Collaboration of Mobile and Pervasive Devices for Embedded Networked Systems

by

Su Jin Kim

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2010 by the
Graduate Supervisory Committee:

Sandeep K. S. Gupta, Chair
Partha Dasgupta
Hasan Davulcu
Yann-Hang Lee

ARIZONA STATE UNIVERSITY

December 2010

ABSTRACT

Embedded Networked Systems (ENS) consist of various devices, which are embedded into physical objects (e.g., home appliances, vehicles, buidlings, people). With rapid advances in processing and networking technologies, these devices can be fully connected and pervasive in the environment. The devices can interact with the physical world, collaborate to share resources, and provide context-aware services. This dissertation focuses on collaboration in ENS to provide smart services. However, there are several challenges because the system must be - scalable to a huge number of devices; robust against noise, loss and failure; and secure despite communicating with strangers.

To address these challenges, first, the dissertation focuses on designing a mobile gateway called Mobile Edge Computing Device (MECD) for Ubiquitous Sensor Networks (USN), a type of ENS. In order to reduce communication overhead with the server, an MECD is designed to provide local and distributed management of a network and data associated with a moving object (e.g., a person, car, pet). Furthermore, it supports collaboration with neighboring MECDs. The MECD is developed and tested for monitoring containers during shipment from Singapore to Taiwan and reachability to the remote server was a problem because of variance in connectivity (caused by high temperature variance) and high interference. The unreachability problem is addressed by using a mesh networking approach for collaboration of MECDs in sending data to a server. A hierarchical architecture is proposed in this regard to provide multi-level collaboration using dynamic mesh networks of MECDs at one layer. The mesh network is evaluated for an intelligent container scenario and results show complete connectivity with the server for temperature range from 25°C to 65°C.

Finally, the authentication of mobile and pervasive devices in ENS for secure collaboration is investigated. This is a challenging problem because mutually unknown devices must be verified without knowledge of each other's identity. A self-organizing region-based authentication technique is proposed that uses environmental sound to autonomously verify if two devices are within the same region. The experimental results show sound could accurately authenticate devices within a small region.

To my parents

ACKNOWLEDGMENTS

I wish to express my deep gratitude to my advisor, Dr. Sandeep K. S. Gupta, for his continuous encouragement, confidence and support, and for sharing with me his knowledge and experience. I am very thankful to Dr. Partha Dasgupta, Dr. Hasan Davulcu and Dr. Yann-Hang Lee who served as my committee.

I would like to thank my colleagues, Guofeng Deng, Krishna Venkatasubramanian, Gianni Giorgetti, Ayan Banerjee, and all of the people at IMPACT lab who have made it such a great place to work. A special thank to Dr. Georgios Varsamopoulos and Dr. Tridib Mukherjee for their valuable advice and encouragement during my study at IMPACT lab.

I also thank Intel Corp. and the National Science Foundation (through grant CT-0831544) for funding my research.

I thank my friends, Myungsoo Bae and Jongwook Kim, for making my time in Arizona enjoyable.

Last but not least, I am very grateful to my parents and my brother for their continuous moral support and encouragement. I hope I will make them proud of my achievements, as I am proud of them. Their love accompanies me wherever I go.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

1. INTRODUCTION

In recent years, advance in computing technology has made computing devices embedded in the environments or objects. At the same time, the rapid deployment of wireless networks has enabled connecting these embedded systems for better services. For example, in a health monitoring application, several medical devices (e.g., thermometers, heart monitors and blood oxygen meters) can be embedded in a human's body. These devices themselves are embedded systems, but adding wireless communication capability to such medical devices can help in accessing patient's health information remotely.

Over the next years, embedded systems, wireless network technology and pervasive computing will combine to provide smart services in people's everyday life.

Pervasive computing (also called ubiquitous computing) is a new computing paradigm that seamlessly integrates with the environment and enables a service to be available anytime anywhere [1]. In pervasive computing, a system senses the physical environment, process information, and adapts its behavior according to this information called context. Context can be computing context (e.g., network connectivity and local available resources), user context (e.g., user profiles and location), physical context (e.g., noise level and temperature), or temporal context (e.g., day, month and season of year). Context awareness of pervasive computing makes the system more intelligent and invisible to users [1].

We refer to this system as an *Embedded Networked System* (ENS) [2]. ENS can be widely used in many areas such as military, industrial and civilian application areas including geophysical monitoring (seismic activity), precision agriculture (soil management), habitat monitoring (tracking of animal herds), transportation (traffic monitoring), business processes (supply chain management), and in the future, possibly cooperating smart everyday things.

