistical methods for securing wireless localization in sensor networks. In *IEEE IPSN*, pages:91-98, April 2005.

Li, F., & Wu., J. (2007). Mobility reduces uncertainty in {MANET}. In *IEEE INFOCOM*, pages:1946-1954, May 2007.

Liu, A. & Ning, P. (2008). TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks. In *IEEE IPSN*, pages:245-256, April 2008.

Liu, D. & Ning, P. (2003). Establishing pariwise keys in distributed sensor networks. In *ACM CCS*, pages:52-61, October 2003.

Liu, D., Ning, P., & Du, W. (2005). Attack-resistant location estimation in sensor networks. In *IEEE IPSN*, pages:99-106, April 2005.

Malan, D., Welsh, M., & Smith, M. (2004). A public-key infrastructure for key distribution in tinyOS based on elliptic curve cryptography. In *IEEE SECON*, pages:71-80, October 2004.

Park, T. & Shin, K. G. (2005). Soft tamper-proofing via program integrity verification in wireless sensor networks. In IEEE Trans. Mob. Comput., 4(3):297-309, 2005.

Parno, B., Perrig, A., and Gligor, V.D. (2005). Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pages:49-63, May 2005.

Parno, B., Luk, M., Gaustad, E., and Perrig, A. (2006). Secure sensor network routing: a cleanslate approach. In *ACM CoNEXT*, December 2006.

Przydatek, B., Song, D., & Perrig, A. (2003). {SIA}: secure information aggregation in sensor networks. In *ACM SenSys*, pages:69-102, November 2003.

Seshadri, A., Perrig, A., van Doorn, L., & Khosla, P. (2004). {SWATT}: softWare-based attestation for embedded devices. In *IEEE Symposium on Security and Privacy*, pages:272-282, May 2004.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages:47-53, August 1984.

Shaneck, M., Mahadevan, K., Kher, V., & Kim, Y. (2005). Remote software-based attestation for wireless sensors. In *ESAS*, July 2005.

Song, H., Zhu, S., & Cao, G. (2007). Attack-resilient time synchronization for wireless sensor networks. *Ad Hoc Networks*, 5(1):112–125, January 2007.

Sun, K., Ning, P., Wang, C., Liu, A., & Zhou, Y. (2006). TinySeRSync: secure and resilient time synchronization in wireless sensor networks. In *ACM CCS*, pages:264-277, 2006.

Sun, Y., Han, Z., Yu, W., & Liu, K. (2006). A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks. In *IEEE INFOCOM*, pages:1-13, April 2006.

Tague, P. & Poovendran, R. (2008). Modeling node capture attacks in wireless sensor networks. In *Allerton Conference on Communication, Control, and Computing* , September 2008.

Wald, A. (2004). Sequential analysis. *Dover Publications*, 2004.

Wang, H., Sheng, B., Tan, C.C., & Li, Q. (2008). Comparing symmetric-key and public-key based security schemes in sensor networks: a case study of user access control. In *IEEE ICDCS*, pages:11-18, 2008.

Wood, A. D. & Stankovic, J. A. (2002). Denial of service in sensor networks. *IEEE Computer* 35(10):54–62, 2002

Xing, K., Liu, F., Cheng, X., & Du, H.C. (2008). Real-time detection of clone attacks in wireless sensor networks. In *IEEE ICDCS*, pages:3-10, June 2008.

Yang, Y., Wang, X., Zhu, S., & Cao, G. (2006). {SDAP}: a secure hop-by-hop data aggregation protocol for sensor networks. In *ACM MOBIHOC*, 2006.

Yang, Y., Wang, X., Zhu, S., & Cao, G. (2007). Distributed software-based attestation for node compromise detection in sensor networks. In *IEEE SRDS*, pages:219-230, October 2007.

Ye, F., Luo, H., Lu, S., & Zhang, L. (2004). Statistical en-route filtering of injected false data in sensor networks. In *IEEE INFOCOM*, 2004.

Ye, F., Yang, H., & Liu, Z. (2007). Catching moles in sensor networks. In *IEEE ICDCS*, June 2007.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, August 2008.

Yu, L. & Li, J. (2009). Grouping-based resilient statistical en-route filtering for sensor networks. To appear in *IEEE INFOCOM*, April 2009.

Zhang, Y., Yang, J., Jin, L., & Li, W. (2006). Locating compromised sensor nodes through incremental hashing authentication. In *DCOSS*, June 2006.

Zhang, W., Tran, M., Zhu, S., & Cao, G. (2007). A random perturbation-based scheme for pairwise key establishment in sensor networks. In *ACM Mobihoc*, pages:90-99, September 2007.

Zhu, S., Setia, S., Jajodia, S., & Ning, P. (2004). An interleaved by hop-by-hop authentication scheme for filtering injected false data in sensor networks. In *IEEE Symposium on Security and Privacy*, pages:259-271, May 2004.

Zhu, B., Addada, V.G.K., Setia, S., Jajodia, S., & Roy, S. (2007). Efficient distributed detection of node replication attacks in sensor networks. In *ACSAC*, pages:257-267, December 2007.

21

Integrity Enhancement in Wireless Sensor Networks

Yusnani Mohd Yussoff, Husna Zainol Abidin and Habibah Hashim
Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia

1. Introduction

Consideration for security level in Wireless Sensor Networks (WSN) should depend on the demand of the intended applications. As energy consumption increase linearly with security level, the security designer should carefully choose the best security technique and the most suitable security parameters enough to protect the intended application. With the advancement and demand of WSNs applications in areas such as the military, structural health monitoring, transportation, agriculture, smart home and many more, the system stands to be exposed to too many potential threats. It is generally considered that applications such as smart home, transportation and agriculture need no security or be less secure compared to military and medical applications. However, sensor networks make large-scale attacks become trivial when private information on the entire system can instantly reach the hand of attackers. Due to the nature of WSNs that are left unattended and limited resources, there exist an urgent need for higher security features in sensor nodes and its overall systems. Without it, attackers with their own intentions and targets combined with their capabilities and sophisticated tools will always become a threat to future WSNs applications. However, latest technology in embedded security combined (low power, on-SOC memory, small size) with trusted computing specifications (ensuring trusted communication and user) is believed to enhance security features for future WSNs applications.

To this instant, research in the security area of WSNs covers development of new security algorithms that consume low energy and memory (Perrig *et al.*, 2002), comparison of energy efficient security algorithm including Public Key Cryptography (PKC) and symmetry cryptography technique (Pathan & Choong Seon, 2008) and finally hardware implementation of security algorithms (Ekanayake *et al.*, 2004, Gaubatz *et al.*, 2005, Huai *et al.*, 2009, Huang & Penzhorn, 2005, Kocabas *et al.*, 2008a, Lee *et al.*, 2008, Suh *et al.*, 2005). Our work is basically inspired by (Grobschadl *et al.*, 2008) suggesting hybrid implementations in securing WSNs applications.

The rest of the paper is organized as follows: Section 2 presents security challenges in WSN area. Section 3 briefly define physical attacks in WSNs. Section 4 will discusses the trusted

platform techniques followed by section 5 which focusses on the related studies on hardware based security for WSN and subsequently section 6 presents the proposed security work. Finally section 7 concludes the paper.

2. Security Challenges in WSN

Security challenges in WSNs can be divided into three different categories that are related to each other. [1] *Network*–Ensuring reliable, secure and trusted communication. [2] *Data*–Ensuring the integrity of the transmitted and processed data and finally [3]*Platform*-Guarantee the integrity of the sensor node exist in the network. Future applications such as medical health, military, system monitoring, smart home and many more, demand higher security levels that include access control, explicit omission or freshness, confidentiality, authenticity and integrity (Verma, 2006). Detailed analysis of security demand in various types of applications with potential security threats can be found in (Amin *et al.*, 2008a). Fig. 1, briefly shows common security goals of WSN based on the works of F.Amin and N.Verma . In order to achieve the above goals, PKC is believed to be capable of supporting asymmetric key management as well as authenticity and integrity. Although the use of PKC in WSN is previously denied due to its high resourced (energy, memory and computational) (Yong *et al.*, 2006), many recent works have proved its feasibility in the WSN area (Kocabas *et al.*, 2008b). Latest, Wen Hu (Hu *et al.*, 2009) used Trusted Platform Module hardware which is based on Public Key (PK) platform to augment the security of the sensor node. They claim that the SecFleck architecture provides internet level PK services with reasonable energy consumption and financial overhead.

Future applications such as medical health, military, system monitoring, smart home and many more, demand higher security levels that include access control, freshness, confidentiality, authenticity and integrity (Verma, 2006). Detailed analysis of security demand in various types of applications with potential security threats can be found in (Amin *et al.*, 2008a). Listed goals in Fig. 1, are achievable through PKC implementation supporting asymmetric key management as well as authenticity and integrity. Although the use of PKC in WSN is previously denied due to its high resourced (energy, memory and computational) (Yong *et al.*, 2006), many recent works have proved its feasibility in the WSN area (Kocabas *et al.*, 2008b). Latest, Wen Hu (Hu *et al.*, 2009) used Trusted Platform Module hardware which is based on Public Key (PK) platform to augment the security of the sensor node. They claim that the SecFleck architecture provides internet level PK services with reasonable energy consumption and financial overhead.

It can be concluded that the demand for higher security levels in WSN increase significantly with the advancements in WSN applications. As mentioned earlier, the feasibility of PKC in WSN security is proven and therefore the choice of PKC as the best cryptography protocol in WSN area has been established. The concern now is what is the best method to implement PKC in the sensor node and is it secure to run security protocol in on unsecured platform considering the nature of the WSN node that is normally expose to software attack and physical attack? Security provided by cryptography depends on safeguarding of cryptographic keys from adversaries. Therefore there is a need to adequately protect the keys to ensure confidentiality and integrity of sensitive data. While majority of the work

done in WSN security have focused on the security of the network (Hu *et al.*, 2009), our proposed works will consider the three challenges describe earlier to secure the WSNs applications from software and physical types of attacks. Beside we will also ensure smallest security parameter in our overall security design.

At this stage, the authors believe that embedding the security parameters in the processor is the most suitable technique for securing wireless sensor node. This technique is believed to be capable of reducing the size of the sensor node, decreasing the processing time and preventing software and physical attacks as well as providing other benefits. Johann et al. in his paper (Grobschadl *et al.*, 2008) also conclude that hardware based security features need to be integrated into the processor to avoid vulnerabilities such as those which exist in today's personal computer. Besides secure implementation, the node also should communicate in a trusted environment. Tiago and Don (Alves *et al.*, 2004) mentioned that the demand in trusted computing is driven by the potentially severe economic consequences due to unsecured embedded applications. Following section will only consider security design for the third type of security challenges with the intention to secure the sensor node from physical attacks and ensure the integrity of the sensor node in the network.

3. Physical Attacks in WSN

Effect on attacks to WSNs applications can either be direct or indirect. While the first can cause disclosure of private information, modification and falsification of data and sensor node failure, the latter will basically cause unreliable services to the WSNs applications such as low data rate, service breakdown and inconsistent communication. Both effects are mostly the result of physical attacks or node tampering.

Tampering
Tampering as defined by A.Becher et.al (Becher *et al.*, 2006) is the ability to get full access to the node and it involves a modification to the internal structure of the chip. Physical attacks on the other hand are referring to attacks that require direct physical access to the sensor node. W.Znaidi et al. On the other hand, defined tampering as an action that involved physical access and node capture (Znaidi *et al.*, 2008). To avoid terminology problem, 'tampering' in this paper is as defined by A.Becher et al. and is seen as impossible in WSNs application as it involved sophisticated tools and takes a longer time to complete (Base station may have terminated communication with this sensor node by this time). Therefore it is not as likely to happen as the attacks that can be carried out in the field.

Physical Attacks
As defined earlier, physical attacks refer to attacks that involves direct connection with the sensor node. Adversaries may perform the attack by connecting their sophisticated tools on the site or taking away the sensor node. Their intention might vary from just to destroy the sensor node to extracting private information to be authenticated or authorized in the network. Sensor nodes can usually be attacked through the JTAG port that is widely used during the development phase and for debugging. With the JTAG port being enabled, adversaries will have the capability to take control of the whole system. Another form of attack is by exploiting the Bootstrap Loader (BSL) and this mostly happens during the boot up

process. With having access to the boot devices and debug session, attackers will be able to study the systems and its operation thus providing them with enough information to clone the system, insert malware and disturb the overall operations of the sensor node and its systems.

Although a total solution to physical attacks are almost impossible, designers should concentrate on methods to secure and protect the sensitive information from physical attacks. The paragraph below discusses possible solutions towards confirming the integrity of codes running in the sensor node and protecting highly sensitive data through Trusted Computing and TrustZone technology.

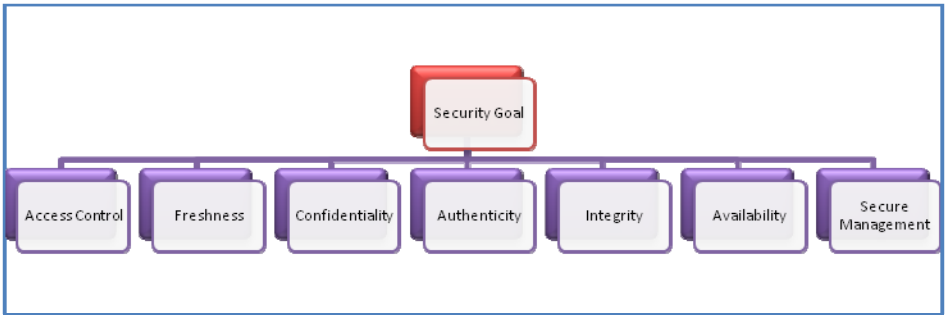


Fig. 1. Common Security Goals in Wireless Sensor Networks

4. Trusted Platform Technique

It is believed that nothing is secured and can be trusted. With enough time and money, attackers will definitely find a way to break and attack any systems. Therefore a clear definition of a trusted system is needed. According to (Grawrock, 2009), trust can be defined as an entity that always behaves in the expected way for the intended function. Basic properties of a trusted computer or systems [referenced from?]can be listed as below.

- Isolation of programs – prevent program A from accessing data of program B
- Clear separation between user and supervisor process – there should be a systems to prevent user applications from interfering with the operating system.
- Long term protected storage – secret values are stored in a place that last across power cycles and other events.
- Identification of current configuration – provide identity of the platform and software or hardware executing on it.
- Verifiable report of the platform identity and current configuration – a way for other users to validate a platform.
- Hardware basis for the protections - protection is a combination of hardware and software.

Demand on a trusted platform in the network environment arrived when merely software based mechanisms became inadequate to provide the desired security level. Trusted Computing Platform Alliance (TCPA) was formed in late 90's and finally emerged as the Trusted Computing Group (TCG) in 2003 (Groups, 2008). TCG has basically worked to develop an inexpensive chip that helps users protect their sensitive information.

Muhammad Amin (Amin *et al.*, 2008b) in his paper discussed on the trends and directions in trusted computing. His paper provides details on advancement of trusted hardware to facilitate security that led to the design and implementation of TCG specific solution. This paper also claims that ARM is the only trusted implementation available for secure embedded applications.

The following section discusses two alternatives that can be used to establish trusted and secure security systems followed by review on hardware-based security implementation.

4.1 Trusted Platform Module

Trusted Computing Groups (TCG) solves security problems through operating environments, applications and secure hardware changes to the personal computer. TCG used secure hardware Trusted Platform Module (TPM) chip as a basis for trusted computing that provides a level of relevant since hardware based security is difficult to compromise than conventional approaches.

TPM verifies the integrity of the system through trusted boot, strong process isolation and remote attestation that verify the authenticity of the platform. Encryption and decryption used RSA algorithm with default 2048-bit, SHA-1 hash and random key generator. TPM can be implemented in a dedicated chip, co-processor or in software (Grobschadl *et al.*, 2008) where the configuration of TPM is vendor specific and is not specified by TCG. Fig. 2 briefly shows block diagram of TPM consisting of ten components to accelerate security processes.

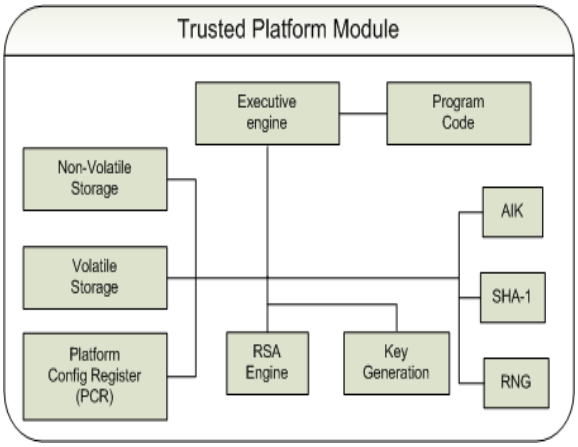


Fig. 2. Standard TPM Components

Unfortunately, the choice of RSA and SHA-1 algorithms has made the platform unsuitable for WSN applications. RSA with 2048 bits has been confirm to consume higher energy and therefore unsuitable for WSN applications and embedded system (Amin *et al.*, 2008a). Moreover, RSA when implemented in hardware demand large silicon area and therefore increase the size of the chip (Kocabas *et al.*, 2008b). An alternative to RSA is Elliptic Curve Cryptography (ECC) and Advance Encryption System (AES). Beside RSA, the choice of

SHA-1 is also mooted. Recent research indicates that many cryptographers doubt the security of SHA-1 and recommend against the use in new design.

To conclude, TPM model may not be the best choice for secure or trusted platform implementation in embedded systems especially in WSN applications due to the performance and security concern. Most importantly, the TPM is designed for the personal computer which does not usually have concerns on resource constraints.

4.2 Trust Zone in ARM Microprocessor

The key feature of the ARM trust zone is "secure to the core". The security features are hard wired into the microprocessor core and therefore promise an extra degree of security over a software only approach and external security chip approach (Halfhill, 2003).

The ARM trust zone is specifically designed for smart phones, handheld devices and embedded systems that can potentially be compromised by malicious hackers. The nature of WSN that exposes it to too many types of attacks and intrusions demand extra security features that not only support security but also trustworthiness.

Wilson et. al (Wilson *et al.*, 2007) in his paper viewed trustzone in ARM as a dual-virtual CPU Systems. The running software looks at the trustzone as two separate virtual processors. The virtualization is achieved through hardware extension within the CPU design. The extensions annotate whether the core is running Normal World or Secure World software and propagate these selections to the memory and peripherals. With this implementation, the secure memory and peripherals can reject the non-secure transactions.

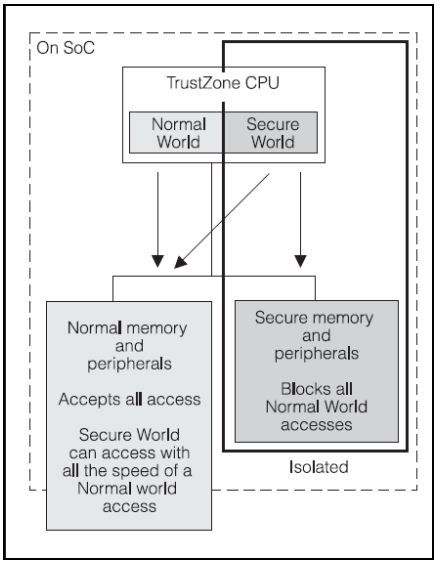


Fig. 3. One core support two operating worlds: secure world and normal world. Courtesy of: Wilson.P et.al (Wilson *et al.*, 2007)

The switching between secure and non-secure world in the ARM processor is established through the Secure Monitor Call (SMC) instruction and interrupts. In line with WSN constraints, the trust zone in the ARM processor eliminates the need for extra security chip. Moreover, security elements can be executed at full processor speed without cache-flushing overhead. It can also save the power as only one of the two virtual processors run at one time. Fig. 3 shows how trustzone mimics two processors.

5. Related Studies

G.Edward Suh et.al (Suh *et al.*, 2007) in his work presented an AEGIS secure processor architecture that secure the embedded system beyond normal security algorithm. AEGIS, a single-chip secure processor, introduces mechanisms that not only authenticate the platform and software but also protect the integrity and privacy of applications from physical attacks. Two new techniques are introduced to overcome physical and software attacks in WSN, Physical Random Functions (PUFs) and off-chip memory functions.