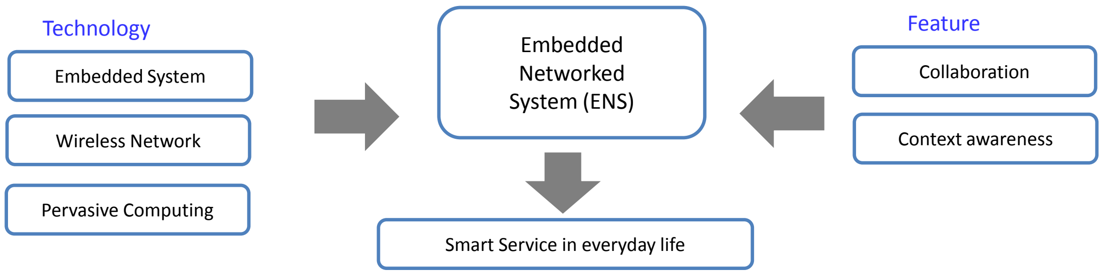


Fig. 1. Embedded Networked Systems (ESN).

1.1. Example Architecture of ENS

To understand the overall system architecture of ENS, this section discusses a type of ENS called a Ubiquitous Sensor Network (USN).

A USN is a type of ENS that couples a wireless sensor network with the concept of pervasive computing [3] [4]. The purpose of a USN is to provide information of objects such as location and condition to anyone irrespective of the time or place. Fig. 2 shows the architecture of USN [3] [4] [5].

A wireless sensor network is a wireless network consisting of large number of sensor nodes equipped with embedded processors, sensors and radios [6]. These sensor nodes collaborate to monitor physical or environmental conditions, such as temperature, humidity, light, vibration, or pressure.

In the USN, Radio-Frequency IDentification (RFID) is considered to be a special type of sensors used for finding the location of moving objects [7]. RFID is a term to describe a system to transmit the identity of an object using radio waves. RFID is similar to bar code identification, but it does not require direct contact and line-of-sight scanning. An RFID system consists of two components: a tag and a reader. The tag (also called a transponder) is the identification device attached to the item to track. The reader

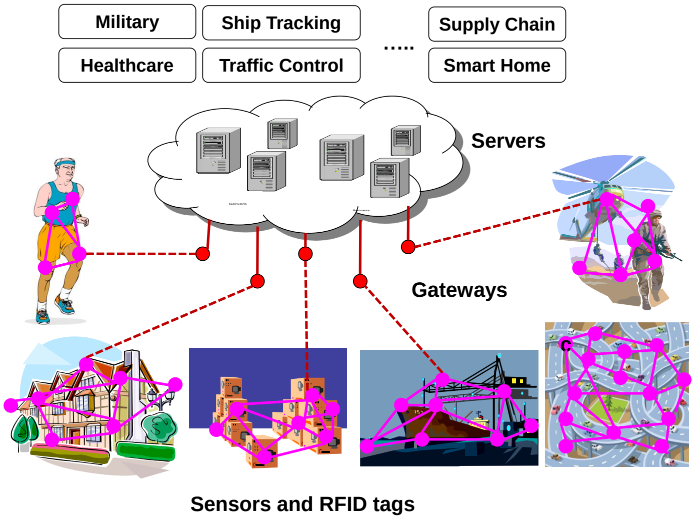


Fig. 2. Architecture of USN.

(also called a transceiver) is the device which reads the signal from tags and transfers the information to a processing device [8]. RFID systems can be used anywhere which needs unique identification, generally for access control and tracking. Because RFID can be used to detect the presence of a moving object, RFID can supplement wireless sensor networks [7].

In the USN, end nodes (sensors and RFID devices) can be attached to any objects. The data sensed by end nodes are sent to the remote server through a gateway. The gateway is the interface between sensor nodes and the server. It collects data from sensor nodes, reformats it, and delivers it to the server.

1.2. Collaboration in Embedded Networked Systems (ENS)

Compared to a general purpose computer (e.g., a desktop or laptop computer) that is designed for a variety of functions, the embedded system has a set of specific tasks for which the system is made [9].

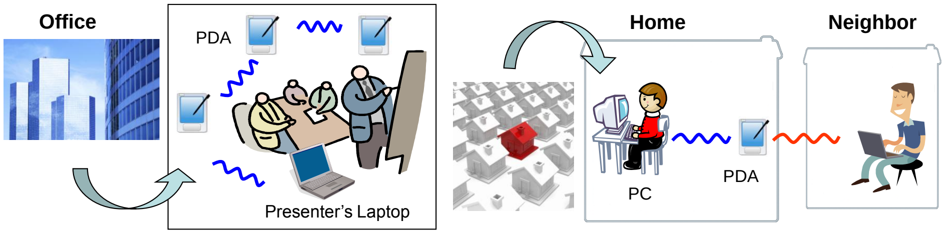


Fig. 3. Example Scenario: Smart Space.

By networking capability, ENS supports the collaboration of these embedded systems that perform their given tasks individually. Such collaboration essentially aims to enhance the system performance as well as context awareness by sharing resources or contextual information among embedded systems, generally systems that are located nearby.

1.2.1. Example Scenarios

To explain benefits by such collaboration in ENS, this section will give example scenarios in two popular applications: a smart space and an intelligent container.

The smart space is one of the most popular applications in ENS [2]. It includes a wide range of applications from home automation, health care, and education applications to game/entertainment applications. Fig. 3 shows one example scenario of collaborative works in the smart space. Suppose Alice has a meeting with her co-workers in a conference room. The mobile devices of people in the conference room (e.g., laptops, smart phones or cell phones) can automatically form a group and exchange their contact information. The presentation files in her laptop can be transfered to a projector in the conference room. This group membership must be dynamically changed based on the user's location. That means those devices are no longer within a group when they leave

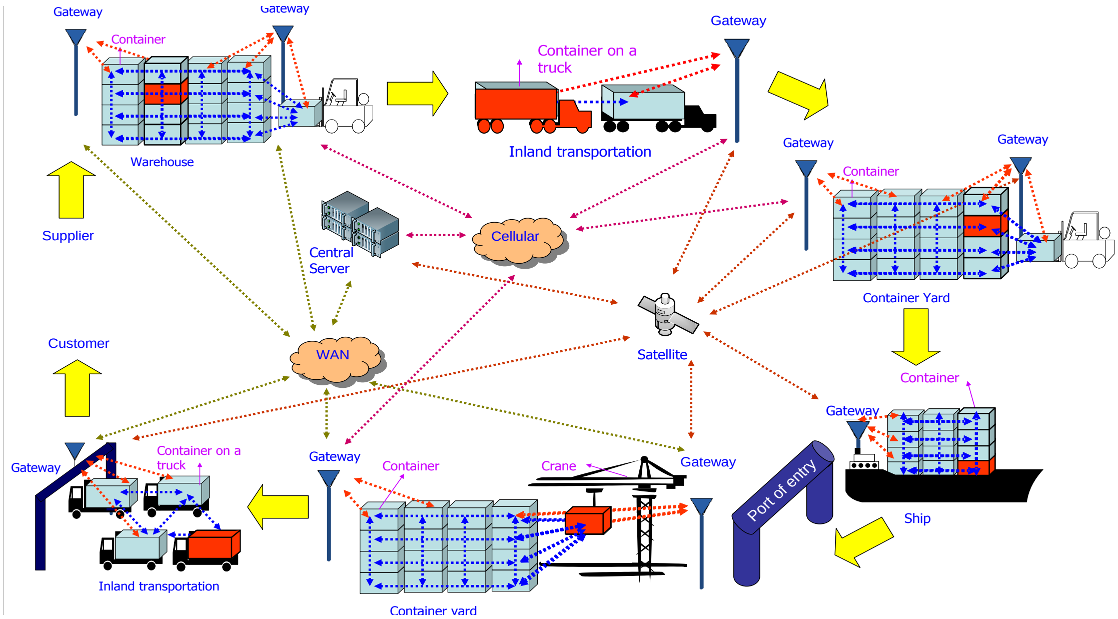
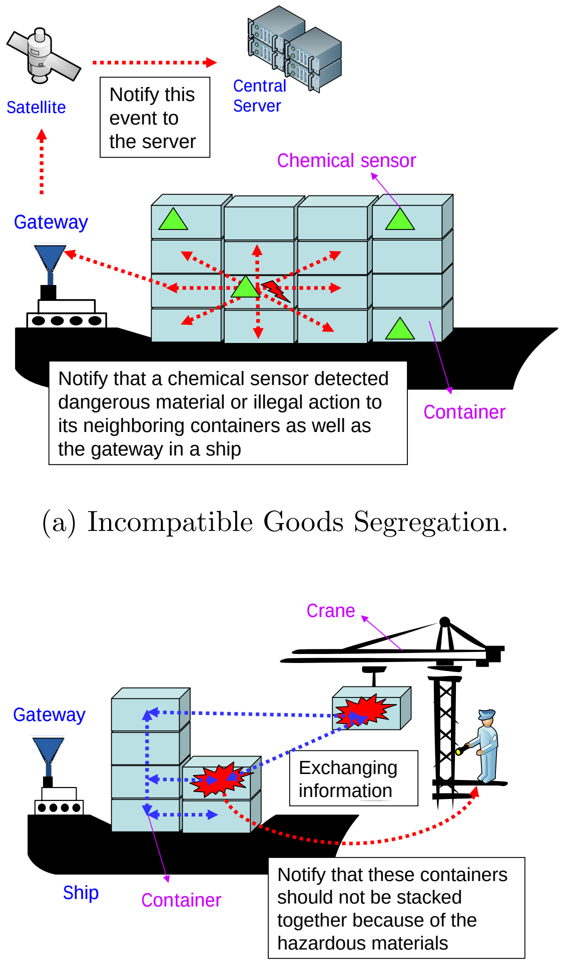