Physical Random Function (PUFs) is a function that generates secret numbers so that users can authenticate the processor that they are interacting with. With PUFs the secret are generated dynamically by the processor and therefore provide higher physical security compared to storing the secrets in non-volatile memory. Besides, PUFs also do not need any special manufacturing process or special programming and testing steps.

Off-chip memory mechanisms ensure the integrity and the privacy of off-chip memory by encrypting and decrypting all off-chip memory data transfer using a one-time pad encryption scheme. To summarize, AEGIS can protect embedded devices from any attacks before program execution, during the execution and also from physical and software attacks through the security mechanism designed. Unfortunately, the added hardware mechanisms had increased the size of the processor core and marginally degrade program performance.

Lie et. al. (Lie *et al.*, 2000) from Stanford University introduced Execute Only Memory (XOM) that enabled copy and tamper resistant software distribution to prevent software piracy. All data leaving the machine is encrypted using symmetric-key encryption and the keys are specifically distributed to each processor using public-private key pair. This technique provides a software tamper-resistant execution environment that is established through tagging or encryption. Unfortunately, hardware assist is considered necessary in XOM architecture to provide fast symmetric ciphers.

SecFleck (Hu *et al.*, 2009) which was mentioned earlier used external TPM chip on the sensor node. This TPM based public key platform facilitates message security services with confidentiality, authenticity and integrity. SecFleck platform consists of hardware and software module and later connects to the Fleck sensor node board. Although the evaluation on the computation time, energy consumption, memory footprint and cost is reasonable and positive, the extra platform connected to the sensor node is unacceptable for sensor node applications. Besides the security algorithm used is not aligned with sensor node constraints.

Another work on hardware based security is done by (PANIANDI, 2006, Pin, 2009) where both works developed a co-processor for security algorithm. While the first work developed RSA co-processor, the second work implements an AES co-processor (VHDL design only) for resource constraint embedded system. RSA co-processor was implemented on Altera Stratix FPGA development board. Both works claim to have better speed and area compared to other research and commercial implementation.

Latest, two studies have embarked on the development of trusted and secure platform utilizing ARM11 trustzone architecture. Johannes Winter(Winter, 2008) and Xu Yang-ling(Xu *et al.*, 2008), both utilize Linux kernel 2.6 and ARM trustzone features. While Johannes merge trustzone features with TCG-style trusted computing concepts, Mobile Trusted Module (MTM), Xu integrate the Mandatory Access Control (MAC) in Linux kernel 2.6 with the trustzone features to enhance the security up to the non-secure environment. The first has designed a robust and portable virtualization framework for handling non-secure guest and the second work presented an embedded system security solution.

6. Proposed Work

This work proposes the development of a sensor node platform utilizing ARM11, a 32-bit processor. This work was prompted due to lack of highly secured sensor node platform to accommodate future wireless sensor networks applications. Almost all available sensor node platforms (Healy *et al.*, 2008) utilize software based security. This work proposed the use of trustzone feature in the ARM11 processor to enhance the security level by limiting the security parameter to a single chip. All important keys and data will be saved in the On-SoC memory thus provide better shielding to private information on the platform.

6.1 Security Architecture

The primary goals are to assert the integrity of the software images executed in the sensor node platform by preventing any unauthorized or malicious modified software from running and to ensure the confidentiality and integrity of the data during communications.

The above objectives are established through proper security architecture designed utilizing ARM trust zone features.

- Secure world – all the sensitive resources will be placed in the secure world memory locations. Trust zone Address space controller (TZASC) is used to configure regions as secure or non-secure. All non-secure process will be rejected to the region that is configured as secure. This ensures the confidentiality of important data.
- Single physical core – safe and efficient execution of code from both normal and secure world. This allows high performance security software to run alongside with normal world operating environment. Secure monitor code will be developed to switch from normal to secure and vice versa.
- Secure boot – Running secure boot algorithm to ensure the integrity of the software images and devices on the platform.
- On-Soc RAM and ROM will ensure no highly sensitive data leaves the chip thus eliminating the possibility of physical attacks.
- * Identity based Encryption Algorithm for confidentiality and integrity of the data during communications. (Communications between sensor node and base station)

By using ARM trust zone, a small on-chip security system is presented in Fig. 4 below to execute the above objectives. It clearly depicts the permanent secure place and dynamic secure place that are accessible through AXI2APB bus system which has the capability to switch from secure process and non-secure process. Trust Zone Memory Adapter (TZMA)

will secure a region within an on-SoC memory such as SRAM where the secure location will be in the lower part of the memory region.

*Not discussed in this paper.

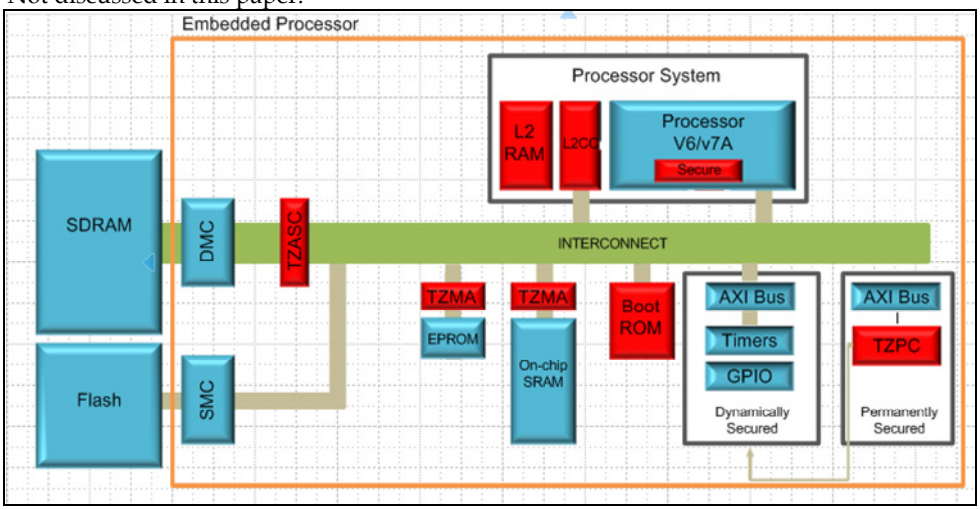


Fig. 4. Proposed security architecture for sensor node using ARM11 with Trust zone features.

Trust zone Address Space Controller (TZASC) will reject any non-secure transaction to a region that is configured as secure. Therefore external memory also can be partitioned into secure and non-secure region. Compared to previous works, the proposed security architecture has extended the security infrastructure throughout the system design. Instead of protecting assets in a dedicated hardware block, this architecture has made the valuable assets secured in the most protected location.

On top of the hardware design, a suitable security protocol such as secure boot will also be configured to complete the security design. Secure boot with the root of trust located in On-SoC ROM will provide a chain of trust for all the secure world software and hardware peripherals and some of the normal world software. With secure boot, the integrity of the OS image, software and peripherals on the platform can be verified to be truly unadulterated. Communications right after the secure boot process can be confirmed coming from a trusted sensor node.

Table 1 clearly depicts the advantage of the proposed security mechanism over previous work. Although the security level of the second technique is comparable with the proposed work, this proposed scheme offers extra advantages in term of power consumption and overall performance. While in AEGIS for example two processors are needed to run secure and normal process, in trustzone the dual virtual CPU will execute one of the processes (secure or non-secure) at one time thus eliminate extra processing work and reducing the chip size. Moreover, AEGIS works is does not consider WSNs constraints. Finally, since extra chip on the embedded applications board are not desirable, the first technique or work can be considered as irrelevant for WSN security implementation.

Previous Worked	Definition	Advantage	Drawback	Secure(S) Trusted (T)	Attacks Physical (PHY) Software (SW)	Consider WSN constraints?
External Hardware **TPM – RSA [3]** **TPM - IBE [18]** **AES - [5]** **RSA – [4][19]**	I Inclusion of a dedicated hardware security module outside of the main processor	Separate chip. Allows high levels of tamper resistance and physical security.	Sensitive resources leave the chip. Increase area and power consumption Physical attacks	T&S T&S S S	SW	**NO**
Embedded Hardware **AEGIS - AES[1]** **XOM- [2]**	Hardware security modules that is located within the SoC.	Significant cost reduction performance improvement over external hardware. Security is comparable to trust zone technique.	Restricted perimeter and only capable of securing on-chip components. Not flexible	T&S S	SW & PHY	**NO**
Embedded security H/W with Dual Virtual CPU (Trustzone (TZ)) **TZ+MTM [6]** **TZ+MAC [7]**	Hardware architecture that extends the security infrastructure throughout the system design. Trustzone architecture enables any part of the system to be made secure.	Significant cost reduction Performance improvement over external h/ware. Only one process exist at one time (secure or non-secure)- reduce power Secure all sensitive resources. Flexible design-can secure up to off-chip components	For mobile appliances	T&S T&S	SW & PHY	**NO** **NO**
Proposed work **ARM11 with Trustzone**	**As above**	**As Above**	**For sensor node**	**T&S**	**SW & PHY**	**YES**

Table 1. Comparison Study on Trusted Implementation for Wireless Sensor Network

7. Conclusion

The security features discussed earlier are intended for highly secure applications dealing with crucial financial information, noncritical military communications, medical data, and critical

corporate information. Detail on security level can be found in (Groups, 2010). Two dominant features that differentiate this work from others are the placement of sensitive resources such as the crypto keys within the embedded system and the denial of extra or dedicated processor core for security purposes. This implementation ensures no sensitive resources leaves the chip and therefore blocks most types of attacks. Besides that it also saves the silicon area and power consumption and also allows high performance security software to run alongside with the normal world operating environment. It is hoped that the outcome from this work can contribute towards higher security level in the area of WSN. Finally the choice of ARM11 as the main processor for the sensor node is in line with the constraint faced in sensor node development as it is rated as the most efficient processor in MIPS/Watt (Vieira *et al.*, 2003).

8. References

Alves, T., D. Felton & ARM (2004): TrustZone: Integrated Hardware and Software Security. In *Technology in-Depth*: 18(Ed)^(Eds).

Amin, F., A. H. Jahangir & H. Rasifard (2008a): Analysis of Public-Key Cryptography for Wireless Sensor Networks Security. *World Academy of Science, Engineering and Technology***:** 529.

Amin, M., S. Khan, T. Ali & S. Gul (2008b): Trends and Directions in Trusted Computing: Models, Architectures and Technologies. In *International MultiConference of Engineers and Computer Scientists*(Ed)^(Eds). Hong Kong.

Becher, A., Z. Benenson & M. Dornself (2006): *Security in Pervasive Computing*: Springer Berlin/Heidelberg.

Ekanayake, V., I. Clinton Kelly & R. Manohar (2004): An ultra low-power processor for sensor networks. In *Proceedings of the 11th international conference on Architectural support for programming languages and operating systems*(Ed)^(Eds). Boston, MA, USA: ACM.

Gaubatz, G., J.-P. Kaps, E. Ozturk & B. Sunar (2005): State of the ART in Ultra Low Power Public key Cryptography for Wireless Sensor Network. In *3rd International Conference on Pervasive Computing and Communications Workshop*(Ed)^(Eds): IEEE Computer Society.

Grawrock, D. (2009): *Dynamics of a Trusted Platform*: Intel Press.

Grobschadl, J., T. Vejda & D. Page (2008): Reassassing the TCG Specifications for Trusted Computing in Mobile Embedded Systems. In *1st IEEE Workshop on hardware-Oriented Security and Trust HOST2008*: 84(Ed)^(Eds): IEEE.

Groups, E. T. (2010): Cryptography for embedded systems(Ed)^(Eds): EE Times Network.

Groups, T. C. (2008): Trusted Platform Module(TPM) Summary(Ed)^(Eds): Trusted Computing Groups.

Halfhill, T. R. (2003): ARM DONS ARMOR: Trustzone Security Extensions Strengthnen ARMv6 Architecture. In *MIcroprocessor*(Ed)^(Eds). Arizona: Reed Electronics Group.

Healy, M., T. Newe & E. Lewis (2008): Wireless Sensor Node hardware: A review. In *Sensors, 2008 IEEE*: 621(Ed)^(Eds).

Hu, W., P. Corke, W. C. Shih & L. Overs (2009): SecFleck: A public key technology platform for wireless sensor networks: 296(Ed)^(Eds). Cork, Ireland: Springer Verlag.

Huai, L., X. Zou, Z. liu & Y. Han (2009): An Energy Efficient AES-CCM Implementation for IEEE802.15.4 Wireless Sensor Networks. In *2009Internatioanal Conference on networks Security, Wireless Communications and Trusted Computing*: 394(Ed)^(Eds): IEEE Computer Society.

Huang, A. L. & W. T. Penzhorn (2005): Cryptographic Hash Functions and Low-Power Techniques for Embedded Hardware. In *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*: 1789(Ed)^(Eds).

Kocabas, O., E. Sabas & J. Grobschadl (2008a): Enhancing an Embedded Processor Core with a Cryptographic Unit for Performance and Security In *4th International Conference on Reconfigurable Computing and FPGAs*: 409(Ed)^(Eds): IEEE.

Kocabas, O., E. Savas & J. Grossschadl (2008b): Enhancing an Embedded Processor Core with a Cryptographic Unit for Speed and Security. In *Reconfigurable Computing and FPGAs, 2008. ReConFig '08. International Conference on*: 409(Ed)^(Eds).

Lee, Y. K., k. Sakiyama, L. Batina & I. Verbauwhede (2008): Elliptic-Curve-Based Security processor for RFID(Ed)^(Eds): IEEE Computer Society.

Lie, D., C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell & M. Horowitz (2000): Architectural support for copy and tamper resistant software. *SIGPLAN Not.* **35:** 168.

PANIANDI, A. (2006): *A Hardware Implementation of Rivest-Shamir-Adleman Co-Processor for Resource Constrainted Embedded Systems*. Master, Universii Teknologi Malaysia, Skudai.

Pathan, A. S. K. & H. Choong Seon (2008): Feasibility of PKC in resource-constrained wireless sensor networks. In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*: 13(Ed)^(Eds).

Perrig, A., R. Szewczyk, J. D. Tygar, V. Wen & D. E. Culler (2002): SPINS: security protocols for sensor networks. *Wirel. Netw.* **8:** 521.

Pin, L. Y. (2009): *Verilog Design of a 256-bits AES Crypto Processor Core*. Master, Universiti Teknologi Malaysia, Skudai.

Suh, G. E., C. W. O'Donnell & S. Devadas (2005): AEGIS: A single-chip secure processor. *Information Security Technical Report* **10:** 63.

Suh, G. E., C. W. O'Donnell & S. Devadas (2007): Aegis: A Single-Chip Secure Processor. *IEEE Des. Test* **24:** 570.

Verma, N. (2006): *Practical Implementation and Performance Analysis On Security of Sensor Networks*. MSc Full Thesis, Rochester Institute of Technology, Rochester, New York.

Vieira, M. A. M., C. N. Coelho, Jr., D. C. da Silva, Jr. & J. M. da Mata (2003): Survey on wireless sensor network devices. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*: 537(Ed)^(Eds).

Wilson, P., A. Frey, T. Mihm, D. Kershaw & T. Alves (2007): Implementing Embedded Security on Dual-Virtual-CPU Systems. In *IEEE Design and Test of Computers*: 582(Ed)^(Eds): IEEE Computer Society.

Winter, J. (2008): Trusted computing building blocks for embedded linux-based ARM trustzone platforms. In *Proceedings of the 3rd ACM workshop on Scalable trusted computing*(Ed)^(Eds). Alexandria, Virginia, USA: ACM.

Xu, Y.-l., W. Pan & X.-g. Zhang (2008): Design and Implementation of Secure Embedded Systems Based on Trustzone. In *Embedded Software and Systems, 2008. ICESS '08. International Conference on*: 136(Ed)^(Eds).

Yong, W., G. Attebury & B. Ramamurthy (2006): A survey of security issues in wireless sensor networks. *Communications Surveys & Tutorials, IEEE* **8:** 2.

Znaidi, W., M. Minier & J.-P. Babau (2008): An Ontology for Attacks in Wireless Sensor Networks(Ed)^(Eds). Montbonnot Saint Ismier: National De Recherche En Informatique Et En Automatique.

Technologies and Architectures for Multimedia-Support in Wireless Sensor Networks

Sven Zacharias and Thomas Newe
University of Limerick
Ireland

1. Introduction

Wireless Sensor Networks (WSNs) are an emerging technology in the area of sensory and distributed computing. A WSN consists of many, theoretically up to some thousand or even millions of sensor nodes. A sensor node is generally defined as a cheap and small piece of hardware, which consists of four main units:

- One or more **sensors** that detect physical phenomena. Common sensors monitor scalar values of temperature, pressure, humidity, light intensity, etc.
- The sensor is coupled with a **data processing unit**. The latter controls sensing, application logic and network transfer. It receives data from the sensors as well as it can filter (e.g. thresholding), compress or correlate data from a series of measurement. The network structure, the communication process and the power management of the node are also organized by the processing unit.
- The data's wireless transmission is provided by a **communication interface**. Most nodes' transfer is usually based on the IEEE 802.15.4 standard because of the low power consumption of this transfer technology and the availability of low cost radios.
- For every operational electronic system an **energy source** is needed. Although significant progress has been achieved in the area of energy harvesting, today's standard power supply for sensor nodes is still the battery.

Generally sensor nodes are designed to be widely spread without pre-configuration. A sink, also called a base station, is normally an embedded or a personal computer which is configured to collect, save or react according to the data. The network between the nodes and the sink is built dynamically and is considered to be self-organizing.

1.1 Specifications of Wireless Multimedia Sensor Networks

In contrast to the scalar data collected by classical WSNs, some applications need to collect multimedia (mm) based data. Mm data can be defined as image, video or sound. These types of data are relatively large and are likely to be represented in an array or stream. Due to the greater amount of data, processing operations on these are more calculation-intensive than on scalar data. So a new class of WSNs has been developed to sense mm data. These Wireless Multimedia Sensor Networks (WMSNs) form a special group of WSNs and need new designs to master their challenges. The main challenges resulting from the amount of produced data are:

- The wireless link has to provide a reliable and fast connection to transmit the produced amount of data. As wireless transfer is quite power consuming and the nodes are mostly battery-driven, the power management of a WMSN has to be sophisticated in order to overcome the power shortage.
- Data can be either transferred as a stream (e.g. a live video) or as snapshot (e.g. a single picture). The requirements for streaming are a high sampling rate as well as a fast connection, which satisfies Quality of Service demands. Therefore packages can be dropped, because reordering and retransferring of old packages would disturb time synchronization even more. On the other hand all packages have to be delivered for a snapshot, and reordering or retransmitting may have to be established.
- Beside the network, the node itself has to do more calculations and therefore has a higher need for performance than in classical scalar WSNs. The tasks on the node include compression and event detection, which can be solved either on the node alone or in cooperation with distributed algorithms.

Not only for the design of the system, but also for the deployment of the nodes a significant difference has to be taken into account.

- Image sensors have a field of view. Although sound propagates wavelike, microphones have directionality, which means they are variably sensitive to sound at different angles. So that mm sensors should be deployed with caution and by plan.

The ideal mm node should have a lot of processing power to work with the data, a high speed network to transfer it, a strong power source to keep the system running and it should be carefully deployed. These demands stand in open contrast to the idea of classical WSNs. In return to all these high demands, nodes with mm capabilities make a wide range of novel applications possible. The remainder of this chapter will give an overview of available wireless transfer standards including their capabilities and limits, as well as node hardware for mm support and the commonly used design patterns for system architectures in WMSNs. Although protocols are an important part of WSNs and should fulfil special requirements to be used in WMSNs, they are not part of this chapter due to page limitation. Likewise, data gathering and mining algorithms as well as software in general are not covered in this chapter.

1.2 Related Work

The basic ideas and tasks of WSNs were presented in (Pottie & Kaiser, 2000) about ten years ago. A good general overview about applications of WSNs is given in (Akyildiz et al., 2002) and (Arampatzis et al., 2005). An overview of WMSNs is provided by Akyildiz et al. in (Akyildiz et al., 2007a), (Akyildiz et al., 2007b) and (Akyildiz et al., 2008). Römer and Mattern introduce classification criteria for the WSN design space and present applications ranked by their classifications (Roemer & Mattern, 2004). Melodia published detailed work on connecting mm sensor nodes to actors and has also proposed a heterogeneous architecture for his system (Melodia, 2007).

2. Wireless Transmitting Technologies

In this section a short overview of the most important wireless technologies is given. The wireless transfer is the main critical task in WSNs. It needs a lot of energy and the limited transfer range is a key factor for the network topologies. Thus the underlying technologies have to be understood in order to understand the design of WSNs.

2.1 IEEE 802.15.4

Region	*Frequency band (MHz)*	*Communication channels*	*Data rate per channel (kb/s)*
Worldwide	2,400.0 – 2,483.5	16 channels	250
North America	902.0 – 928.0	10 channels (2003), 30 channels (2006)	40 (2003), 250 (2006)
Europe	868.0 – 868.6	1 channel	20 (2003), 100 (2006)

Table 1. The Industrial, Scientific and Medical (ISM) bands used in IEEE 802.15.4.

The Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 standard (*IEEE Std 802.15.4-2003*, 2003) is designed for very low-power Wireless Personal Area Networks (WPAN). The Physical Layer and accompanying MAC protocols of the Data Link Layer are defined by this standard. The medium access operates based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). One of the three unlicensed Industrial, Scientific and Medical (ISM) frequency bands is used for transfer. In the basic IEEE 802.15.4 (2003) standard there is a total number of 27 channels in the ISM bands. The properties of the free frequency bands are shown in Table 1. The typical usage range is 30 – 50 m and can reach up to 100 m. The data throughput is low, but a 30 ms network join time can be achieved. 802.15.4 can be used for different topologies, like star or peer-to-peer. For energy efficiency the duty cycle of communication is around 1 % and results in a very low power average. In order to safe power a beacon mode is supported. The number of supported devices in a network is high, with a support for up to 216 devices. The power saving concepts can result in a long life time of over a year on typical batteries and make this standard a good choice for scalar WSN applications. ZigBee may use 802.15.4 on the lower layers.
The suitability of 802.15.4 as a base for mm data transfer is limited. The frame length is limited to 127 Byte of payload. A realistic payload is around 80 Byte, when using extended addressing and full security information. The definition of a data frame according to the IEEE 802.15.4 standard is shown in Figure 1. Additional Bytes may be needed in higher network layers. The small packet size causes a lot of fragmentation for transferring big mm data. Through the chapter some solutions and ideas to overcome the transfer shortages of 802.15.4 as well as some more powerful transfer technologies will be presented.

2.2 ZigBee

ZigBee (*ZigBee Alliance Webpage*, 2010) is a suite of high level communication protocols for small, low-power digital radios. It sits on top of the layers of the IEEE 802.15.4 standard. Therefore ZigBee uses the free ISM bands and operates with a radio output power of 1 mW. A range from 10 to 100 m can be achieved. The ZigBee standard provides three types of devices:

- The ZigBee **Coordinator** (ZC) is the most powerful device, which maintains and coordinates the network with overall network knowledge.
- The ZigBee **Router** (ZR) works as a router in the network by passing on data.
- The ZigBee **End Device** (ZED) only has limited functionalities to safe cost and complexity. It just reports to his parent device node.

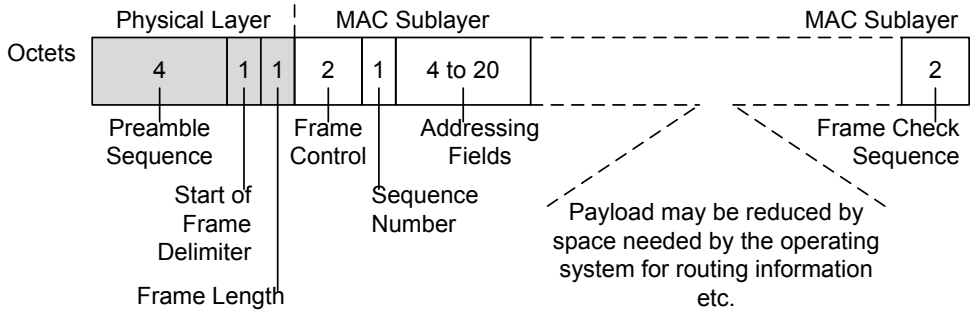


Fig. 1. An IEEE 802.15.4 standard data frame. The full frame can reach a maximum of 127 Byte.

Figure 2 gives an overview of the topology of a ZigBee network. ZigBee provides two network modes: a non-beacon CSMA mode and a beacon-enabled mode with Guaranteed Time Slots.

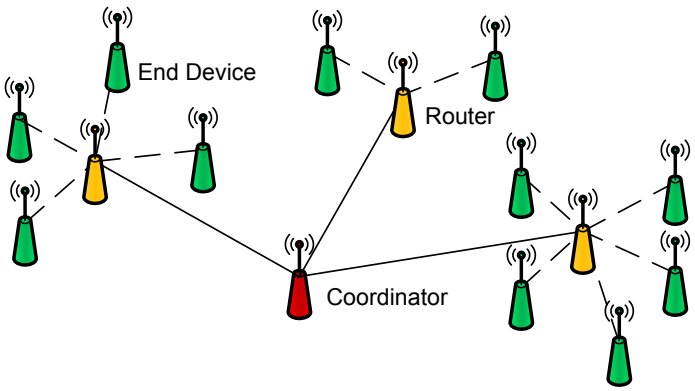


Fig. 2. ZigBee network topology. End devices with limited functionalities send their data to Routers that forward the data to the Coordinator, the device that finally maintains the network structure.

2.3 Bluetooth/IEEE 802.15.1

Bluetooth (*Bluetooth - How it Works*, 2010) is designed to be a low-cost, medium-power, robust, short-range communication protocol for wireless links to replace cables (RS-232) for mobile phones and computers. It was initially published by Ericsson and is now managed by the Bluetooth Special Interest Group (SIG). Bluetooth also uses the worldwide free 2.4 GHz ISM band. It covers in comparison to the 802.15.4 standard a whole product including radio-frequency transceiver, baseband and protocol stack. So it might be comparable to ZigBee running on 802.15.4. An attempt to give a comparative overview of 802.15.4 + ZigBee and

Bluetooth is made in section 2.6. The transfer range of Bluetooth differs with the used Bluetooth class, it is: 1, 10 or up to 100 m. Table 2 shows the details for the different Bluetooth classes.

Bluetooth Class	*Free-Space Range (m)*	*Maximum Output Power (mW)*
1	100	100
2	10	2.5
3	1	1

Table 2. Bluetooth classes.

Bluetooth allows data rates of 1 Mb/s in version 1.2 and up to 3 Mb/s in Version 2.0 with Enhanced Data Rate (EDR). In July 2010 Bluetooth Version 4.0 was formally adopted. This new version supports Bluetooth Low Energy (see section 2.4), formerly known as WiBree, and a High Speed specification. The Bluetooth radio is designed for busy environments with lots of users. The network topology consists of clusters. Up to eight devices join a Piconet. One device is the master of a Piconet, the others are slave devices. Piconets can connect together as Scatternets. The seven slaves of a Piconet are in active communication with the master. Up to another 248 (= 256 - 8) slaves can work passively, while listing for the synchronization with the master, but they can become active at any time. This network topology is shown in Figure 3.

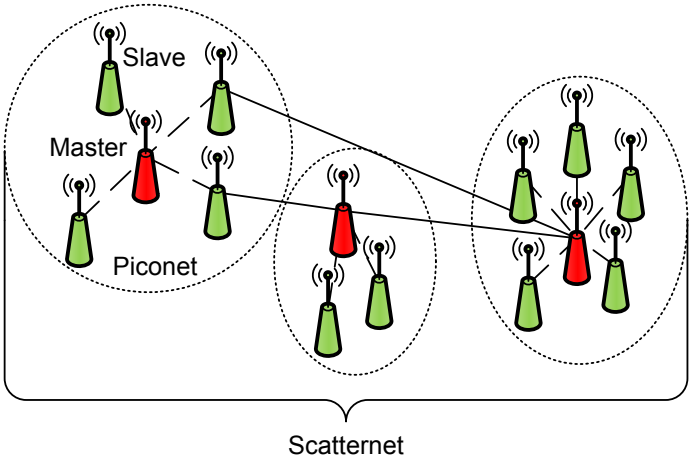


Fig. 3. Bluetooth network topology. A Piconet consists of one master and up to seven slaves. A Piconet master can be a slave in another Piconet. All connected Piconets form a Scatternet.

Bluetooth uses 79 different 1 MHz wide channels and can avoid interference with other ISM devices (either 802.11, 802.15.4 or other Bluetooth devices) by using Frequency Hopping Spread Spectrum (FHSS). The carrier switching is controlled by the Piconet master. Bluetooth also provides adaptive power control, Channel Quality Driven Data Rate (CQDDR) and Adaptive Frequency Hopping (AFH). Some nodes use Bluetooth. Bluetooth has the advantage that it can communicate directly with many laptops or smart phones and is a widely accepted standard in industry. Another advantage is the higher data rate, which allows live audio

streaming. A clear disadvantage is the higher energy consumption in comparison to 802.15.4. In the next section Bluetooth Low Energy/WiBree, as a specialized part of the new Bluetooth version, is discussed.

2.4 Bluetooth Low Energy/WiBree

Bluetooth Low Energy (*Sig Introduces Bluetooth Low Energy Wireless Technology, The Next Generation Of Bluetooth Wireless Technology*, 2010), formerly known as WiBree (Hunn, 2006), is designed to work with Bluetooth. It covers scenarios for end devices with very low capabilities or energy resources, so it is suitable for sensor nodes. In contrast to classic Bluetooth it has a lower application throughput and is not capable of streaming voice. The data rate is 1 Mb/s and the packet length ranges from 8 to 27 Byte. Instead of the Scatternet topology it uses a one-to-one or star topology. Over 4 billion devices can be connected by using a 32 bit address space. This new standard widens the spectrum of applications of Bluetooth and creates an overlapping use case with ZigBee.

2.5 Wi-Fi/IEEE 802.11

Some WMSNs avoid data rate problems by using IEEE 802.11 (*IEEE Std 802.11-2007*, 2007). This standard is commonly known as Wi-Fi or Wireless LAN. This technology has a theoretical data rate up to 11 Mb/s (802.11b) or 54 Mb/s (802.11a, g), but is much more power consuming than the already discussed standards. Even more than Bluetooth, this standard has the advantage that it is widely spread in today's usage and therefore nodes can be included into existing networks. Beside these advantages, IEEE 802.11 is quite improper for small wireless nodes because of its high energy consumption, the complex network stack and expensive hardware units. The usage requires an embedded computer and seems therefore improper for the classical idea of small, low-cost and battery-driven nodes.

2.6 Comparison of ZigBee, Bluetooth and Wi-Fi

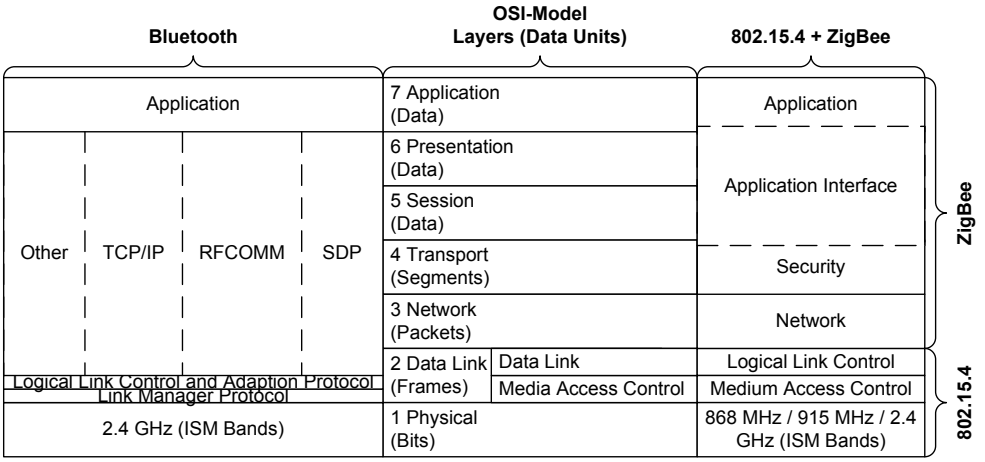


Fig. 4. Comparison of ZigBee and Bluetooth layers based on the OSI-Reference-Model-Layers.

IEEE 802.15.4 + ZigBee, Bluetooth and Wi-Fi are the most frequently used communication technologies for WSNs. Because of their acceptance and the widely available hardware a

short summary and use cases for them are given in the following section. For more comparisons see also (Sidhu et al., 2007). ZigBee is meant to target scalar sensors and the remote control market with very low power consumption and very little communication. ZigBee does not allow streaming of any mm data. Bluetooth allows interoperability and the replacement of cables and targets on wireless USB, hand- and headsets, so that audio-streaming is supported. Figure 4 shows a comparison of ZigBee and Bluetooth based on the well-known OSI-Reference-Model-Layers. Wi-Fi is designed for computer networks and allows high data rates, but it needs a lot of energy and is quite expensive in hardware costs. Wi-Fi allows even video-streaming in high quality. However, even scalar nodes, such as the Tag4M (Ghercioiu, 2010), (Folea & Ghercioiu, 2010), (Ursutiu et al., 2010), use Wi-Fi because of its wide availability and good integration into the Internet. Table 3 shows all technical details in a comparison of the presented technologies.

2.7 Summary

Technology	*Theoretical Data Rate (Mb/s)*	*Output Power (mW)*	*Free-Space Range (m)*	*Frequency Band (GHz)*
IEEE 802.15.4	0.25	1	100	0.868, 0.915, 2.4
Bluetooth	1 – 2	100	100	2.4
IEEE 802.11a	54	40 – 800	120	5
IEEE 802.11b	11	200	140	2.4
IEEE 802.11g	54	65	140	2.4

Table 3. Survey of common transfer technologies. Properties are the theoretical values defined by the standard.

The low bandwidth of the nodes is a problem for streaming media in the network. Live uncompressed video streaming with meaningful resolutions is often impossible. All given transfer rates are the theoretical maximum of the different standards. The real transfer rates will be much slower because of necessary calculations for sending, wrapping to layers and interference on the communication channel.
A single-hop communication between a SunSPOT sensor node and the SunSPOT base station can be mentioned as a real world example. These nodes use a proprietary protocol based on 802.15.4, they have a 180 MHz CPU and can be programmed in Java. Figure 6(a) shows an image of the node and Table 4 provides the basic properties of the SunSPOT. For more information about these nodes see (Sun, 2007) and (*Sun SPOT World*, 2010). The SunSPOTs have, by using the Java objects for easy communication programming, a throughput on the application layer (goodput) of approximately 3 kB/s for big amounts of automatic fragmented data, as an array that is typically used for mm data. The underlying layers provide encryption and security mechanisms, so that the available throughput is small. This example shows that the overhead of underlying layers is big compared to theoretical data rates. Wireless communication can be also jammed and interfered, which decrease the achieved data rate in the real world. More problems will come up in a multi-hop network. To sum up the different transfer technologies, Table 3 gives an overview about the different standards.

Name	*Sun Small Programable Object Technology (SunSPOT)*
Manufacturer	Sun/Oracle
Main Processor	180 MHz 32-bit Atmel ARM920T
Random Access Memory (RAM)	512 kB
Flash Memory (to store programs)	4 MB
Radio Chip	Chipcon/Texas Instruments CC2420
Operating System	Squawk Java Virtual Machine (JVM) on the bare metal
Programming Language	Java ME Squawk implementation

Table 4. Technical preferences of a Java SunSPOT sensor node. Data taken from (Sun, 2007).

3. Multimedia in Wireless Sensor Networks

The following section presents applications offered by WMSNs. Then sensor nodes and basic platforms are described. Systems and architectures are discussed afterwards.

3.1 Applications

Mm **surveillance sensor networks** can be used for monitoring public places and events, private properties, borders or battlefields. One of the first wireless sensor networks was designed in 1967 by the US army to monitor troop movements in the Vietnam War. The so called Igloo White system consists of air-dropped sensors made of analog technology. Acoustic and seismic data was sent by a radio and received by special aircrafts. Around 20,000 sensors were deployed (Correll, 2004). Military target classification is still a wide research topic today. In (Malhotra et al., 2008) target tracking and classification is done by acoustics. The sounds of moving ground vehicles are recorded by mm nodes. The network is able to classify the vehicles with the help of a distributed k-nearest neighbor classification method. Another application is the combination of a WSN with cameras for surveillance of roads or paths (He et al., 2004).

For civil use a parking space finder was developed, which is intended to provide the service of locating available parking spaces near a desired destination. A set of cameras detects the presence of cars in spaces and updates a distributed database, so that a navigation system for finding available spaces can be realized (Campbell et al., 2005). The paper of (Ardizzone et al., 2005) describes the work to design and deploy a system for the surveillance and monitoring of an archaeological site, the "Valley of the Temples" in Agrigento, Italy. The archaeological site must be monitored to be protected. Wireless sensors have advantages because of the size of the area and they are less intrusive than wires which would have to run all across the site. Ardizzone et al. developed an architecture for the surveillance of the site and for monitoring the visitors' behavior.

WMSNs can be used for **habitat monitoring** and **environmental research**. Hu et al. developed a wireless acoustic sensor network for the automatic recognition of animal vocalizations to census the populations of native frogs and an invasive introduced species (Cane Toads) in the monsoonal woodlands of northern Australia (Hu et al., 2005). WMSNs are also able to classify birds by their voices (Wang, Elson, Girod, Estrin & Yao, 2003), (Wang, Estrin & Girod, 2003). Mainwaring et al. deployed a sensor network at James San Jacinto Mountains Reserve (*James San Jacinto Mountains Reserve website*, 2010) for long-term environmental observation. A coastal imagining application was developed by Campbell et al. in collaboration

with oceanographers of the Argus project (*The Coastal Imaging Lab Web*, 2010) on base of IrisNet (Campbell et al., 2005).
Wireless sensors with mm capabilities can be used in **industrial environments**. 42 nodes were deployed in a coal mine to improve security and rescue operations in case of an emergency. The used WMSN provides real-time voice streaming (Mangharam et al., 2006).
An emerging area for all kinds of sensors is **elderly care** and **elderly support** by **home automation**. The Aware Home is a combination of many heterogeneous WSNs (Kidd et al., 1999). For example there is a vision-based sensor to track multiple individuals in an environment based on the system presented in (Stillman et al., 1998). The usage of the combination of audio and image, which are also the main information sources for human perception, are presented in (Silva, 2008). Silva presents the possibilities of smart sensing using a multitude of sensors such as audio and visual sensors in order to detect human movements. This can be applied in home care and home security in a smart environment. The combination of audio and video sensors increases the variety of different detectable events. A prototype implementation to detect events like falling, walking, standing, shouting etc. was presented. In (Meyer & Rakotonirainy, 2003) requirements for sensor networks to enhance the quality of life for people at home are shown. Meyer and Rakotonirainy give an overview of using sensors for different tasks in everyday's home life. Mm sensors can help to solve a lot of tasks like tracking persons, interaction via gestures and speech recognition for house automation and so on.
The key to acceptance of sensor networks at private homes is to provide an improved and safe environment for the individual. The paper of (Mynatt et al., 2000) shows the support of elderly people by a monitored home. Image cameras are used to identify some scenarios, like the immobility of a person either due to a fall or a collapse and they monitor dangerous situations in a household. WMSNs can deliver novel technology for new medical equipment. The publication of (Itoh et al., 2006) presents a one-chip camera for capsule endoscopes. A pill-sized prototype supports a resolution of 320 $\times$ 240 pixels with the help of a 0.25 µm Complementary Metal–Oxide–Semiconductor (CMOS) image sensor. Pill-sized wireless sensors like this could revolutionize medical treatments in many areas and improve diagnosis for illnesses.
Another big field of application will be **education** and **entertainment**. Srivastava et al. have developed a WMSN to be used in early childhood education. The system of software, wireless sensor-enhanced toys and classroom objects is called "Smart Kindergarten" (Srivastava et al., 2001).