Fig. 4. Example Scenario: Intelligent Container.

that conference room. After work, her mobile devices should join a home network to communicate with other home appliances and devices.

Another example application is the intelligent container system [10] [11] that provides a real-time monitoring system for cargo containers for homeland security and global supply chain management. In the global supply chain, cargo containers move together in a ship, truck, or train and the network can be dynamically realigned with new neighboring containers throughout the whole process of the supply chain shown in Fig. 4. The collaboration between neighboring containers via this network can support monitoring of containers and also enhance the *security of containers*. We will give more details below.

- *Incompatible Goods Segregation*: As per the UN (United Nations) Recommendations on the Transport of Dangerous Goods, incompatible goods shall be segregated from one another during transport [12]. For example, two substances are considered mutually incompatible when their stowing together may result in undue hazards



Notify this event to the server

Notify that a chemical sensor detected dangerous material or illegal action to its neighboring containers as well as the gateway in a ship

(a) Incompatible Goods Segregation.

Exchanging information

Notify that these containers should not be stacked together because of the hazardous materials

(b) Hazardous Materials Monitoring.

Fig. 5. Sharing Information via Container Networks.

in the case of leakage, spillage, or any other accident. This container network can routinely ensure the segregation of incompatible materials through automatic sensing and interchange of information via the mesh shown as Fig. 5(a).

- *Hazardous Materials Monitoring*: Containers which transport hazardous materials are rigorously tracked throughout the supply chain. Hazardous materials may be radioactive, flammable, explosive, toxic, corrosive, biohazardous, an oxidizer, an asphyxiant, a pathogen, an allergen, or may be anything that can harm people, other living organisms, or the environment [13]. Expensive chemical or biological sensors may not be installed in every container. In this case, sensed data is communicated to not only a container on which the sensors are mounted, but by taking advantage of the mesh network characteristics, also to containers in the vicinity. Therefore, exchanging information about hazardous materials with neighboring containers can extend the safety of the entire network shown as Fig. 5(b).

1.3. Design Principles and Challenges

This section discusses the basic design principles and challenges for ENS.

- *Scalability*: ENS consists of a large number of devices that continue to grow. Data generated by these devices can be enormous. Additionally, these devices may need to communicate this data to the base station. Due to this characteristic, ENS should be able to handle an ever-increasing number of devices as well as data and thus scalability is critical in designing ENS.
- *Minimal user distraction*: As the number of personal devices per person grows, it is not reasonable to expect a user to pay attention and efforts to manage these many devices (e.g., configuration, recharging, and updating new softwares). Ideally, a

pervasive computing solution is completely invisible to users so that the technology disappears from a user's conciousness. Because it is very hard to achieve perfect invisibility, in practice, user distraction needs to be minimized.

- *Robustness*: Robustness is one of the challenging problems in ENS development. ENS can be deployed in harsh environments such as the sea and forest, or place with a lot of blockages. Therefore, it may be subject to increased node and link failure rates. It is essential that the system can meet the mission requirements despite the increased failure rates.
- *Resource Constraint*: One of important characteristics of ENS is resource constraints. Most of devices in ENS are battery powered and small-scale mobile devices. Typically, size and energy constraints mean limited resources such as the CPU performance, memory, and wireless communication bandwidth. While the system is capable to support required functionality, it should be efficient in terms of computation, communication and energy.
- *Heterogeniety*: Another issue is heterogeneity. ENS consists of various types of devices from a laptop to a sensor which is designed for a specific set of tasks. Therefore, they are diverse with respect to computation, communication, and energy capabilities. To deal with such heterogeneous devices, the system should be flexible.