3.2 Sensor Nodes with Multimedia Capabilities

WMSNs have high demands on the hardware of the nodes. In the following section nodes and sensor boards, which address these demands, are presented. The range of processors currently used in nodes starts at simple 8 bit processors and ends at embedded computer systems. In small low-power nodes as the MEMSIC's Iris Mote (MEM, 2010c) an ATMEL ATmega1281 (Atm, 2007) microprocessor is used. The MEMSIC's TelosB Mote (MEM, 2010d) uses a Texas Instruments' MSP430 (Tex, 2010) processors. On the high performance side, nodes as the MEMSIC's Imote2 (MEM, 2010a) are built on an Intel/Marvell XSCALE PXA271 processor (Int, 2005). This processor is also used in handhelds and portable media centres and supports "Single Instruction, Multiple Data" (SIMD) extensions such as "Multi Media Extension" (MMX) and "Streaming SIMD Extension" (SSE). These extensions allow the usage of a mathematical operation on more than one value at a time. This kind of vector operations is a major advantage in working with mm data. Filter and other operations on mm data can be boosted

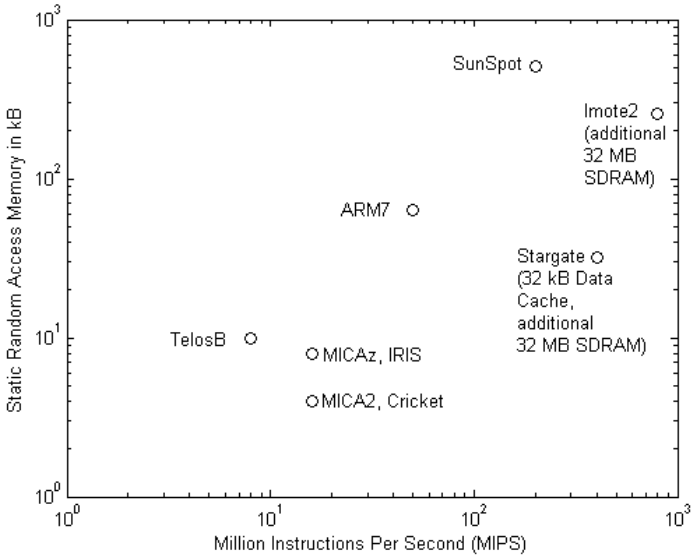


Fig. 5. Plot of processor performance and memory of different nodes. The performance can differ on the clocking of the processors. MIPS values are given by producers/distributors. RAM amount can differ if memory is not onboard, access speed may also differ.

with using these extensions. Even embedded computers, e.g. the discontinued Crossbow's Stargate Platform (Cro, 2007), can be used as sensor nodes.
An overview of the performance of the nodes is given in Figure 5.

3.2.1 Cyclops

The Cyclops imaging platform was a collaboration project between Agilent Technology Inc. and the University of California. Cyclops is a board for low-resolution imaging that can be connected to a host node such as Crossbow's MICA2 or MICAz. It also provides software libraries for image processing on the node. Although it found interest in the research community this project was not a success. As of January 2008 Cyclops is no longer supported by Agilent (Rahimi & Baer, 2005), (Rahimi et al., 2005). The Cyclops board with an attached MICA2 node is shown in Figure 6(b).

3.2.2 ARM7 Based Wireless Image Sensor

Downes et al. present the design of a node for distributed image sensing. The node is based on a 48 MHz 32-bit ARM7 microcontroller with 64 kB of memory on the chip. The communication is based on the IEEE 802.15.4 standard. The image acquisition provides interfaces for two Common Intermediate Format (CIF) resolution (352 $\times$ 288 pixels) sensors and four low resolution (30 $\times$ 30 pixels) sensors. So up to six different image sensors can be connected to one node (Downes et al., 2006).

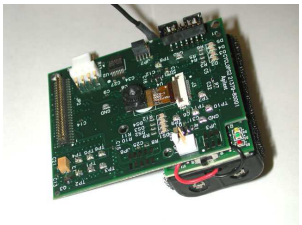
3.2.3 Wireless Smart Camera

A so called Wireless Smart Camera (WiCa) is presented in (Kleihorst et al., 2007). It is a sensor node based on an 8051 microcontroller and ZigBee, and thereby IEEE 802.15.4 compatible, transfer module. It has two cameras and provides the direct storage of two images of a resolution of 256 $\times$ 256 pixels. The term "Smart Camera" is used in the field of computer vision for cameras with integrated image processing capabilities. In (Belbachir, 2010) "a smart camera is defined as a vision system which, in addition to image capture circuitry, is capable of extracting application-specific information from the captured images, along with generating event descriptions or making decisions that are used in an intelligent and automated system."

3.2.4 Stargate Board with Webcam

Stargate is a processing platform for WSNs which can be used itself as a sensor node. It was developed by Intel Research and was sold by Crossbow (Cro, 2007). This platform is often chosen for video sensor networks. The Stargate board is connected to a webcam. This node provides medium-resolution imaging. Since low-power radios are limited, live streaming of video is only possible with Wi-Fi, the Stargate board has no wireless interface at all, but it can be connected to a sensor node or a Wi-Fi card. Normally embedded Linux is used as operating system. The processor is a 400 MHz Intel PXA255 model. Feng et al. present a comparison of the Panoptes video sensors: one based on Strong ARM PDA and the other based on the Crossbow Stargate platform (Feng et al., 2005). The Stargate board with an attached webcam is shown in Figure 6(c).

(a) Java SunSPOT sensor node (*Sun SPOT World*, 2010).



(b) Cyclops with an attached MICA2 node (Rahimi et al., 2005).

(c) The Crossbow Stargate platform with an attached webcam (Feng et al., 2005).

Fig. 6. Images of sensor nodes.

3.2.5 MeshEye

MeshEye is a vision system with two layers. It consists of a low resolution stereo vision system to determine position, range and size of moving objects and a high resolution color camera for further image processing. The system is ARM7-based and is used for real-time object detection. An IEEE 802.15.4 compatible transfer module is provided for interconnection. A power model is also presented to estimate battery lifetime for the node (Hengstler et al., 2007).

3.2.6 CMUcam

CMUcam3 is an open source programmable embedded color vision platform. The CMUcam3 is developed at the Robotics Institute at Carnegie Mellon University and is the latest of a series of embedded cameras. It is based on an ARM7 processor and includes an Omnivision CMOS camera sensor module. CMUcam3 supports CIF resolution with a RGB color sensor and can do some basic image processing on its own processor. Open source libraries and example programs are provided to develop C programs for the camera. There is the possibility to connect it to wireless sensor nodes like the Tmote Sky and FireFly (Car, 2007).

3.2.7 Imote 2 with Multimedia Sensor Board (IMB400)

The Imote multimedia board is a new sensor board for the Imote 2 sensor node. It includes Passive InfraRed sensor (PIR), color image and video camera for image processing, microphone, line input, miniature speaker as well as line output for audio processing. The Imote 2 is considered to be a high-performance sensor with many different power modes and can be clocked up to 416 MHz. The Imote 2 processor even supports MMX and SSE integer instructions, so it is suitable for mm operations. While there is a special version of the Imote 2 for development with the .net microframework, the mm board is not yet supported by the .net microframework, but it is expected to be supported in future. The board is quite recent, so there are no publications or projects available yet (MEM, 2010a), (MEM, 2010b).

3.3 Sensor Networks with Multimedia Support

After introducing some nodes the following section gives an overview about WMSNs. The focus is on the architecture and the design of the whole system.

3.3.1 Meerkats

Meerkats is a wireless network of camera nodes for monitoring and surveillance of wide areas. On the hardware side it is based on the Crossbow Stargate platform. The whole architecture includes a number of techniques for acquiring and processing data from image sensors on the application level. These include acquisition policies, visual analysis for event detection, parameter estimation and hierarchical representation. The architecture also covers resource management strategies that level power consumption versus application requirements (Boice et al., 2004), (Margi et al., 2006).

3.3.2 SensEye: A Multi-tier Camera Sensor Network

SensEye is a multi-tier network of heterogeneous wireless nodes and cameras. It consists of three different camera sensors. There are Cyclops nodes for the lowest layer, ordinary webcams for the middle layer, and pan-tilt-zoom (PTZ) cameras for the highest layer. Details of the different layers are shown in Table 5. The system fulfils three tasks: object detection, recognition and tracking (Kulkarni et al., 2005).

Camera	*Power (mW)*	*Cost ($)*	*Resoultion*	*Features*
Cyclops	33	unpriced	128×128	10 fps, fixed-angle
Webcam	600	75	640×480	30 fps, auto-focus
PTZ camera	1,000	1,000	1024×768	30 fps, retargetable pan-tilt-zoom

Table 5. Different camera sensors of the SensEye-architecture and their characteristics. (Kulkarni et al., 2005)

3.3.3 IrisNet

IrisNet is an Internet-scale architecture for mm sensors. It provides a software framework to connect webcams worldwide via the Internet. The pictures are taken by a Logitech QuickCam Pro 3000 with 640 $\times$ 480 pixels. IrisNet stores the sensor readings in a distributed XML database infrastructure. IrisNet provides a number of mm processing primitives that guarantee the effective processing of data in-network and at-sensor (Campbell et al., 2005).

3.3.4 Explorebots

Dahlberg et al. present the Explorebot, a wireless robot built around the MICA2 node. The low-cost Explorebots can be used as a mobile network experimentation testbed. The robot is equipped with sonic sensors, bumper switches and a magnetic 2-axis compass. Additionally it uses a X10 Cam2 with a resolution of 320 $\times$ 240 pixels, which communicates over its own proprietary wireless transmitter with 15 fps (Dahlberg et al., 2005).

3.3.5 Mobile Emulab

Johnson et al. have developed a robotic wireless and sensor network testbed. While simulation is the dominant research methodology in wireless and sensor networking, there are few real world testbeds. Even fewer testbeds exist for WSNs with mobile nodes. In order to overcome this weakness and to allow more and cheaper experiments in real world environments the Emulab testbed was created. This testbed provides software, which allows remote access. Robots carry sensor nodes and single board computers through a fixed indoor field of sensor-equipped nodes, of which all of them are running the user's selected software. In real-time, interactively or driven by a script, remote users can place the robots, control all the computers and network interfaces, run arbitrary programs, and log data. Webcams are used to supervise the experiments by remote control. The Hitachi KP-D20A cams have a resolution of 768 $\times$ 494 pixels and provide a vision-based tracking system accurate to 1 cm (Johnson et al., 2006).

3.3.6 iMouse

The iMouse system consists of static sensor nodes that sense scalar data and mobile sensor nodes for taking images of the detected events. The system is shown in Figure 7. The mobile nodes are based on a Crossbow Stargate processing board connected to a node for IEEE 802.15.4 communication, an 802.11 WLAN card, a webcam and a Lego-based car to provide mobility. This connection of a mobile sensor with a classical static WSN can provide advanced services at lower cost than traditional surveillance systems (Tseng et al., 2007).

3.3.7 PlantCare

Robots can deliver new services in a WSN. LaMarca et al. used a robot in a WSN to take care of houseplants in an office. The used nodes are UC Berkeley motes, commercially available under the MICA brand, running TinyOS. The robot is based on the Pioneer 2-DX platform and uses a laser scanner for orientation. The robot has a human calibrated sensor board equal to the static nodes, so the robot improves calibration of the distributed nodes (LaMarca et al., 2002). Robot and sensors are shown in Figure 8.

3.4 Summary

In this section WMSN applications, their hardware as well as their system architecture have been reviewed. Table 6 summarizes the presented applications. Even if the "killer application" of WMSNs is still missing, they have already started influencing classical WSNs and the

Name	*Basic Devices*	*Network-Size (real deployment)*	*Communication*	*Sensors*	*Mobility*
Meerkats	Stargate	not mentioned	802.11b	Logitech Quick Cam Pro 4000	no
SensEye	MICA2	not mentioned	802.15.4	CMUcam	no
	Stargate	not mentioned	802.15.4, 802.11	Webcam	no
	Embedded Computer	not mentioned	Ip-based	Sony SNC-RZ30N Pan-Tilt-Zoom camera	no
IrisNet	PC	$\approx$ 500	Internet	Logitech Quick Cam Pro	no
Explorebots	MICA2	< 10	802.15.4	Sonic sensors, bumper switches, magnetic 2–axis compass, X10 Cam2	yes
Mobile Emulab	Stargate board connected to MICA2	6	802.11b, 802.15.4	Infrared proximity sensors	yes
	MICA2	25	802.15.4	not mentioned	no
	Webcam	6	not wireless	Hitachi KP-D20A	no
iMouse	MICAz	17	802.15.4	light intensity sensor	no
	Stargate	2	802.15.4, 802.11	light detector, infrared receiver	yes
PlantCare	MICA	not mentioned	802.15.4	photo resistor (light level), thermistor (temperature), irrometer (soil moisture), power charge	no
	Pioneer 2-DX	not mentioned	802.11 b	Human-calibrated sensor node (see row above), laser scanner	yes

Table 6. Overview of selected WMSN architectures. For heterogeneous architectures every stage is presented. The properties are based on publications mentioned in this section.

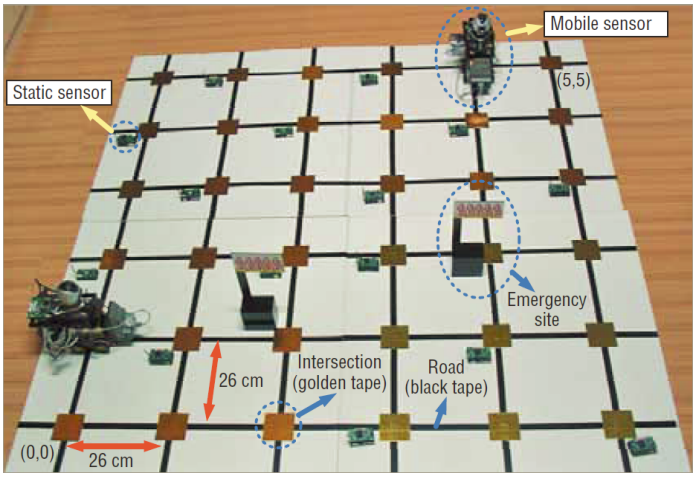
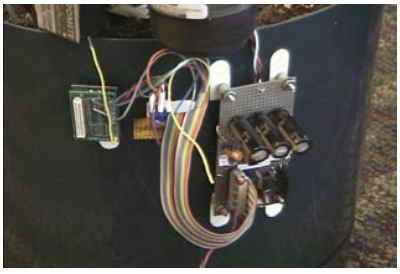


Fig. 7. The iMouse testbed. Static sensors and Lego-based robots on an 6×6 grid-like sensing field (Tseng et al., 2007).

Internet. Their impact has gone beyond their original use cases for military applications. A very important fact for WSNs in general, but even more urgent for WMSNs, is data security and privacy. The picture of a human face or the recording of a voice are very personal and can be dedicated to a person via software. Most of the discussed publications and this chapter have not accomplished further research on security and privacy issues. Nevertheless, first prototype nodes and systems have been designed and deployed for research purposes. In the next section, conclusions are drawn from the existing deployments, which will be classified into patterns of system architectures.

4. Architectures of Wireless Multimedia Sensor Networks

The basic architecture for a WSN, which senses scalar values, is a flat homogeneous network of equal sensor nodes reporting to a single base station. This concept is very limited and even scalar WSNs have been designed in different ways. For demanding WMSNs there has



(a) PlantCare sensor (LaMarca et al., 2002).

(b) PlantCare robot (LaMarca et al., 2002).

Fig. 8. Images of the PlantCare sensor network.

not been found a reference architecture yet, but most systems can be grouped into one of the following four architectures.

4.1 Homogeneous Networks of Multimedia Sensor Nodes

This type of network uses the classical WSN technology presented in section 3.2. However the IEEE 802.15.4 standard is designed for very low-power, delay tolerant and slow networks with a very small duty cycle and the theoretical data rate is just 250 kb/s. This is not usable for fluent image transfers. An uncompressed 640 $\times$ 480 pixel black-white image would for instance be transferred in over one second under the best theoretically possible conditions. Multi-hopping, interference and network traffic make this impossible for a real application, as it is shown in the SunSPOT example in section 2.7.

A solution would be to transfer less data. In order to achieve this, the requirements on the data collection have to be checked. In many applications the data analysis result is important and not the data itself. So reducing the amount of data can sometimes already be achieved while monitoring.

Zheng et al. present the approach of using line scan cameras instead of two-dimensional cameras (Zheng & Sinha, 2007). In comparison to other image processing methods, this concept is less computationally intensive. They sum up the capabilities of the sensors in data processing, compression, and streaming in WSNs. They focus on several unsolved issues such as sensor setting, shape analysis, robust object extraction, and real-time background adapting to ensure long-term sensing and visual data collection via networks. All the developed algorithms are executed in constant complexity, which reduces the sensor and network burden. The latter algorithms can for example be applied in traffic monitoring. Another usage of line cameras in WSNs is shown in (Chitnis et al., 2009).

Computation is less power consuming than sending data via the radio. The restrictions of a weak processing unit and a short battery capacity produce a need to further investigate algorithms. These are either algorithms with small complexity running on a single node or distributed algorithms running in the network.

Culurciello et al. present a low complex compression algorithm for videos based on pixel-change-events, which can run on today's nodes' hardware (Culurciello et al., 2007). Besides its low computational costs this algorithm compresses a 320 $\times$ 240 pixel video to the point where it can be transferred by nodes with over 10 fps. The idea of Address Event Image Sensors presented in (Teixeira et al., 2006) is biologically inspired and keeps the privacy of monitored people. Therefore it is suitable for monitoring of elderly people at home or other privacy-sensitive applications.

An example for a distributed algorithm is given in (Oeztarak et al., 2007). They present a framework for mm processing in WSNs and consider the needs of surveillance video applications. This framework automatically extracts moving objects, treats them as intruder events and exploits their positions for efficient communication. Then a joint processing of collected data at the base station is applied to identify events using fuzzy (multi-valued logic) memberships and to request the transfer of real image data from the sensors to the base station.

4.2 Heterogeneous Networks of Scalar Sensor Nodes Connected to Multimedia Sensor Nodes

As shown in the previous sections and based on the bandwidth problems that occur, not many existing WMSNs rely on sensor nodes with mm capabilities. A common design is the combination of a scalar WSN with a second network, which is triggered, to measure mm data. This architecture tries to overcome the restrictions of classical WSNs by the usage of computer

networks. The mm network is mostly an Internet protocol-based computer network using the IEEE 802.11 standard. This architecture is quite easy to realize and is widely used as shown by the amount of applications using this architecture in section 3.3. The disadvantages of using a personal computer or even an embedded computer instead of a microcontroller are big size, high power consumption and high costs.

4.3 Wireless Sensor Networks with Mobile Nodes

Another concept to collect more information in a WSN is the usage of mobile nodes, as presented in section 3.3.4, 3.3.5, 3.3.6 and 3.3.7. While static nodes are mostly low-power, unreliable and cheap, the mobile node or robot can be equipped with high-class sensors, which make more detailed measurements and take pictures or videos. Beyond this, a robot can accomplish a whole new class of missions, like node replacement, deployment, recharging and redeployment or hole recovery (Sheu et al., 2005), (LaMarca et al., 2002). The architecture can still vary between one network or two connected networks and the control of the robot can be done via a server or it can be decentralized. With the usage of mobility new problems arise as the localization of the robot, the creation of a map and the navigation through the WSN, which are just some new challenges. As far as the authors know, none of the mobile nodes has been used in real-life environments yet.

4.4 Wireless Sensor Networks without Base Station/Instrumentation Cloud

Recently, sensor nodes have been connected directly to the Internet. When the nodes are computers as in (Campbell et al., 2005), a direct Internet connection is easy. In the trend of Cloud Computing some WSNs deny the need of a base station. Ghercioiu (Ghercioiu, 2010) presents the word "Instrumentation Cloud". In this architecture sensors send their results directly to the Internet. The results will be available to every device with a standard browser and Internet connection. Everything, apart from the physical Input/Output, will take place on the web (Ursutiu et al., 2010), (*Tag4M Cloud Instrumentation*, 2010). If security is a major concern, a closed system should be used alternatively. Hereby, the advantage is that the data is not leaving the private network. Thus, automation and security monitoring are no suitable applications for the Instrumentation Cloud.

4.5 Summary

Figure 9 gives an illustrative summary of the discussed architectures for WMSNs without mobility. The design concepts of WMSNs are still developing. Even if there is no widely used reference pattern yet, the authors believe that publishing the data on the Internet is a key point to success. And as a learned lesson from the Internet as the network of networks, homogeneous network architectures seem to be not flexible enough to stand the challenges of the future. Internet Protocol Version 6 (IPv6) has the potential to be used in WSNs. IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) as part of the new protocol standard will clear the way for an enormous amount of nodes to be directly addressable worldwide (*IPv6.com - The Source for IPv6 Information, Training, Consulting & Hardware*, 2010), (Hui & Culler, 2008). So it will be probably possible to search the Internet for live sensor data in the near future. The technological bases are already developed and since search providers (e.g. Google) search real-time web-applications (e.g. Twitter), this vision is not far away. Internet-based WSN real-time data storage is already available today (*pachube - connection environments, patching the planet*, 2010).

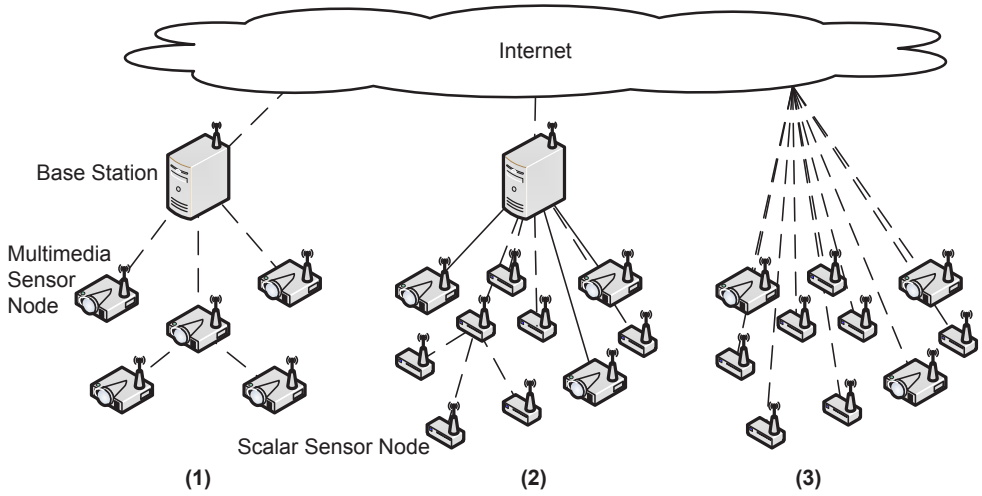


Fig. 9. Three of the most common architectures for Wireless Multimedia Sensor Networks without mobility. The illustrations assume that the sensor data will be uploaded on the Internet. (1) Homogeneous network of multimedia sensor nodes (2) Heterogeneous network of scalar sensor nodes connected to multimedia sensor nodes (3) Instrumentation Cloud

5. Conclusion and Outlook

This chapter reviewed available transfer technologies and hardware for WMSNs. Applications were presented and their architectures have been discussed. The advantages and disadvantages for each of the architectures have been shown. At the moment there are many fast evolving standards and new technologies for WSNs. Mm support is still a minority requirement but has grown in the last few years. Mobile nodes will become a source of information: not only in the form of robots but also as devices that can be carried around by humans. Even today's mobile phones are full of sensors and will be part of tomorrow's WSNs. Other sources of data will be the sensors built in cars or digital Internet-connected meters sensing the electricity, gas and water consumption of a household. These new meter devices are called "Smart Meters" and the vision of a network of many households is named "Smart Grid". All in all, an increasing number of devices will be active on the Internet without direct assistance of humans. New quantities of information will be available and will allow the development of new knowledge. This widening of the possibilities of the Internet will lead to the new version of the Internet, referred as the "Internet of Things".

The connection of actuators in WSNs will also become more important in the next few years. With more reliable WSNs and event recognition algorithms WSNs will become integrated into automation applications. Wireless technologies, as WirelessHART (*HART Communication Protocol - Wireless HART Technology*, 2010) or ISA100.11a (*ISA-100 Wireless Compliance Institute*, 2010), will be used more and more in industry in the next years. Image processing is an important part of today's process for quality controlling, so the authors expect wireless image processing nodes to be part of new WMSNs for automation (Melodia, 2007).

All these new emerging developments create new research challenges. As the authors believe, research is not only needed in the direct realization in terms of hardware or basic transfer

technologies, but also in security and privacy. Maintenance, like wireless update delivery, coexistence of networks as well as the redelivery, recycling and disposing of sensor nodes, will become an important topic of future research. Middleware for the connection of all the novel networks will be also needed. Finally new operating systems, programming models and patterns will be created for efficient usage of the WSNs.

6. Acknowledgement

The authors wish to thank the following for their financial support: the Embark Initiative and Intel, who fund this research through the Irish Research Council for Science, Engineering and Technology (IRCSET) postgraduate Research Scholarship Scheme.

7. References

Akyildiz, I. F., Melodia, T. & Chowdhury, K. R. (2007a). A survey on wireless multimedia sensor networks, *Computer Networks* **51**(4): 921 – 960.

Akyildiz, I. F., Melodia, T. & Chowdhury, K. R. (2007b). Wireless multimedia sensor networks: A survey, *IEEE Wireless Communications Magazine* pp. 32 – 39.

Akyildiz, I. F., Melodia, T. & Chowdhury, K. R. (2008). Wireless multimedia sensor networks: Applications and testbeds, *Proceedings of the IEEE*, Vol. 96, pp. 1588 – 1605.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey, *Computer Networks* **38**(4): 393 – 422.

Arampatzis, T., Lygeros, J. & Manesis, S. (2005). A survey of applications of wireless sensors and wireless sensor networks, *Proceedings of the 2005 IEEE International Symposium on Intelligent Control, Mediterrean Conference on Control and Automation*, pp. 719 – 724.

Ardizzone, E., La Cascia, M., Re, G. L. & Ortolani, M. (2005). An integrated architecture for surveillance and monitoring in an archaeological site, *VSSN '05: Proceedings of the third ACM international workshop on video surveillance & sensor networks*, ACM, New York, NY, USA, pp. 79 – 86.

Atm (2007). *8-bit Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash ATmega640/V ATmega1280/V ATmega1281/V ATmega2560/V ATmega2561/V Preliminary Summary*.

Belbachir, A. N. (2010). *Smart Cameras*, Springer USA.

Bluetooth - How it Works (2010). `http://www.bluetooth.com/English/Technology/Works/Pages/default.aspx`.

Boice, J., Lu, X., Margi, C., Stanek, G., Zhang, G., Manduchi, R. & Obraczka, K. (2004). Meerkats: A power-aware, self-managing wireless camera network for wide area monitoring, *Technical report*, Department of Computer Engineering, University of California, Santa Cruz.

Campbell, J., Gibbons, P. B., Nath, S., Pillai, P., Seshan, S. & Sukthankar, R. (2005). Irisnet: an internet-scale architecture for multimedia sensors, *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, ACM, New York, NY, USA, pp. 81 – 88.

Car (2007). *CMUcam3 Datasheet*.

Chitnis, M., Liang, Y., Zheng, J. Y., Pagano, P. & Lipari, G. (2009). Wireless line sensor network for distributed visual surveillance, *PE-WASUN '09: Proceedings of the 6th ACM symposium on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, ACM, New York, NY, USA, pp. 71 – 78.

Correll, J. T. (2004). Igloo white, *Airforce Magazine* **87**: 56 – 61.

Cro (2007). *Stargate X-Scale, Processor Platform Datasheet*.

Culurciello, E., Park, J. H. & Savvides, A. (2007). Address-event video streaming over wireless sensor networks, *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007*, pp. 849 – 852.

Dahlberg, T. A., Nasipuri, A. & Taylor, C. (2005). Explorebots: a mobile network experimentation testbed, *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on experimental approaches to wireless network design and analysis*, ACM, New York, NY, USA, pp. 76 – 81.

Downes, I., Rad, L. B. & Aghajan, H. (2006). Development of a mote for wireless image sensor networks, *Cognitive Systems and Interactive Sensors (COGIS)*, Paris.

Feng, W.-C., Kaiser, E., Feng, W. C. & Baillif, M. L. (2005). Panoptes: scalable low-power video sensor networking technologies, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* **1**(2): 151 – 167.

Folea, S. & Ghercioiu, M. (2010). *Radio Frequency Identification Fundamentals and Applications*, IN-TECH, chapter 17, a Wi-Fi RFID Active Tag Optimized for Sensor Measurements, pp. 287 – 310.

Ghercioiu, M. (2010). A new approach to wireless sensors: The instrumentation cloud, *Control Engineering Europe*.

HART Communication Protocol - Wireless HART Technology (2010). `http://www.hartcomm.org/protocol/wihart/wireless_technology.html`.

He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T., Luo, L., Stoleru, R., Yan, T., Gu, L., Hui, J. & Krogh, B. (2004). Energy-efficient surveillance system using wireless sensor networks, *MobiSys '04: Proceedings of the 2nd international conference on mobile systems, applications, and services*, ACM, New York, NY, USA, pp. 270 – 283.

Hengstler, S., Prashanth, D., Fong, S. & Aghajan, H. (2007). Mesheye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance, *6th International Symposium on Information Processing in Sensor Networks, 2007. IPSN 2007*, pp. 360 – 369.

Hu, W., Tran, V. N., Bulusu, N., Chou, C. T., Jha, S. & Taylor, A. (2005). The design and evaluation of a hybrid sensor network for cane-toad monitoring, *IPSN '05: Proceedings of the 4th international symposium on information processing in sensor networks*, IEEE Press, Piscataway, NJ, USA, p. 71.

Hui, J. W. & Culler, D. E. (2008). Ip is dead, long live ip for wireless sensor networks, *SenSys '08: Proceedings of the 6th ACM conference on embedded network sensor systems*, ACM, New York, NY, USA, pp. 15 – 28.

Hunn, N. (2006). An introduction to wibree, *White paper*, Ezurio Ltd.

IEEE Std 802.11-2007 (2007).

IEEE Std 802.15.4-2003 (2003).

Int (2005). *Intel PXA27x Processor Family Data Sheet*.

IPv6.com - The Source for IPv6 Information, Training, Consulting & Hardware (2010). `http://ipv6.com/articles/sensors/IPv6-Sensor-Networks.htm`.

ISA-100 Wireless Compliance Institute (2010). `http://www.isa100wci.org/`.

Itoh, S., Kawahito, S. & Terakawa, S. (2006). A 2.6mw 2fps qvga cmos one-chip wireless camera with digital image transmission function for capsule endoscopes, *IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006.*

James San Jacinto Mountains Reserve website (2010). `http://www.jamesreserve.edu`.

Johnson, D., Stack, T., Fish, R., Flickingery, D. M., Stoller, L., Ricci, R. & Lepreau, J. (2006). Mobile emulab: A robotic wireless and sensor network testbed, *IEEE INFOCOM*, number 25.

Kidd, C. D., Orr, R. J., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E. & Newstetter, W. (1999). The aware home: A living laboratory for ubiquitous computing research, *Proceedings of the Second International Workshop on Cooperative Buildings*.

Kleihorst, R., Schueler, B. & Danilin, A. (2007). Architecture and applications of wireless smart cameras (networks), *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, Vol. 4, pp. IV–1373 – IV–1376.

Kulkarni, P., Ganesan, D., Shenoy, P. & Lu, Q. (2005). Senseye: a multi-tier camera sensor network, *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, ACM, New York, NY, USA, pp. 229 – 238.

LaMarca, A., Koizumi, D., Lease, M., Sigurdsson, S., Borriello, G., Brunette, W., Sikorski, K. & Fox, D. (2002). Making sensor networks practical with robots, *Technical report*, Intel Research.

Malhotra, B., Nikolaidis, I. & Harms, J. (2008). Distributed classification of acoustic targets in wireless audio-sensor networks, *Computer Networks* **52**(13): 2582 – 2593.

Mangharam, R., Rowe, A., Rajkumar, R. & Suzuki, R. (2006). Voice over sensor networks, *27th IEEE International Real-Time Systems Symposium, 2006. RTSS '06*, pp. 291 – 302.

Margi, C. B., Petkov, V., Obraczka, K. & Manduchi, R. (2006). Characterizing energy consumption in a visual sensor network testbed, *2nd International IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2006)*.

Melodia, T. (2007). *Communication and Coordination in Wireless Multimedia Sensor and Actor Networks*, Dissertation, Georgia Institute of Technology.

MEM (2010a). *Imote2 High-performance wireless sensor network node*. formerly Crossbow.

MEM (2010b). *Imote2 Multimedia - IMB400*. formerly Crossbow.

MEM (2010c). *Iris Wireless Measurement System*. formerly Crossbow.

MEM (2010d). *TelosB Mote Platform Datassheet*. formerly Crossbow.

Meyer, S. & Rakotonirainy, A. (2003). A survey of research on context-aware homes, *ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, Vol. 21, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 159 – 168.

Mynatt, E. D., Essa, I. & Rogers, W. (2000). Increasing the opportunities for aging in place, *CUU '00: Proceedings on the 2000 conference on Universal Usability*, ACM, New York, NY, USA, pp. 65 – 71.

Oeztarak, H., Yazici, A., Aksoy, D. & George, R. (2007). Multimedia processing in wireless sensor networks, *IEEE Innovations* .

pachube - connection environments, patching the planet (2010). `http://www.pachube.com/`.

Pottie, G. J. & Kaiser, W. J. (2000). Wireless integrated network sensors, *Commun. ACM* **43**(5): 51 – 58.

Rahimi, M. & Baer, R. (2005). *Cyclops: Image Sensing and Interpretation in Wireless Sensor Networks, Reference Manual*, Agilent Corporation, University of California.

Rahimi, M., Baer, R., Iroezi, O. I., Garcia, J. C., Warrior, J., Estrin, D. & Srivastava, M. (2005). Cyclops: in situ image sensing and interpretation in wireless sensor networks, *SenSys*

'05: Proceedings of the 3rd international conference on embedded networked sensor systems, ACM, New York, NY, USA, pp. 192 – 204.

Roemer, K. & Mattern, F. (2004). The design space of wireless sensor networks, *Wireless Communications, IEEE* **11**(6): 54 – 61.

Sheu, J.-P., Cheng, P.-W. & Hsieh, K.-Y. (2005). Design and implementation of a smart mobile robot, Vol. 3, pp. 422 – 429.

Sidhu, B., Singh, H. & Chhabra, A. (2007). Emerging wireless standards - wifi, zigbee and wimax, *World Academy of Science, Engineering and Technology*, Vol. 25, pp. 308 – 313.

Sig Introduces Bluetooth Low Energy Wireless Technology, The Next Generation Of Bluetooth Wireless Technology (2010). `http://www.bluetooth.com/English/Press/Pages/PressReleasesDetail.aspx?ID=4`.

Silva, L. C. D. (2008). Audiovisual sensing of human movements for home-care and security in a smart environment, *International Journal On Smart Sensing And Intelligent Systems* **1**(1): 220 – 245.

Srivastava, M., Muntz, R. & Potkonjak, M. (2001). Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments, *MobiCom '01: Proceedings of the 7th annual international conference on mobile computing and networking*, ACM, New York, NY, USA, pp. 132 – 138.

Stillman, S., Tanawongsuwan, R. & Essa, I. (1998). A system for tracking and recognizing multiple people with multiple cameras, *Proceedings of Second International Conference on Audio-Visionbased Person Authentication*, pp. 96 – 101.

Sun (2007). *Sun Small Programmable Object Technology (Sun SPOT) Theory of Operation*.

Sun SPOT World (2010). `http://www.sunspotworld.com/index.html`.

Tag4M Cloud Instrumentation (2010). `http://www.tag4m.com/`.

Teixeira, T., Lymberopoulos, D., Culurciello, E., Aloimonos, Y. & Savvides, A. (2006). A lightweight camera sensor network operating on symbolic information, *The First Workshop on Distributed Smart Cameras (held in conjunction with ACM SenSys)*.

Tex (2010). *MSP430 Ultra-Low-Power Microcontrollers Product Brochure*.

The Coastal Imaging Lab Web (2010). `http://cil-www.oce.orst.edu/`.

Tseng, Y.-C., Wang, Y.-C., Cheng, K.-Y. & Hsieh, Y.-Y. (2007). imouse: An integrated mobile surveillance and wireless sensor system, *IEEE Computer* **40**(6): 60 – 66.

Ursutiu, D., Ghercioiu, M., Cotfas, P. A., Cotfas, D. T., Samoila, C. & Auer, M. (2010). Web instrumts, *IEEE EDUCON*, Madrid.

Wang, H., Elson, J., Girod, L., Estrin, D. & Yao, K. (2003). Target classification and localization in habitat monitoring, *ICASSP*.

Wang, H., Estrin, D. & Girod, L. (2003). Preprocessing in a tiered sensor network for habitat monitoring, *EURASIP J. Appl. Signal Process.* **2003**: 392 – 401.

Zheng, J. Y. & Sinha, S. (2007). Line cameras for monitoring and surveillance sensor networks, *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, ACM, New York, NY, USA, pp. 433 – 442.

ZigBee Alliance Webpage (2010). `http://www.zigbee.de`.

23

Security and Privacy in Wireless Sensor Networks

Arijit Ukil
Innovation Labs, Tata Consultancy Services
Kolkata, India

1. Introduction

Wireless Sensor Network (WSN) consists of mostly tiny, resource-constraint, simple sensor nodes, which communicate wirelessly and form ad hoc networks in order to perform some specific operation. Due to distributed nature of these networks and their deployment in remote areas, these networks are vulnerable to numerous security threats that can adversely affect their proper functioning. Simplicity in WSN with resource constrained nodes makes them very much vulnerable to variety of attacks. The attackers can eavesdrop on its communication channel, inject bits in the channel, replay previously stored packets and much more. An adversary can easily retrieve valuable data from the transmitted packets that are sent (Eavesdropping). That adversary can also simply intercept and modify the packets' content meant for the base station or intermediate nodes (Message Modification), or retransmit the contents of those packets at a later time (Message Replay). Finally, the attacker can send out false data into the network, maybe masquerading as one of the sensors, with the objectives of corrupting the collected sensors' reading or disrupting the internal control data (Message Injection). Securing the WSN needs to make the network support all security properties: confidentiality, integrity, authenticity and availability. Attackers may deploy a few malicious nodes with similar or more hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack. The sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node. As a result, WSN has to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services. These threats can be categorized as follows:

- Common attacks
- Denial of service attack
- Node compromise
- Impersonation attack
- Protocol-specific attacks

The ad hoc or infrastructure less feature brings a great challenge to WSN security as well. For example, the dynamics of the whole network inhibits the idea of pre-distribution of a shared key between the base station and all sensors. Several random key pre-distribution schemes have been proposed in the context of symmetric encryption techniques (Chan, et al. (2003), Liu, et al. (2005)). In the context of applying public-key cryptography techniques in sensor networks, an efficient mechanism for public-key distribution is necessary as well. In the same way that distributed sensor networks must self-organize to support multi-hop routing, they must also self-organize to conduct key management and building trust relation among sensors. If self-organization is lacking in a sensor network, the damage resulting from an attack or even the hazardous environment may be devastating. Since WSN is a wireless service-oriented infrastructure, one of the most problematic attacks that it may face is the Denial of Service (DoS) attack. A DoS attack on WSN may take several forms: node collaboration, in which a set of nodes act maliciously and prevent broadcast messages from reaching certain section(s) of the sensor network; jamming attack, in which an attacker jams the communication channel and avoids any member of the network in the affected area to send or receive any packet; and exhaustion of power, in which an attacker repeatedly requests packets from sensors to deplete their battery life. Newsome et al. describe the Sybil attack as it relates to wireless sensor networks (Newson, et al. 2004). Simply put, the Sybil attack is defined as a "malicious device illegitimately taking on multiple identities" (Newson, et al. (2004)). It was originally described as an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks. In a nutshell, the security vulnerability of a WSN can be listed as:

- Denial of Service (DoS) attacks
- Link layer attacks
- Network layer attacks
- Transport layer attacks
- Link and physical layer attacks

Apart from security concern, privacy preservation in WSN is a big challenge. The explosive growth and advancement of the information age, data collection and data analysis have exploded both in size and complexity. This in turn has impacted on the privacy preservation of the data of individual users or the network itself. Privacy in our context can be defined as the control over access to information about oneself. Privacy is also the limited access to a person or a process and to all the features related to the person. Privacy preservation is important from both individual as well as organizational perspectives. There are three types of privacy threats. If an adversary can determine the meaning of a communication exchange because of the existence of a message and the context of the situation, there is a content privacy threat. If an adversary is able to deduce the identities of the nodes involved in a communication, there is an identity privacy threat. And if the adversary is able to infer the physical location of a communication entity or to approximate the relative distance to that entity, there is a location privacy threat.