1.4. Outline and Contributions

In this section, we discuss the principal contributions of the dissertation. This dissertation focuses on collaboration of neighboring gateways because of the following reasons: 1) In ENS, interaction between embedded systems is mostly local and 2) neighboring systems could detect same events. For example, when there is an earthquake or fire, several local systems will detect it and send an alert to the remote server. Instead of passing all messages to the remote server, a gateway can remove redundant information sent to the remote server. Further, the gateway also can share this information to other local gateways directly. By supporting collaboration among local gateways, we can reduce the amount of remote communication as well as the response time.

The goals of the dissertation are:

- to support collaboration among neighboring gateways that consist of a large number of end devices with high and unpredictable mobility patterns.
- to ensure reliable end-to-end communication between end nodes and remote users even though there are node or link failures.
- to support authentication for a dynamic network of neighboring gateways that have no knowledge about each other.

1.4.1. Scalability Problem in ENS

The first reseach question is how to support collaboration of neighboring gateways for a large ESN. It is important because of the following reasons. First, the ESN consists of a larger number of end nodes and the network size may further keep increasing to millions or billions of nodes in the near future. Moreover, several end nodes installed on a moving object (e.g., a person or vehicle) are moving around the world and the mobility pattern of each object may be unknown. It is difficult to dynamically handle these network changes for a large network. Second, gateway level collaboration requires additional local data processing on a gateway as well as additional local communication with neighboring gateways. Therefore, scalability problem becomes more critical.

To address this problem, this dissertation uses a *mobile gateway* that can move with objects. The mobile gateway is installed on a moving object such as a person or vehicle and manages a local network of these end nodes in a distributed manner. Even though an object moves to the other place, any changes of the other networks do not affect the internal network of the object. The distributed system with such mobile gateways is shown to scale well with the increasing network size (see Section 3.1).

This work focuses on USN to design such a mobile gateway called a *Mobile Edge Computing Device* (MECD). An MECD manages the local network of sensor nodes that are associated with an object. In order to supports local processing required by USN applications, we investigate the general functional requirements of USN applications. Based on these requirements, we design a component-based functional architecture of the MECD in Section 3.2.

Section 3.3 provides an MECD prototype implementation using the off-the-shelf devices. To show the feasibility of our MECD, we apply this design to an intelligent container system discussed in Section 1.2.1 which is one of the popular applications of USN. Section 3.4 presents the details of an intelligent container case study including the test-bed implementation and tests in an actual shipment scenario.

The primary contributions of this work are:

- *MECD*: The concept of MECD is introduced. The feasibility of the proposed component-based architecture is shown by implementing the prototype.
- *Case Study*: An intelligent container system is developed and tested in an actual shipment scenario. During this test, we learn important lessons for the future research directions.

1.4.2. Unreachability Problem in ENS

In our ENS, the mobile gateway plays an important role as an interface between the end nodes and the server. Although it is responsible for connecting the end nodes to the external network, it might not have a direct connection to the server because of the physical environmental constraints. From the experimental studies in Section 3.4, we learned that containers can be in a harsh environment such as sea that affects connectivity between the mobile gateway and the server. Further, because of the high interference from metal and dense materials in and around the containers, the container stacked at the bottom might not be able to communicate with the server.

In order to solve this problem of *unreachability*, this work uses a *dynamic mesh network among neighboring gateways* so that they can collaborate in sending sensed data to the server. This mesh-networking approach can support *reliable connectivity* to the server by redundant paths through the gateways.

In Section 4.1, we provide a *hierarchical* system architecture that support dynamic mesh networks among MECDs at one level separately.

Section 4.2 provides evaluation of the MECD-level network in terms of reachability to the server, network latency, and energy efficiency. Considering the intelligent container scenario, we perform simulations with different temperatures, network densities, and network sizes. The simulation results show that the MECD-level network can support complete server reachability for ISO (International Organization for Standardization) standard containers (6.1 m $\leq$ the container length $\leq$ 16.2 m) within the temperature range of 25°C to 65°C. We also measure the average path length and the total energy consumption of the MECD-level network to analyze the network latency and energy efficiency. The simulation results reveal that the additional network delay and energy consumption generated by the MECD-level network is small. Finally, the simulation is conducted to see the effects of the transmission power on the server reachability, network latency, and energy efficiency.

The contributions of this work are:

- *Dynamic Mesh Networking Approach*: The dynamic mesh network of local MECDs is used in order to ensure the reliable communication between the mobile gateway and the server.
- *Performance Analysis*: The MECD-level network is analyzed in terms of the server reachability, network latency, and energy efficiency. Further, an experimental study shows the need for a dynamic scheduling of transmission power on the basis of the temperature and container size to minimize the total energy consumption.