In this book chapter, more emphasis will be given to privacy issues. It is understood that good amount of research works are directed (Karlof, et al. (2003), Law, et al. (2006), Gaubatz, et al. (2005) towards solving the problems of WSN security, whereas lesser effort have been put towards mitigating the problems related to WSN privacy. In fact, with the advent of the concept ubiquitous computing (Weiser, et al. (1991)), privacy becomes as important as

security. So, we mainly focus on WSN privacy issues and highlight the WSN security in brief considering the large volume of work has been already done.

2. WSN Security

WSNs provide unique opportunities of interaction between computing devices and their environment. The adhoc nature and wireless vulnerability make WSN a soft target for security attacks. In order to understand the security aspects of WSN, we provide a brief description of the different attacks and then present the possible solutions. First, we find out the requirements of WSN security. Then we present some of the typical attacks on WSN security and lastly we describe some well-known mechanisms for preventing some the attacks.

2.1 WSN requirements

WSN can be considered as a highly distributed database with wireless links. Security goals for distributed databases are very well studied. The data should be accessible only to authorized users (confidentiality), the data should be genuine (integrity), and the data should be always available on the request of an authorized user (availability). All these requirements also apply to WSNs and their users. Data confidentiality is the most important issue in network security. The objective of confidentiality is required in sensors environment to protect information travelling among the sensor nodes of the network or between the sensors and the base station from disclosure. With the implementation of confidentiality, an adversary may be unable to steal information. However, this doesn't mean the data is safe. The adversary can change the data, so as to send the sensor network into disarray. For example, a malicious node may add some fragments or manipulate the data within a packet. This new packet can then be sent to the original receiver. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit. Authentication in sensor networks is essential for each sensor node and base station to have the ability to verify that the data received was really sent by a trusted sender or not. This authentication is needed during the clustering of sensor node in WSN. We can trust the data sent by the nodes in that group after clustering. Integrity controls must be implemented to ensure that information will not be altered in any unexpected way. Many sensor applications such as pollution and healthcare monitoring rely on the integrity of the information to function with accurate outcomes. Secure management is needed at base station, clustered nodes, and protocol layer in WSN. Because security issues like key distribution to sensor nodes in order to establish encryption and routing information need secure management. Even if confidentiality and data integrity are assured, we also need to ensure the freshness of each message. Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. This requirement is especially important when there are shared-key strategies employed in the design. Typically shared keys need to be changed over time. Another important issue is the availability factor of the nodes or the transmission media. The network should remain operational all the time. It must have some redundancy to counter link failures and have the capability to survive against different attacks.

It also needs to be understood that these requirements are to be satisfied under some kinds of limitations. Among them, limitation of device resources (limited energy, memory and computation power), unreliable communication (packet drop, latency, transmission conflicts) and unattended operation (no centralized control) need to be taken care of.

2.2 WSN attacks

WSNs are vulnerable to various types of attacks. These attacks can be broadly categorized as passive and active. Passive attacks do not disrupt the operation of the network. In this case the attacker snoops the data exchanged inside the network without modifying it. Detection of passive attacks is very difficult since the operation does not get affected. Where as in active attacks, data is altered and thus disturbing the normal network activities. In this chapter, we mostly focus on active attacks. It can be noted that attacks on WSNs are not limited to simply denial of service attacks, but rather encompass a variety of techniques including node takeovers, attacks on the routing protocols, and attacks on a node's physical security. We present the typical attacks from the perspective of protocol layers from where they are initiated.

2.2.1 Physical layer attack

Physical layer is responsible transmission of raw data bits. This is mostly involved in modulation, coding, signal detection and data encryption. Broadly two types of attacks are possible. Jamming attack is responsible for disturbing and disrupting the transmission between sender and receiver (Shi, et al. (2004)). In device tempering attack, the sensor device is physically tempered by the attacker to extract or alter the cryptographic keys and other important information (Wang, et al. (2005), Wang, et al. (2004)).

2.2.2 Link layer attack

In link layer, artificial collision creation, resource exhaustion, unfair and unbalanced resource allocation kind of attacks take place (Akyildiz, et al. (2002). In fact, unfairness is a kind of weak DoS attack (Wood (2002). In this scenario, the attacker attempts to degrade the time-critical applications of other nodes by disrupting their frame transmission. Another link-layer threat to WSNs is the denial-of-sleep attack. This attach prevents the node from going into sleep mode (Raymond (2006)).

2.2.3 Network layer attack

Network layer of WSN is vulnerable to various attacks. In wormhole attack, the attacker receives packets at one location in the network and tunnels them to another location inside the network, where the packet is resent into the network (Hu, et al. (2003)). The tunnel between the colluding attacker nodes is referred as wormhole. A particularly harmful attack against sensor networks is known as the Sybil attack, where a node illegitimately claims multiple identities. Newsome et al. describe the Sybil attack as it relates to WSNs. Sybil attack is defined as a "malicious device illegitimately taking on multiple identities" (Douceur, et al. (2002)). It was originally described as an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks. Another well-known attack which produces great amount of harm is traffic-analysis attack.

For example, a rate monitoring attack simply makes use of the idea that nodes closest to the base station tend to forward more packets than those farther away from the base station. An attacker need only monitor which nodes are sending packets and follow those nodes that are sending the most packets. In a time correlation attack, an adversary simply generates events and monitors to whom a node sends its packets (Deng, et al. (2004)). Attacks where adversaries have full control of a number of authenticated devices and behave arbitrarily to disrupt the network are referred to as Byzantine attacks. The goal of a Byzantine node is to disrupt the communication of other nodes in the network, without regard to its own resource consumption (Awerbuch, et al. (2004)). So, it is very hard to detect. In fact, a basic Byzantine attack is a black hole attack where the adversary stops forwarding data packets, but still participates in the routing protocol correctly. Routing attack is launched at disrupting the data transmission of the network. In routing attacks, routing table overflow, routing table poisoning, packet replication, rushing attacks (Hu, et al. (2003)) are reported. The most general attacks to WSN routing are spoofing, replaying, or altering routing-control information. In these attacks the adversary injects bogus routing information into the network. This leads to routing inconsistencies, and, as a consequence increases end-to-end delays and packet loss in the network. Fortunately, these types of attacks can be effectively prevented using link-layer authentication and anti-replay techniques. In a sinkhole attack, an attacker makes a compromised node look more attractive to its neighbors by forging the routing information.

2.2.4 Transport layer attack

At the Transport Layer attacks target the protocols that provide transfer of data between end systems. When explicit connections between identifiable nodes are used, either end of the connection maintains some form of connection control block. An attacker can issue a large number of connection setup requests that result in the exhaustion of memory at the end nodes. This is called a TCP SYN flood attack. Flooding and de-synchronization attacks are specific to transport layer. Flooding can be as simple as sending many connection requests to a susceptible node. In this case, resources must be allocated to handle the connection request. Eventually a node's resources will be exhausted, thus rendering the node useless. Another vulnerability is by session hijacking attack, where the adversary takes control over a session between two nodes. The adversary node masquerades as one of the end nodes of the session and hijacks the session. Another kind of Transport Layer attack is the desynchronization attack. This attack targets the transport protocols that rely on sequence numbers. An attacker issues forged packets with wrong sequence numbers and, as a result, causes retransmissions, which waste both energy and bandwidth.

2.2.5 Multilayer attack

Multilayer attacks are those that could occur in any layer of the network protocol stack. In a denial-of-service (DoS) attack, an attacker attempts to prevent legitimate users from accessing information or services (Wood, et al. (2002)). DoS attacks are commonly launched from one or more points on the Internet that are external to the victim's own system or network. In many cases, the launch point consists of one or more systems that have been subverted by an intruder via a security-related compromise rather than from the intruder's own system or systems. DoS attacks on the Internet may be launched by botnets and carried

out by compromised machines running zombie processes in the background unbeknownst to the owner of the machine, thus the risk for physical identification and apprehension of the attacker is reduced.

2.3 WSN security mechanisms

In this section, we briefly describe the different important security mechanisms to prevent some of the above mentioned attacks. Good amount of research efforts are engaged in finding solutions to nullify the adversary's intention. WSN security mechanisms mainly consist of robust cryptographic techniques, efficient key management, certification and other advanced methods. It is indispensable to provide basic security primitives to the sensor nodes in order to give a minimal protection to the information flow and a foundation to create secure protocols. Those security primitives are Symmetric Key Cryptography (SKC), hash primitives, and Public Key Cryptography (PKC). Since sensor nodes are highly constrained in terms of resources, implementing the security primitives in an efficient way (using less energy, computational time and memory space) without sacrificing the strength of their security properties is one of the major challenges in this area, a challenge that most of the state-of-the-art have managed to achieve. SKC primitives use the same secret key for both encryption and decryption. Instances of these primitives are able to provide confidentiality to a certain information flow, given that the origin and the destination of the data share the same secret key. They can also provide integrity and authentication if a certain mode of operation is used. These algorithms are usually not very complex, and they can be implemented easily in resource-constrained devices. Symmetric cryptography is therefore the typical choice for applications that cannot afford the computational complexity of asymmetric cryptography. Symmetric schemes utilize a single shared key known only between the two communicating hosts. This shared key is used for both encrypting and decrypting data. The traditional example of symmetric cryptography is DES (Data Encryption Standard). The use of DES, however, is quite limited due to the fact that it can be broken relatively easily. In light of the shortcomings of DES, other symmetric cryptography systems have been proposed including 3DES (Triple DES), RC5, AES, and so on (Schneier, (1996)). It can be noted that PKC is better solution where key management is an issue. In the case, where the sensor nodes can manage some amount of computational resources to perform PKC, it is always advisable to apply PKC. SKC suffers from key management problem. PKC, also known as asymmetric cryptography, is a form of cryptography that uses two keys: a key called private key, which has to be kept private, and another key named public key, which is publicly known. Any operation done with the private key can only be reversed with the public key, and vice versa. This nice property makes all PKC-based algorithms useful for authentication purposes. Still, the computational cost of calculating their underlying operations had hindered its application in highly-constrained devices, such as sensor nodes. One of the most promising PKC primitives in the field of WSN security is Elliptic Curve Cryptography (ECC), due to the small size of the keys, the memory and energy savings, and the simplicity of its underlying operation, the scalar point multiplication (Kobiltz. (1987), Liu, et al. (2005)). In order to securely distribute the cryptographic keys among the sensor nodes, efficient key management scheme needs to be deployed. Broadly, WSN key management has two categories: deterministic and probabilistic. In functional terms, three keying models are used tto cater for WSN Security and operational requirements: Network Keying, Pairwise Keying, and Group Keying.

Network keying has the advantage of being simple, flexible, and scalable. It allows data aggregation and fusion and it is able to self-organize (a key requirement in WSN). But it lacks robustness. Pairwise keying provides authentication for each node and it is by far the most robust in nature, which in turn makes it non-scalable, non-flexible and unable to self-organize. Group keying on the other hand is more robust than network keying. It allows group collaboration and multi-cast. It is able to self-organize with in cluster, but cluster formation information is application dependent. It also lacks efficient storage for group keying in IEEE 802.15.4. One of the promising WSN key distribution mechanisms is due to Eschenauer and Gligor (Eschenauer, L. & Gligor, V.D, 2002)). This protocol is simple, elegant and provides effective tradeoff between robustness and scalability. In this scheme a large pool of keys are generated (eg: 10,000 keys). Randomly take 'K' keys out of the pool to establish a key ring (K << N). Path key discovery is made When two nodes communicate they search for a common key within the key ring by broadcasting their identities (ID's) of the keys they have. Let M be the number of distinct cryptographic keys that can be stored on a client node. At the pre-deployment phase, a random pool of keys K out of the total possible key space is chosen. For each node, M keys are randomly selected from the key pool K and stored into the node's memory. This set of M keys is called the node's key ring. The number of keys in the key pool, $|K|$, is chosen such that two random subsets of size M in K shares at least one key with some probability p. After the client nodes are deployed, a key-setup phase is performed. The nodes first perform key-discovery to find out with which of their neighbors they should share a key. This key discovery is securely performed by Merkle puzzle policy (Merkle. (1978)), where each client node issues M client puzzles (one for each of the M keys) to each neighboring node. Any node that responds with the correct answer to the client puzzle is thus identified as a trusted client, who knows the associated key. Client nodes which discover that they contain a shared key in their key rings then verify that their neighbor actually holds the key through a challenge-response protocol. The shared key then becomes the key for that link. After key-setup is complete, a connected graph of secure links is formed.

One needs to find the right parameters such that the graph generated during the key-setup phase is connected. Consider a random graph G (n, p_c) a graph of n clients for which the probability that a link exists between any two nodes is p_c. Erdos and Renyi showed that for monotone properties of a graph G (n, p_c) there exists a value of p_c over which the property exhibits a "phase transition", i.e., it abruptly transitions from "likely false" to "likely true". So, it is possible to calculate some expected degree d for the vertices in the graph such that the graph is connected with some high probability c. Eschenauer and Gligor calculated the necessary expected node degree d in terms of the size of the network n as:

$$d = \left(\frac{n-1}{n}\right)(\ln(n) - \ln(-\ln(c)))$$

From the formula, d (degree of the client node) = O(log n). It can be observed that the key distribution we presented is a generalized one and it can be deployed in multi-hop network. The scheme is scalable and it requires less than N-1 keys to be stored. But it lacks authentication process and does not clearly define any process for revoking or refreshing keys. The dynamic handshaking process prevents any form of data aggregation (eg: one

event detected by two neighboring nodes will result in two separate signals.). it provides no support for collaborative operations and no node is guaranteed to have common key with all of its neighbors, there is a chance that some nodes are unreachable. It also fails to satisfy security requirement authentication and operational requirement accessibility. LEAP is another important key management scheme which needs mentioning. LEAP (Zhu, et al. (2003)) uses four types of keys: Individual, group, cluster and pairwise shared keys. The authentication mechanism known as μ-TESLA is used for the broadcast authentication of the sink node, which ensures that the packets sent with the group are from the sink node only. It also employs one-way hash-key mechanism for source packet authentication. LEAP uses a pre-distribution key to help establish the four types of keys. The individual key is first established using a function of a seed and the ID of the node. Then nodes broadcast their IDs. The receiving node uses a function, seeded with an initial key, to calculate the shared key between it and all of its neighbors. Thirdly, the cluster key is distributed by the cluster head using pairwise communication secured with the pairwise shared key. Lastly for distributing the network-wide group key, the sink node broadcasts it in a multihop cluster-by-cluster manner starting with the closest cluster. It has μ-TESLA and one-way key chain authentication as well as key revocation and key refreshing. The scheme is scalable and able to perform cluster communications. But it works on the assumption that the sink node is never compromised.

Another threat needs to be considred is physical tempering. It can be noted the sensor nodes are embdedded platform, so we have to provide platform security, which is temper proof. Recently, good amount of development has taken place in embedded platform security. Among the commercial relaeses, Trusted Platform Module by Atmel and Trustzone by ARM are worth mentioning. Trusted platform module (TPM) is to provide the minimal hardware needs to build a trusted platform in software. While usually implemented as a secure coprocessor, the functionality of a TPM is limited enough to allow for a relatively cheap implementation – at the price that the TPM itself does not solve any security problem, but rather offers a foundation to build upon. Thus, such a module can be added to an existing architecture rather cheaply, providing the lowest layer for larger security architecture. The main driver behind this approach is the Trusted Computing Group (TCG), a large consortium of the main players in the IT industry, and the successor to the Trusted Computing Platform Alliance (TCPA). TrustZone consists of a hardware-enforced security environment providing code isolation, together with secure software that provides both the fundamental security services and interfaces to other elements in the trusted chain, including smartcards, operating systems and general applications. TrustZone separates two parallel execution worlds: the non-secure 'normal' execution environment, and a trusted, certifiable secure world. TrustZone offers a number of key technical and commercial benefits to developers and end-users. TrustZone software components are a result of a successful collaboration with software security experts, Trusted Logic, and provide a secure execution environment and basic security services such as cryptography, safe storage and

integrity checking to help ensure device and platform security. By enabling security at the device level, TrustZone provides a platform for addressing security issues at the application and user levels. Below (fig. 1 & 2) we show the hardware and software architecture of ARM trustzone for reader's better understanding of a secure computing environment.

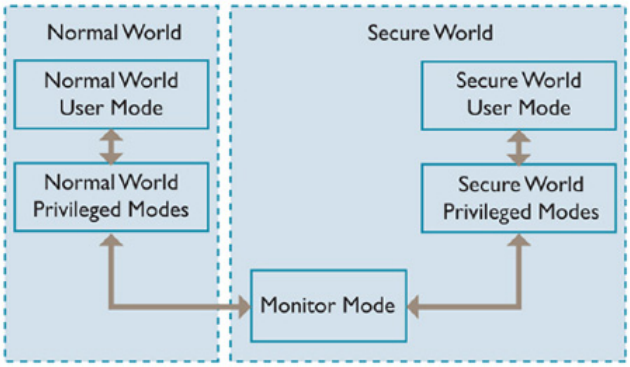


Fig. 1. Trustzone hardware architecture

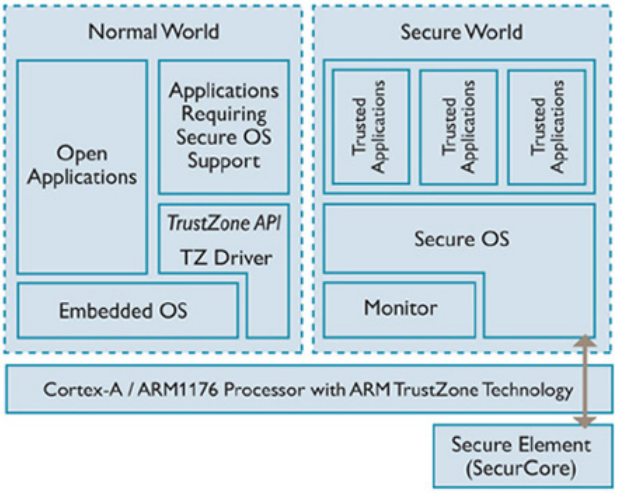


Fig. 2. Trustzone software architecture

2.4 WSN trust and reputation management

Another important aspect of WSN security is trust and reputation management. Secure trust management policy has the responsibility that network activity can continue as securely as possible without affecting the benign entities. It has the additional duty of isolating malicious agents and also to warn benign entities. Good amount of research effort has been made to find practical and reliable trust management models (Josang, et al. (2007), Xiong, et al. (2004)). In fact, trust management which is introduced in (Blaze, et al. (1996)) defined it as "a unified approach to specifying and interpreting security policies, credentials, and

relationships which allow direct authorization of security-critical actions". In (Grandison, et al. (2002)), trust management is defined in a broader sense as: "Trust management is the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships". Traditionally trust management is studied under decentralized control environment (Li., et al. (2003). The authors described different aspects of the trust management problem. They have formulated security policies and security credentials, determined whether particular sets of credentials satisfy the relevant policies, and how deferring trust to third parties could provide better stability of the networks. Rahman and Hailes (Rahman, et al. (1997)) presented a distributed recommendation-based trust model, where conditional transitivity of trust concept is proposed. They have quantified trust as a multi-value concept.

Apart from research community, business houses and commercial organizations use and practice trust management modeling very frequently. Ebay uses reputation based trust management. It has the simple trust rating system for its users. For each successful transaction, sellers and buyers are invited to rate each other on the scale of 1. +1 is positive, 0 for neutral, -1 for negative response. Last six months ratings are taking in account by eBay to calculate a reputation of a user.

There are mainly two approaches for developing trust management system: policy based and reputation based. Policy based mechanisms employ different policy and engines for specifying and reasoning on rules for trust establishment (Stab, et al. (2004)). These mechanisms mostly rely on access control. Trust management based on distribution of certificates is presented in (Davis. (2004)) where trust is re-established by carrying out weighted analysis of the accusations received from different entities. On the other hand, reputation-based approaches have been proposed for managing trust in public key certificates, in P2P systems, mobile ad-hoc networks and in the Semantic Web. Reputation-based trust is used in distributed systems where a system only has a limited view of the information in the whole networks. It can be observed that reputation based trust management system is dynamic in nature (Duma, et al. al. (2005)) and new trust relationship is established frequently based on the malicious activities in the network. The main issues characterizing the reputation based trust management systems are the trust metric generation and the management of reputation data. In (Boukerch, et al. (2007)), agent-based trust and reputation management scheme (ATRM) for WSNs is presented. From this background we develop our reputation based trust modeling. In this model the nodes with collaboration from others form an honest opinion about each other. This model has two layers. In first layer trust model is formed against the selfish behavior of a node. This means that nodes with selfish behavior pattern will be identified, punished and if required isolated from performing any operations. The other layer is the trust modeling against malicious nodes, which falsely accuse other nodes as untrustworthy in order to disrupt the normal network activity.

In order to illustrate this, we refer to fig. 3. In this architecture, there are N number of sensor nodes and they communicate wirelessly. The sensor nodes through multi-hop routing send the sensed data to other nodes in another network or to internet through a cluster head or gateway. In order to properly maintain the self-configuring nature of the network, the nodes need to collaborate. Every node when needs to communicate to the gateway has to route the data in multi-hop. For this, it needs to take help of its neighborhood nodes. Let us consider

the case depicted in fig. 3. Node A needs to send a data to the gateway. Its neighbor consists of the nodes B, C and F. The shortest path for A to reach the gateway is through C and then C-D. But it may turn out that the shortest path is not the trusted path. Node a sends the data to C, but C maliciously drop it or send it to node I, which is another malicious node. So, for A to effectively send the data to gateway it has to first find the trustworthiness of the neighborhood nodes. If A finds B is a trusted node, it sends the data to B for forwarding ignoring C. If A discovers F is more trusted than B, A sends the data to F. The objective is to send the data through the most trusted node even that does not guarantee in shortest path, but this ensures reliability. We can observe that in mission critical or defense application data security and reliable transmission is often much more required than mere energy efficiency. In this case, node A needs to find out the trustworthiness of its neighborhood to update its data. Neighborhood of node A consists of node B, node C and node F. We define few terms as below:

$$T_{A\to B/C/F} = Trust\ value\ of A\ by\ B/C/F$$
$$R_{C\to B/F} = Reputation\ value\ of\ C\ by\ B/F$$
$$R_{B\to C/F} = Reputation\ value\ of\ B\ by\ C/F$$
$$R_{F\to B/C} = Reputation\ value\ of\ F\ by\ B/C$$
$$A_{A\to C} = Age\ of\ reputation\ value\ of\ A\ at\ C$$
$$A_{A\to B} = Age\ of\ reputation\ value\ of\ A\ at\ C$$
$$A_{A\to F} = Age\ of\ reputation\ value\ of\ A\ at\ F$$

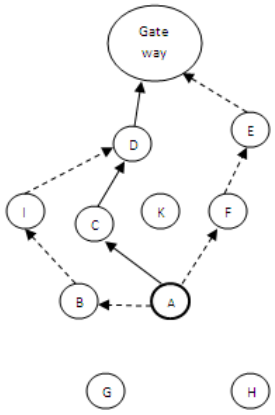


Fig. 3. Trusted node identification in WSN

In the network, individual nodes broadcast the computed reputation value of its entire neighborhood. When a particular node receives such a notification, it stores the values related to its neighbor nodes only and ignores the values of other nodes. For node A, it only accepts the reputation values of node B, C and F, i.e node A considers the reputation values $R_{C,G,H,I,F,K,F,D,E\to B}$, $R_{B,G,H,I,F,K,F,D,E\to C}$. and $R_{C,G,H,I,B,K,F,D,E\to F}$ for nodes B,C and F.

Accordingly, node A finds the reputation values of other nodes C and F. It can be noted that this reputation value cannot be taken as the sole source of trustworthiness of a node. There

are other factors like the age of the reported reputation value and the previous trust value of those nodes which are to be considered to compute the overall trust factor of the node. Taking this into account, reputation value of B by C is:

$$RN_{B\to C} = R_{B\to C} \times T_{B\to C} \times A_{A\to C}$$

where, $RN_{B\to C} = New/\ adjusted\ reputation\ value\ of\ B\ by\ C$

After computing the reputation value of node B, node A computes the trust value of node b as:

$$T_{B\to A} = \frac{\sum_{n\in C,F,G,H,K,D,E,I} RN_{B\to n}}{\sum_{n\in C,F,G,H,K,D,E,I}(T_{B\to n} \times A_{B\to n})}$$

Same way node A computes the trust value of node C and F (its neighborhood nodes). It should be remembered that even if node A does not require sending data, it is always required to compute the trust values of its neighborhood. Otherwise the computed trust value does not reflect the trust history of a node, which may lead to wrong judgment. Based on the latest computed trust values of its neighborhood, node A decides to send the packet through one of its neighbor nodes.

$$\text{Find Max } (T_{n\to A}), \text{where } n \in B, C, F$$

This is to find out the most trustworthy neighbor

Let,

$$T_A = \text{Max } (T_{n\to A}), \text{where } n \in B, C, F$$

Where, T_A is the most trustworthy node

$$\text{Find Min } (S_{n\to A}), \text{where } n \in B, C, F$$

Where , $S_{n\to A}$ is the distance between node A to other neighborhood nodes. This is to find out the shortest possible path.

Let,

$$S_A = \text{Min } (S_{n\to A}), \text{where } n \in B, C, F$$

Where, S_A is the most tshortest path node

Based on the trust values and shortest path parameters available to node A, it decides on the route to send data as per the rule below:

1. *If, $T_A = S_A$, select that node to send data from A*
2. *Else if, $T_{A-1} = S_{A-1}$, select that node where S_{A-1} is the node with next shortest path.*

where T_{A-1} is the node with next best trustworthiness.

3. *Else select T_A, irrespective of S_A*

Now if we consider the generalized case of N number of neighbors for node A, the selection procedure continues upto N/2, i.e. trust value from 1 upto N/2th will be compared with

that of the node with shortest path. Whichever is found the earliest, is selected, else the most trusted node is selected. In other words,

$$\begin{gathered} \textit{If, } T_A = S_A \textit{, select that node to send data from A} \\ \textit{Elseif, A = A-1, upto A= A-N/2-1} \\ \textit{Else, select } T_A \end{gathered}$$

The above stated algorithm enforces reliability of data transfer by selecting the trusted node, even if it is required to send the data through not the shortest path. This algorithm enhances reliability to a larger extent with some extra communication cost by sending data through a non-shortest route. This is very much required for reliable transmission and to adapt to noncooperation in a collaborative computing environment. Our algorithm finds an optimized path between reliability and efficiency. Though at the end, reliability is given preference (when no matching of trusted node and shortest path is found) over efficiency.

3. WSN Privacy

Privacy preservation is an important issue in today's context of extreme penetration of Internet and mobile technologies. It is more important in the case of WSNs where collected data often requires in-network processing and collaborative computing. Researches in this area are mostly concentrated in applying data mining techniques to preserve the privacy content of the data. These techniques are mostly computationally expensive and not suitable for resource limited WSN nodes.

With ubiquitous connectivity, people are increasingly using electronic technologies in business-to-consumer and business-to-business settings. This in effect helps a third party to acquire the confidential and private information from various avenues. Depending upon the nature of the information, users may not be willing to divulge the individual values of records. This has lead to concerns that the private data may be misused for a variety of purposes. Privacy can be defined as the limited access to a person or a process and to all the features related to the person or the process. Privacy preservation is important from both individual as well as organizational perspectives. For example, customers might send to a remote database queries that contain private information. Two competing commercial organizations might jointly invest in a project that must satisfy both organizations' private and valuable constraints, and so on. In order to alleviate these concerns, a number of techniques have recently been proposed to perform the data mining tasks in a privacy-preserving way, which is called Privacy Preserving Data Mining (PPDM). The research of PPDM is aimed at bridging the gap between collaborative data mining and data privacy. Privacy-preserving data mining finds numerous applications in surveillance, in-network processing, which are naturally supposed to be "privacy-violating" applications. The key is to design methods (Sweeney, (2005)), which are effective without compromising on security. In the literature, number of techniques has been illustrated to effectively preserve the privacy of the source data. One of most popular method is randomization. The randomization method is a technique in which noise is added to the data to be privacy-protected. This is done to mask the attribute values of records (Agrawal, et al. (2000). The noise added to the data is sufficiently large so that individual values cannot be recovered.

Therefore, techniques are designed to derive aggregated distributions from the perturbed data values. Subsequently, data mining techniques can be developed in order to work with these aggregate distributions. The randomization method has been traditionally used in the context of distorting data by probability distribution for methods such as surveys. There are two major classes of privacy preservation schemes are applied. One is based on data perturbation techniques, where certain distribution is added to the private data. Given the distribution of the random perturbation, the aggregated result is recovered. In another technique, randomized data is used to data to mask the private values. However, data perturbation techniques have the drawback that they do not yield accurate aggregation results. It is noted by Kargupta et al. (Kargupta, et al. (2005)) that random matrices have predictable structures in the spectral domain. This predictability develops a random matrix-based spectral-filtering technique which retrieves original data from the dataset distorted by adding random values. There are two types data perturbation. In additive perturbation, randomized noise is added to the data values. The overall data distributions can be recovered from the randomized values. Another is multiplicative perturbation, where the random projection or random rotation techniques are used in order to perturb the values. In tune of their argument, we can apply the second technique of masking the private data by some random numbers to form additive perturbation.

Our one of the objectives of privacy preserved secured data aggregation falls under the broad concept of Secure Multiparty Computation (SMC) (Goldreich. (2002)). SMC and privacy preservation are closely related, particularly when some processing or computation is required on the data records. Historically, the SMC problem was introduced by Yao (Yao, et al. (2008)), where a solution to the so-called Yao's Millionaire problem was proposed. In general SMC problem deals with computing any (probabilistic) function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output. Consider a system model (fig. 4). There are N numbers of source nodes. Each source i owns a value x_i which it is not willing to share with other parties. Suppose that the sum is in the range [0, M]. Our objective is to find out the sum X privately without revealing the private data x_i, i=1,2, … , N to each other as well as to the server.

$$X = \sum_{i=1}^{N} x_i$$

The process is initiated by the server. The server randomly chooses one of the source nodes and signals it to initiate the process. The source node first chosen by the server is denoted by c_1. This node possesses its private data x_1 and it generates one random number r_1 between the range [0, M], which is denoted as r_1. It then computes R_1.

$$R_1 = (r_1 + x_1) mod P$$

where P is an arbitrarily large number

After computing R_1, the source node c_1 performs neighborhood discovery to find out the other source nodes it is connected to. This information c_1 passes to the server. Server keeps the knowledge of the nodes already participated. If the source nodes connected to c_1 is not already participated, the server randomly chooses one of those non-participated source nodes and sends that message to c_1. Let this next source node be c_2. Now, accordingly c_1 passes R_1 to c_2.
The source node c_2 computes R_2.

$$R_2 = (R_1 + x_2) mod P$$

The source node follows the same procedure as c1 and sends R_2 to c_3. This way c_N is reached, which computes R_N.

$$R_N = (R_{N-1} + x_N) mod P$$

The server, when it finds out that all the nodes are participated, it asks the last node to send R_N to it. Server now directs the first source node c_1 to compute the summation as:

$$X = (R_N - r_1) mod P$$

The source node after computing the summation sends that value to the server. The server may process it or sends that value for further processing.

Ukil and Sen (Ukil & Sen, (2009)) considers a scenario where data aggregation needs to be done in privacy-preserved way for distributed computing platform. There are number of data sources which collect or produce data. The data collected or produced by the sources is private and the owner or the source does not like to reveal the content of the data. But the collected data from the source is to be aggregated by an aggregator, which may be a third party or part of the network, where the data sources belong. The data sources do not trust the aggregator. So the data needs to be secure and privacy protected. The computation for the aggregation is based on the concept of SMC. SMC allows parties with similar background to compute results upon their private data, minimizing the threat of disclosure. Consider a set of parties who neither trust each other, nor the channels by which they communicate. Still, the parties wish to correctly compute some common function of their local inputs, while keeping their local data as private as possible. Generally, this problem can be seen as a computation of a function f (x1, x2, ..., xn) on private inputs x1, x2, ...,xn in a distributed network with n participants where each participant i knows only its input xi and no more information except output f (x1, x2, ..., xn) is revealed to any participant in the computation. In this case the function is SUM.In this scheme, the property of modular arithmetic to recover the aggregated value is considered and data privacy is preserved through randomization process. The security part is handled by random key pre-distribution method which is modified version of (Eschenauer, L. & Gligor, V.D, 2002). The scheme is simple in nature with low computational complexity, which makes it suitable for practical implementation particularly in the case where the source nodes do not have much computational capabilities.

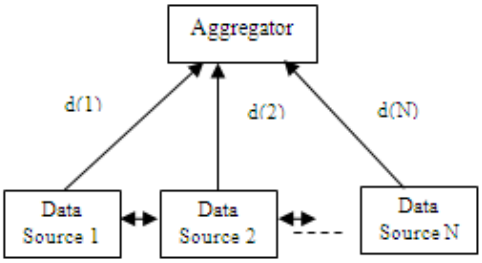


Fig. 4. SMC scheme illustration

The aggregation methods of privacy-preservation are dealt well in (Conti, et al. (2009)). In (He, et al. (2007)), He et.al. propose schemes to achieve data aggregation while preserving privacy. The scheme they proposed, CPDA (Cluster-based Private Data Aggregation) performs privacy-preserving data aggregation in low communication overhead with high computational overhead. This privacy-preservation data aggregation policy is based on the additive property of the polynomial. The objective of this algorithm is that the server or the aggregator can not make out the individual content of the data sent be the sink node. In the system model described, the friend pairs' data are aggregated together. After receiving the aggregated data of all the friend pair the server sends that to the base station. It is shown in the Fig. 5. In order to illustrtae this, we assume server/aggregator as node 'A' and two sink nodes of the friend pair is 'S1' and 'S2'.This algorithm consists of two parts:

1. Value distortion: Let the data values in the sink node S1 and S2 be x and y and z be the dummy variable at the aggregator node 'A'. In the first step, the server/aggregator sends three seeds a,b and c to the friend pairs. Based on that A computes

$$\alpha_{S1}^{A} = z + R_1^A b + R_2^A b^2$$
$$\alpha_{S2}^{A} = z + R_1^A c + R_2^A c^2$$
$$\alpha_{A}^{A} = z + R_1^A a + R_2^A a^2$$

where R_1^A and R_2^B are two random numbers generated by A.
Similarly, S1 computes

$$\alpha_{S1}^{S1} = x + R_1^{S1} b + R_2^{S1} b^2$$
$$\alpha_{A}^{S1} = x + R_1^{S1} a + R_2^{S1} a^2$$
$$\alpha_{S2}^{S1} = x + R_1^{S1} c + R_2^{S1} c^2$$

Similarly S2 computes

$$\alpha_{A}^{S2} = y + R_1^{S2} a + R_2^{S2} a^2$$
$$\alpha_{S1}^{S2} = y + R_1^{S2} b + R_2^{S2} b^2$$
$$\alpha_{S2}^{S2} = y + R_1^{S2} c + R_2^{S2} c^2$$

where R_1^{S1} and R_2^{S1} are two random numbers generated by sink node S1, R_1^{S2} and R_2^{S2} are other two random numbers generated by sink node S2. After that, the calculated, α_{S1}^{A} and α_{S2}^{A} are sent to sink node S1 and sink node S2 by A, securely as described earlier. Similarly,

$\alpha_A^{S1}$and α_{S2}^{S1} are sent to sink node S2 and A by sink node S1 and α_A^{S2} and α_A^{S2} and α_{S1}^{S2} are sent to A and sink node S1 by sink node S2.

2. Value aggregation: After the private data values (x and y) are distorted, all the nodes aggregates the values available to them and generates aggregated result. Sink node calculates Ψ_{S1} , sink node S2 calculates Ψ_{S2} and A calculates Ψ_A.

$$\Psi_A = \alpha_A^A + \alpha_A^{S1} + \alpha_A^{S2} = (x + y + z) + R_1 a + R_2 a^2$$
$$\Psi_{S1} = \alpha_{S1}^A + \alpha_{S1}^{S1} + \alpha_{S1}^{S2} = (x + y + z) + R_1 b + R_2 b^2$$
$$\Psi_{S2} = \alpha_{S2}^A + \alpha_{S2}^{S1} + \alpha_{S2}^{S2} = (x + y + z) + R_1 c + R_2 c^2$$

where, $R_1 = R_1^A + R_1^{S1} + R_1^{S2}$ and $R_2 = \mathrm{R}_2^{\mathrm{A}} + \mathrm{R}_2^{\mathrm{S1}} + \mathrm{R}_2^{\mathrm{S2}}$. These aggregated results from sink node S1 and sink node S2 are securely sent to the aggregator A. Now, the aggregator has the simple task to solve the above equation for (x+y+z) with the knowledge of the values of a,b,c and Ψ_A , Ψ_{S1} and Ψ_{S2} . After solving for D = x+y+z, node A internally knows its own data z, so it can find out the result (x+y).

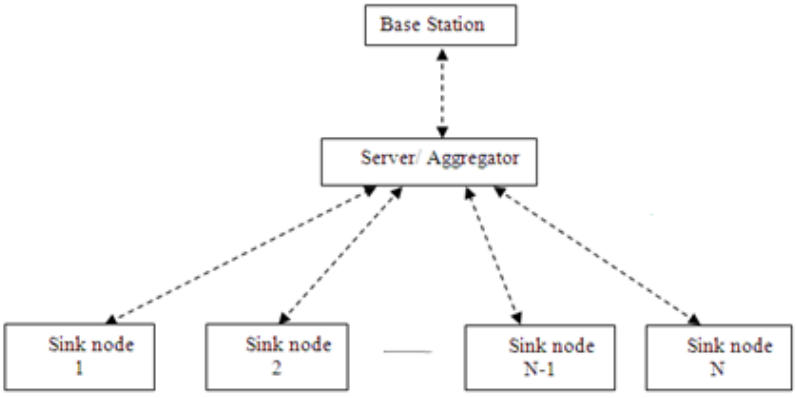


Fig. 5. CPDA scheme illustration

The privacy-preserving data aggregation scheme by Conti et al. (Conti et al. (2009)) first establishes twin keys for different pairs of sensor nodes in a network. Twin key establishment is an anonymous process that prevents each node in a pair from deriving the identity of the other node with which it is sharing a twin key. Then, for each aggregation phase, it uses an anonymous liveness announcement protocol to declare the liveness of each twin key. In the end, during the aggregation phase, each node encrypts its own value by adding shadow values computed from the lively twin keys it holds. In this way, the contribution of the shadow values for each twin key will cancel out each other and the correct aggregated result is finally obtained. Data Aggregation Different Privacy-levels Protection (DADPP) (Yao, et al. (2008))) offers different levels of data aggregation privacy based on different node numbers for pre-treating the data. This protocol is inspired by the work of Shao et al. in terms of different levels of privacy as well as the CPDA in terms of the privacy achieving method (Shao et al. (2007)). In DADPP, a hierarchical wireless sensor network is first constructed in such that sensor nodes form several clusters each of which

has a fixed cluster head below the energy efficient Base sation. According to the desired privacy level, all nodes within the same cluster are partitioned into multiple groups belonging to the same privacy level. Data are pretreated only in the same group and privacy levels are defined by the size of groups. The lowest privacy level consists of partitioned groups that have at least 3-sensor-nodes. The upper privacy level corresponds to portioned groups with 4-sensor-nodes. By analogy, if all sensor nodes of a cluster belong to a single group, they consider this case as the highest privacy level. The data aggregation process is similar to that of the CPDA. First, original data are pretreated in each group. Secondly, the cluster head aggregates all pretreated data. Finally, data are aggregated on the plane of the cluster head up to the BS. The hierarchical wireless sensor network is illustrated in Figure 6. Although DADPP reduces traffic by partitioning a cluster with n sensor nodes into multiple in-networks with pretreatment of groups according to the desired privacy-levels, it suffers from the inherent high communication and computation overheads. Furthermore, these overheads increase with increasing privacy level.