1.4.3. Authentication for ENS

The last part of this dissertation is to provide a new authentication mechanism for ENS. An ENS is connected over a mix of wireless and wired networks. Due to the inherent characteristic of wireless networks, they are more vulnerable to various attacks. Therefore, authentication is necessary before access to resources and services via networks is granted. The existing authentication mechanisms use a combination of the user input such as a password, a trusted-third party, a pre-shared key or some biometrics. However, such mechanisms for traditional computing environments are not appropriate for ENS because of the following reasons.

- Mobile devices in ENS can move anywhere and communicate with strangers. Therefore, mutually unknown devices need to verify each other's identity.
- Another challenging problem is that this authentication process should be invisible to users because of the design requirement of ENS as discussed in Section 1.3.

- In ENSs, there is no fixed infrastructure. Thus, the authentication process must be autonomous.

Therefore, the goal of this work is to develop a *self-organizing authentication* scheme for mobile and pervasive devices that have never met before and have no pre-shared trust.

In this dissertation, we define the *region-based authentication* problem as the process to prove one's identity and authenticity by detecting its current region. In a traditional authentication mechanism, a user claims who he is, however, in this work, a user claims where he is. This work focuses on the region-based authentication problem because in many ESN applications (e.g., smart home, supply chain management, and location-based billing applications), devices in a particular region (e.g., a room, warehouse or building) form a local network to collaborate.

To address the region-based authentication problem, we propose to take a cyber-physical approach [14] which uses physical environments to generate a secret in Section 5.3. Especially, this work uses the environmental sounds to generate a feature. The primary reason is that only devices within the same region (e.g., a room) can hear the similar sound. Therefore, the similarity of acoustic features is then used to verify the device's current region.

For the self-organizing region-based authentication using environmental sounds, the main research question is which acoustic feature extraction technique is appropriate. This question is addressed because of the resource limitation of ENS. There have been many studies on acoustic feature extraction in speech recognition and audio classification. However, they are too expensive to perform on the resource limited devices in ENS. Section 5.3.2 presents an experimental study on acoustic feature extraction techniques to meet requirements for the region-based authentication problem. In addition, trade-off between accuracy and overhead is discussed in Section 5.3.2. Section 5.3.4 explains the proposed authentication protocol and Section 5.3.5 provides the experimental results in a smart home scenario. The results show that with the threshold for the feature similarity, $t \geq 40\%$, complete dinstincitiveness can be achieved. The proposed scheme with sound could work within a small region, but it may not work for a large region.

The contributions of this work are:

- *Acoustic-based Solution*: To address the self-organizing region-based authentication problem, a simple acoustic-based solution is proposed.
- *Experimental Study*: Different acoustic feature extraction techniques are performed with different environmental sounds. Our expermental results show that the frequency domain technique provides more accurate distinctiveness than the time domain techniques. In addition, the trade-off between accuracy and overhead is studied experimentally.
- *Performance Evaluation*: The proposed acoustic-based solution is implemented on a Google Android Dev 1 phone and evaluated in terms of the false positive rate and false negative rate.

2. BACKGROUND

This chapter describes the system model, assumptions, and technical/theoritical background to understand the problems in designing ENS.

2.1. System Model and Assumptions

This dissertation assumes that the network consists of *end nodes, gateways*, and *remote servers*. An end node itself is an embedded system that has an embedded processor and wireless communication capability. However, the end node may not be able to support long-distance communication. End nodes are usually small, resource-limited and inexpensive. They can be static, but mostly mobile. Further, it is assumed that mobile end nodes can move freely around the world. When a node moves, it must participate in a local network to collaboration with neighboring nodes. Therefore, the network is assumed to be dynamic.

The remote server is assumed to have a considerably higher performance than the end nodes. While an end node is performing given tasks, data can be collected and processed at a remote server. Therefore, it is necessary to connect end nodes to the external network.

The gateway is assumed to be capable of local and remote communication so that it acts as an interface between end nodes and servers. It is also assumed that gateways have more computational resources than end nodes to perform some local data processing. The gateways also can be static or mobile. In this dissertation, they are assumed to collaborate with other neighboring gateways.