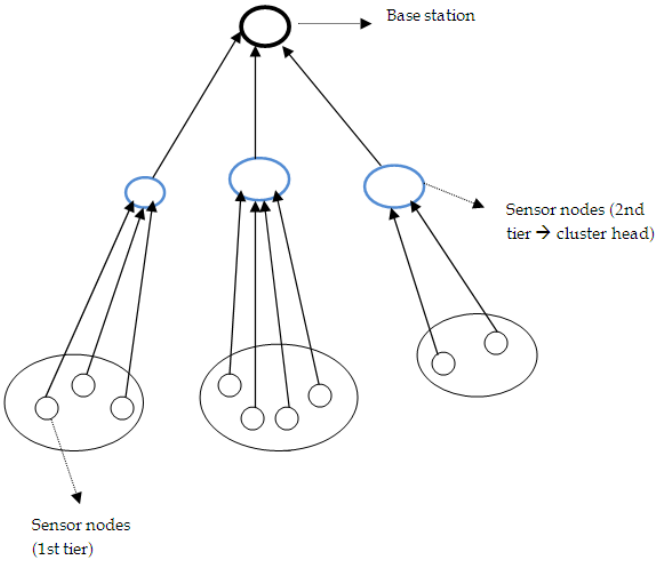


Fig. 6. Hierarchical WSN

Zhang et al. (Zhang, et al. (2008)) proposed the Perturbed Histogram-based Aggregation (PHA) to preserve privacy for queries targeted at special sensor data or sensor data distribution. The perturbation technique is applied to hide the actual individual readings and the actual aggregate results sent by sensor nodes. For this, every sensor node is preloaded with a unique secret number which is known exclusively by the sink and the node itself. Sensor nodes and the sink form a tree. The basic idea of PHA is to generalize the values of data transmitted in a WSN, such that although individual data content cannot be decrypted, the aggregator can still obtain an accurate estimate of the histogram of data distribution and thereby approximate the aggregates. In particular, before transmission, each sensor node first uses an integer range to replace the raw data. Next, with a certain

granularity, the aggregator plots the histogram for data collected and then estimates aggregates such as MIN, MAX, Median and Histogram. Although the PHA supports many data aggregation functions, it has the following disadvantages. First, the final aggregated result is an approximation value of the sensor data rather than the real data. Secondly, the PHA requires a large size payload (message/data) because all sensor data need to be replaced by an integer range. Moreover, the bandwidth consumption of this protocol increases as the number of ranges increases. Finally, storing interval ranges to replace the original data consumes a significant amount of memory. To address Privacy-preserving Integrity-assured data Aggregation (PIA) for WSNs, recently, Taban et al. proposed four distinct symmetric-key solutions (Taban et al. (2009)). In their single aggregator model, an aggregator node is used as an intermediary between the user (i.e., a third party) and the sensor nodes that aggregates the sensor data and forwards the query response to the user. The problem is that the user wants to verify the integrity of the received aggregate value whereas the network owner does not want the user to access the original data. Privacy Homomorphism (PH) has a special feature that allows arithmetic operations to be performed on cipher-text without decryption. This technique is fast and resource-efficient for privacy-preserving data aggregation, but it has a limitation that it performs only addition and multiplication operations. Before sensor data are sent to the aggregators, they are encrypted by using the respective keys of sensor nodes and they are added or multiplied without decryption. Concealed Data Aggregation (CDA) (Ferrer. (2002)) is a type of PH scheme, which conceals the process of data aggregation in WSN by using Domingo-Ferrer's (DF) approach (Deng, et al. (2006)). In this protocol, each sensor node splits its data into d parts ($d \geq 2$), encrypts them by using a public key and transmits them to the aggregator node. The aggregator node operates on the encrypted data, computes an aggregated value from the data without decryption and sends it to the sink.

Context-oriented privacy protection focuses on protecting contextual information, such as the location (Xi. Et al. (2006)) and timing (Kamat, et al. (2007)) information of traffic transmitted in a WSN. Location privacy concerns may arise for such special sensor nodes as the data source (Mehta, et al. (2007)) and the base station (Jian, et al. (2007). Timing privacy, on the other hand, concerns the time when sensitive data is created at data source, collected by a sensor node and transmitted to the base station. This type of privacy is also of primary importance, especially in the mobile target tracking application of WSNs, because an adversary with knowledge of such timing information may be able to pinpoint the nature and location of the tracked target without learning the data being transmitted in the WSN. Furthermore, the adversary may be able to predict the moving path of the mobile target in the future, violating the privacy of the target. Similar to data-oriented privacy, context-oriented privacy may also be threatened by both external and internal adversaries. Nonetheless, existing research has mostly focused on defending against external adversaries, because such adversaries may be able to compromise context privacy easily by monitoring wireless communication. Within the category of external adversaries, one can further classify adversaries into two categories, local attackers and global attackers; based on the strength of attacks an adversary is capable of launching. Local attackers can only monitor a local area within the coverage area of a WSN, and therefore have to analyze traffic hop-by-hop to compromise traffic context information. On the other hand, a global attacker has the capability (e.g., a high-gain antenna) of monitoring the global traffic in a WSN. One

can see that a global attacker is much stronger than a local one. To further protect the location of the data source, fake data packets can be introduced to perturb the traffic patterns observed by the adversary. In particular, a simple scheme called Short-lived Fake Source Routing was proposed in (Kamat, et al. (2005)) for each sensor to send out a fake packet with a pre-determined probability. Upon receiving a fake packet, a sensor node just discards it. Although this approach perturbs the local traffic pattern observed by an adversary, it also has limitations on privacy protection. Specifically, to maintain the energy-efficiency of the WSN, the length of each path along which fake data is forwarded is only one hop, therefore, an adversary is able to quickly identify fake paths and eliminate them from consideration.

Another aspect of privacy preservation is anonymity, where the identity of the origin and/or the destination of a conversation is hidden from adversaries unless it is intentionally disclosed by the user. Ring signature (Rivest, et al. (2001)) is a signer-ambiguous signature scheme, first introduced by Cramer et al in 1994. With ring signature, a set of possible users (signers) should be specified and each user should be associated with the public key of some standard signature scheme such as RSA. To generate a ring signature, the actual signer declares an arbitrary set of possible signers that must include himself, and computes the signature of any message by himself using only his secret key and the other's public keys. Ring signatures can be verified by the intended recipient as a valid signature from one of the declared signers, without revealing exactly which signer actually produced the signature. Ring signatures provide an elegant way to leak authoritative secrets in an anonymous way and can be used to solve multiparty computation problems. In the case of anonymous access authentication, ring signatures allow a legitimate user to hide his true identity among an arbitrarily selected set of other users. The non-linkability of multiple transactions of the same user is also well protected.

4. Conclusion

In this chapter, we present on the issues of security and privacy in WSN. We provide a comprehensive study regarding the requirements, different kind of well-known attacks and some of the proposed solution to counter the security attacks on WSN. We also emphasise on the embedded device security where industry has recently given a lot of attention. We have touched upon the concept of trust and reputation based security analysis in WSN. In fact, we attempt to make the main focus of this chapter on privacy preservation aspects of WSN. It is found that WSN security is well-researched compared to the privacy preserving issues. So, our endeavour was to bring that privacy protection problem in WSN. In that regard, we have provided detailed description of some of the important schemes and present the privacy preservation of WSN both from functional and requirement perspectives.

5. References

Chan, H.; Perrig, A. & Song, D. (2003). *Random key predistribution schemes for sensor networks*, Proceedings IEEE Symposium on Security and Privacy, pp. 197 - 213. IEEE Computer Society.

Liu, D.; Ning, P. & Li, R. (2005). *Establishing pairwise keys in distributed sensor networks*, ACM Trans.Inf. Syst. Secur., vol. 8, no. 1, pp. 41–77.

Newsome, J.; Shi, E.; Song, d. & Perrig, A. (2004). *The Sybil Attack in Sensor Networks: Analysis & Defenses*, IEEE International Workshop on Information Processing in Sensor Networks (IPSN'04), Berkeley, USA.

Weiser, M. (1991). *The Computer for the Twenty First Century*, Scientific American, pp. 94-104, September, 1991.

Karlof, C. & Wagner, D. (2003). *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasure*, Ad-Hoc Networks, vol. 1, no. 2-3, pp. 293-315, Elsevier, September 2003.

Law, Y. W.; Doumen, J. & Hartel, P. (2006). *Survey and Benchmark of Block Ciphers for Wireless Sensor Networks*, ACM Transactions on Sensor Networks, vol. 2, no. 1, pp. 65-93, February, 2006.

Alarifi, A. & Du, W. (2006). *Diversifying Sensor Nodes to Improve Resilience against Node Compromise*, 2006 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06),Alexandria, USA, October 2006.

Gaubatz, G.; Kaps, J.P.; Öztürk,E. & Sunar, B. (2005*). State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks*, IEEE International Workshop on Pervasive Computing and Communication Security (PerSec'05), Hawaii, USA, March 2005.

Shi, E. & Perrig, A. (2004). *Designing secure sensor networks*, Wireless Communication Magazine, vol. 11, no. 6, pp. 38-43, December 2004.

Wang, X.; et al. (2005). *Search-based physical attacks in sensor networks: modeling and defense*, Technical report, Department of Computer Science and Engineering, Ohio State University, February 2005.

Wang, X.; et al. (2004). *Sensor network configuration under physical attacks*, Technical report (OSU-CISRC-7/04-TR45), Department of Computer Science and Engineering, Ohio State University, July 2004.

Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y. & Cayirci,E. (2002). *A survey on sensor networks*, IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, August 2002.

Wood, A.D. & Stankovic,J.A. (2002). *Denial of service in sensor networks*, IEEE Computer, vol. 35, no. 10, pp. 54-62.

Hu, Y.; Perrig,A. & Johnson,D.B. (2003). *Packet Leashes: A defence Against Wormhole Attacks in Wireless adhoc Networks*, IEEE INFOCOM, vol. 3, pp. 1976 – 1986.

Newsome, J.; Shi, E.; Song, D. & Perrig, A. (2004). *The sybil attack in sensor networks: analysis & defenses*, Proceedings of the third international symposium on Information processing in sensor networks, pp. 259–268. ACM Press.

Douceur, J. (2002). *The sybil attack*, Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), February 2002.

Deng, J.; Han, R. & Mishra, S. (2004). *Countermeasuers against traffic analysis in wireless sensor networks*, Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

Awerbuch, B.; et al. (2004). *Mitigating Byzantine Attacks in Ad HocWireless Networks*, Technical Report version 1, March 2004.

Hu, Y.; Perrig, A. & Johnson, D.B. (2003). *Rushing Attacks and Defense in Wireless ad Hoc network Routing protocols*, ACM workshop on Wireless Security, pp. 30 – 40, 2003.

Raymond, D.; et al. (2006). *Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocols*, Proceedings of 7th Annual IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop (IAW), pp. 297–304.

Karlof, C. & Wagner, D. (2003). *Secure routing in wireless sensor networks: Attacks and countermeasures*, Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, May 2003, pp. 113-127.

B. Schneier. (1996). *Applied Cryptography*, Second Edition, John Wiley & Sons.

Kobiltz, N. (1987). *Elliptic curve cryptosystems*, Mathematics of Computation, vol. 48, pp. 203-209.

Liu, A. & Ning, P. (2005). *TinyECC: Elliptic Curve Cryptography for Sensor Networks* (version 0.1), September 2005.

Eschenauer, L. & Gligor, V.D. (2002). *A key-management scheme for distributed sensor networks*, 9th ACM Conference on Computer and Communication Security, pp. 41–47.

Merkle, R. (1978). *Secure communication over insecure channels*, Communications of the ACM, vol. 21, no.4, pp. 294–299.

Spencer, J. (2000). *The Strange Logic of Random Graphs*, Algorithms and Combinatorics, no.22, 2000.

Zhu, S.; Setia, S. & Jajodia, S. (2003). *LEAP: Efficient security mechanism for large –scale distributed sensor networks*, Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 62-72, New York, NY, USA, ACM Press.

www.atmel.com

www.arm.com

https://www.trustedcomputinggroup.org

Sweeney, L. (2005). *Privacy Technologies for Homeland Security, Testimony before the Privacy and Integrity Advisory Committee of the Department of Homeland Security*, Boston, MA, Sep. 28, 2005.

Agrawal, R. & Srikant, R. (2000). *Privacy-Preserving Data Mining*, ACM Sigmod, pp. 439–450.

Kargupta, H.; Dutta, S.; Wang, Q. & Sivakumar, K. (2005). *Random-data perturbation techniques and privacy-preserving data mining*, Knowledge and Information Systems, vol. 7, no. 4, pp. 387–414.

Goldwasser, S. (1997). *Multi-party computations: Past and present*, 16th Annual ACM symposium on Principles of distributed computing, pp. 1–6.

Conti, M.; et al. (2009). *Privacy-preserving robust data aggregation in wireless sensor networks*, Security and Communication Networks (Wiley), vol. 2, pp. 195–213.

Wright, M.; Adler, M.; Levine, B.N. & Shields, C. (2003). *Defending anonymous communications against passive logging attacks*, IEEE Symposium on Security and Privacy, pp. 28–41.

Eschenauer, L. & Gligor, V.D. (2002). *A key-management scheme for distributed sensor networks*, 9th ACM Conference on Computer and Communication Security, pp. 41–47.

Goldreich, O. (2002). *Secure multi-party computation*, Working Draft, First version posted in June, 1998 and final revision posted in Oct, 2002.

Yao, A. (1982). *Protocols for secure computations*, 23rd Annual Symposium on Foundations of Computer Science, pp. 160–164.

He, W.; Liu, X.; Nguyen, H.; Nahrstedt, K. & Abdelzaher, T. (2007). *PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks*, IEEE Infocom, pp. 2045–2053.

Rivest, R.; Shamir, A. & Tauman, Y. (2001). *How to leak a secret*, Advances in Cryptology - ASIACRYPT 2001.

Conti, M.; Zhang, L.; Roy, S.; Pietro, R.D.; Jajodia, S. & Mancini, L.V. (2009). *Privacy-preserving robust data aggregation in wireless sensor networks*, Secur. Commun. Netw, no. 2, pp.195–213.

Yao, J.; & Wen, G. (2008). *Protecting classification privacy data aggregation in wireless sensor networks,* Proceedings of the 4th International Conference on Wireless Communication, Networking and Mobile Computing, WiCOM, Dalian, China, October 12–14, 2008; pp. 1–5.

Shao, M.; Zhu, S.; Zhang, W. & Cao, G. (2007). *Pdcs: Security and privacy support for data-centric sensor networks*, Proceeding of 26th IEEE International Conference on Computer Communications, INFOCOM, Anchorage, AK, USA, May 6–12, 2007; pp. 1298–1306.

Zhang, W.S.; Wang, C. &Feng, T.M. (2008). *GP2S: Generic privacy-preservation solutions for approximate aggregation of sensor data, concise contribution*, Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom, Hong Kong, China, March 17–21, 2008; pp.179–184.

Taban, G. & Gligor, V.D. (2009). *Privacy-preserving integrity-assured data aggregation in sensor networks*, Proceeding of International Symposium on Secure Computing, SecureCom, Vancouver, Canada, August 29–31, 2009; pp. 168–175.

Ukil, A. & Sen, J. (2010). *Secure Multiparty Privacy Preserving Data Aggregation by Modular Arithmetic*, International conference on parallel, distributed, and Grid Computing, pp. 329 - 334, Oct, 2010.

Sen, J. (2009). *A Survey on Wireless Sensor Network Security*, International Journal of Communication Networks and Information Security (IJCNIS), vol. 1, no. 2, pp.55 - 78 , Aug. 2009.

Girao, J.; Westhoff, D. & Schneider, M. (2005). *CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks*, In Proceedings of IEEE International Conference on Communications, ICC, Seoul, Korea, May 16–20, 2005; volume 5, pp. 3044–3049.

Domingo-Ferrer J. (2002). *A provably secure additive and multiplicative privacy homomorphism*, Proceedings of the 5th International Conference on Information Security, Sao Paulo, Brazil, September 30–October 2, 2002; pp. 471–483.

Deng, J.; Han, R. & Mishra, S. (2006). *Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks*, Pervasive and Mobile Computing Elsevier, vol. 2, no. 2, pp.159–186.

Xi, Y.; Schwiebert, L. & Shi, W.S. (2006). *Preserving source location privacy in monitoring-based wireless sensor networks*, Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), April 2006.

Kamat, P.; Xu, W.Y.; Trappe, W. & Zhang, Y.Y. (2007). *Temporal privacy in wireless sensor networks*, Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS 2007), June 2007, pp. 23–23.

Mehta, K.; Liu, D.G. & Wright, M.(2007). *Location privacy in sensor networks against a global eavesdropper*, Proceedings of the IEEE International Conference on Network Protocols (ICNP 2007), October 2007, pp. 314–323.

Jian, Y.; Chen, S.G.; Zhang, Z. & Zhang, L. (2007). *Protecting receiver-location privacy in wireless sensor networks*, Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007), May 2007, pp. 1955–1963.

Kamat, P. Zhang, Y.Y.; Trappe, W. & Ozturk, C. (2005*). Enhancing source location privacy in sensor network routing*, Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005), June 2005, pp. 599–608.

Grandison, T. & Sloman, M. (2000). *A Survey of Trust in Internet Applications*, IEEE Communications Surveys and Tutorials, vol. 3, no. 4, September 2000.

Jøsang, A.; Ismail, R. & Boyd, C. (2007). *A survey of trust and reputation systems for online service provision*, Decision Support Systems, vol. 43, no. 2, pp.618–644, March 2007.

Blaze, M. Feigenbaum, J. & Lacy, J. (1996). Decentralized trust management, In Proceedings of IEEE Conference on Security and Privacy.

Grandison T. & Sloman, M. (2002). *Specifying and analysing trust for internet applications; Towards The Knowledge Society: eCommerce, eBusiness, and eGovernment*, The Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002), IFIP Conference pp. 145–157.

Li, N. & Mitchell, J.C. (2003). *Datalog with Constraints: A Foundation for Trust-management Languages*, Proceedings of the Fifth International Symposium on Practical Aspects of Declarative Languages pp. 58–73, January 2003.

Abdul-Rahman, A. & Hailes, S. (1997). *A distributed trust model,* Proceedings of New Security Paradigms Workshop, ACM, pp. 48 – 60, 1997.

www.ebay.com.

Staab, S.; et al. (2004). *The pudding of trust*, IEEE Intelligent Systems, vol. 19, no. 5, pp.74–88.

Davis, C. (2004). *A localized trust management scheme for ad-hoc networks*, 3rd international conference on Networking.

Duma, C.; Shahmehri, N. & Caronni, G. (2005). *Dynamic trust metrics for peer-to-peer systems*, Proc. of 2nd IEEE Workshop on P2P Data Management, Security and Trust, August 2005.

Boukerch, A.; Xu, L. & EL-Khatib,K. (2007). *Trust-based Security for Wireless Ad Hoc and Sensor Networks*, Computer Communication, vol. 30, pp. 2413-2427.

Xiong, L. & Liu, L. (2004). *PeerTrust: Supporting reputation based trust in peer to peer communities*, IEEE Transactions on Data and Knowledge Engineering, Special Issue on Peer to Peer Based Data Management, vol. 16, no. 7, pp. 843–857, July 2004.