2.2. Preliminaries

2.2.1. Wireless Mesh Networks

Wireless mesh networking is an active research field. A mesh network is a type of network that establishes an ad hoc network and maintains mesh connectivity [16]. In a

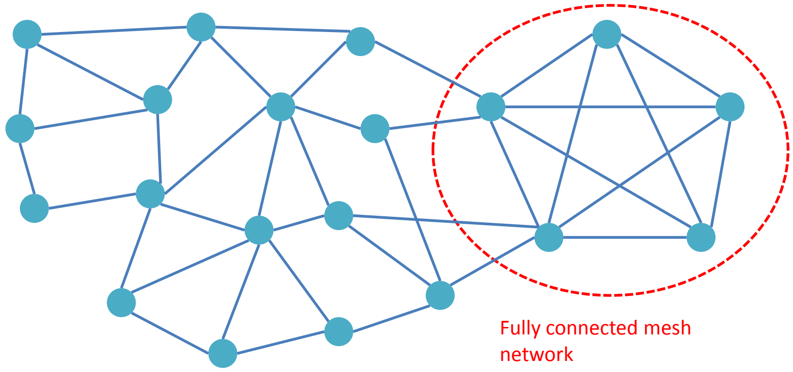


Fig. 6. Mesh Networks.

fully connected mesh topology, all nodes are connected directly as shown in Fig. 6. It is the ideal case with respect to the network delay because every node is within one hop. Typically, a node has direct connection to some of the other nodes because of the cost.

In mesh networking, each node only needs to transmit a packet to the next node that is connected directly. Nodes also act as repeaters to deliver a packet to the destinations that is several hops away from the source node. In other words, the mesh network supports *multi-hop* communications. The data is passed from a source to a destination through a path that can be the shortest (in terms of hops), most reliable, or most energy-efficient. Through multi-hop communications, the same coverage with classic wireless networks can be achieved at a much lower cost and transmission power [17].

Wireless mesh networking is appropriate for wireless sensor networks because of the following benefits.

- A wireless mesh network is *reliable* because the redundant paths of mesh networks ensure alternative data paths and no single point of failure. If one node drops out of the network, due to hardware failure or any other reason, its neighbors simply find another route by the *self-healing* capability.

- A wireless mesh network has *self-configuring* capability. When a node is added to a network, the network automatically incorporates a new node into the existing structure without needing any adjustments by a network administrator. Once the node is initially configured, it continuously discovers its neighbors and recomputes the path in a process known as *dynamic routing*.
- If it is needed to extend coverage, improve redundancy and link quality, it can be simply achieved by adding additional nodes. Furthermore, the nodes in mesh networks are easy to install and maintain (as simple as plug-and-play). Due to self-configuring and redundancy characteristics of wireless mesh networks, they support *flexibility* that allows route formation from any source node to any destination node within the network and *scalability* that enables supporting thousands of individual nodes.

2.2.2. Basic Security Requirements and Algorithms

Security in ENS is an important problem, but the security for this new computing environment has not been explored in depth. One research direction of the dissertation is to secure collaborative work in ENS. This section discusses the basic security concepts.

Most security services involve encryption and decryption. Encryption is the translation of original data called plaintext into an unintelligible form called ciphertext. Decryption is the reverse process which converts encrypted data back into its original form. These operations are controlled both by cryptographic algorithms and keys. Broadly, cryptographic algorithms can be classified into three classes as follows [18]:

- *Symmetric ciphers* (also called secret-key or shared-key algorithms) use a single key for both decryption and encryption. A sender uses the key to encrypt plaintext and sends ciphertext to a receiver. The receiver applies the same key to decrypt ciphertext and recover plaintext. They are typically used for ensuring confidentiality of data. Twofish, Serpent, AES, Blowfish, CAST5, RC4, TDES, DES, 3DES and IDEA are examples of symmetric ciphers. Symmetric ciphers are significantly faster than asymmetric ciphers, but the key must be exchanged securely beforehand.

- *Asymmetric ciphers* (also called public-key algorithms), on the other hand, use two different keys: a private key that is kept secret and known to only one person and a public key that is public and available to everyone. The two keys are mathematically interrelated, but it is computationally infeasible to derive one key from the other. A sender encrypts plaintext using a receiver's public key and sends ciphertext to the receiver. The receiver then decrypts it using his private key. They are typically used for verifying certificates that identify communicating peers and for exchanging symmetric cipher keys. RSA and Diffie-Hellman are examples of asymmetric ciphers. These algorithms rely on the use of computationally intensive mathematical functions, such as modular exponentiation, for encryption and decryption.
- *Hashing functions* (also called message digests and one-way encryption) such as MD5 and SHA provide ways of mapping an arbitrary block of data (with or without a key) into a fixed-length string, thereby providing message integrity checks and digital signatures. The ideal hash function has four main properties: (a) it is easy to compute the hash for any given message, (b) it is extremely difficult to construct a message that has a given hash, (c) it is extremely difficult to modify a given text

without changing its hash, and (d) it is extremely difficult to find two different messages with the same hash.

Because asymmetric ciphers tend to be significantly more computationally intensive, they are usually used in combination with symmetric ciphers. In fact, the asymmetric cipher is used to establish a shared key and the symmetric cipher is used to encrypt the actual message using that shared key. This gives the benefits of asymmetric ciphers with the speed of symmetric ciphers.

Security attacks can be divided into two types: passive and active attacks [19].

- *Passive attack*: A passive attacker attempts to eavesdrop the communication that he is not supposed to hear, but he does not alter messages or break a system. Traffic analysis is a type of passive attack that attempts to infer useful information (e.g., the user's identity, location, and communication frequency) without reading the content.
- *Active attack*: An active attacker is actively attempting to harm to a network or system. Thus, it is a more serious problem than the passive attack. For example, an attacker can insert his own data into a message or delete a message. He can playback the message from another connection or previous connection. A man-in-the-middle attack is an example of active attacks. The attacker sits in the middle of the communication link, intercepts messages and changes them with his own messages. In this way, he tries to make the parties believe they are talking to each other directly, while they really are talking to the attacker.

In order to protect the network and system against such attacks, there are five fundamental properties (can be called the security requirements) as follows [20]:

- *Authentication*: is to assure that communication is authentic. In other words, authentication process is the process of proving one's claimed identity. A basic mechanism for this property is checking something you know (e.g., a username and password) or checking something you have (e.g., fingerprint, smart card or voice recognition) [22].
- *Confidentiality*: is to protect a message from disclosure to an unauthorized person. Protecting the confidentiality of a message on a communication channel between A and B means ensuring that no one other than A and B can read the message. The principal mechanism for this is encryption [19].
- *Integrity*: is to assure that the received message has not been altered in any way from the original. In practice, it is impossible to prevent an attacker who has control of a communication channel from modifying a message. Thus, it actually means ensuring that no one can modify the message without the intended receivers noticing [22].
- *Availability*: means that any service or system should be available to an authorized user when it is requested. The threat to availability is called denail-of-service (DoS) [19]. DoS attacks attempt to inhibit normal use of communications facilities. One common way is sending a lot of requests to the target device, such that it disturbs the response to legitimate traffic, or it drains the battery [22].
- *Non-repudiation*: means that communication as well as content of the message can not be denied by one of parties (the sender and receiver) later. Non-repudiation of origin proves that data has been sent, and non-repudiation of delivery proves that it has been received. In other words, when a message is sent, the receiver can verify that the message was in fact sent by the party who claimed to send the

message. When a message is received, the sender can verify that the message was in fact received by the party who claimed to receive the message [19].

2.2.3. Authentication

This section discusses authentication techniques. Authentication is sometimes confused with authorization. Usually, the authorization process is performed after authentication. Authentication is the process of verifying a user's identity while authorization is the process of verifying whether that user has access to a specific resource [19].

The simplest authentication is to use *passwords*. A user proves his identity by transmitting his password. This is what users normally do to log into a computer. Passwords are stored on storage units and they become a valuable target to an attacker. Storing the hashed passwords can solve this problem because an attacker can not reconstruct any password from them easily. Despite this improvement, it is vulnerable to *dictionary attacks* that try to derive a password from a list of words like a dictionary. Generally, dictionary attacks succeed because most people often choose their passwords which are easy to remember, short, and simple (e.g., common words or date). In addition, passwords must be transmitted to the associated entities in a secure way. If a password or a hashed password is transmitted in plaintext across a network, an attacker can record and replay it later to impersonate her. Because the attacker does not need the actual password, hashing does not offer protection. This kind of attack is called *replay attack* [21].

One time passwords are a possible solution for this problem. In this technique, the password is valid for only one session or transaction. One time passwords are typically generated as the output of an algorithm whose inputs are a seed, a counter, or clock [19]. An example technique is using a chain of hashed passwords [21]. Let a series of passwords be $p_0, p_1, \cdots, p_n$. Two entities involved in the authentication process use a hash of p_i at ith authentication step sequentially. Another example is a challenge-response technique. In this technique, the authenticator sends a random callenge and then the peer returns the response on that challenge. The response typically is the random challenge encrypted with their shared key. Both peers must know the shared key and encryption algorithm that will be used for this process, but the key must be unknown to all others [19].

Another way of authentication is using biometrics [23]. Biometric authentication is a method of identifying a person on a physiological or behavioral characteristic. Physiological characteristics include hand or finger images, facial characteristics, and iris recognition. Behavioral characteristics are related to the behavior of a person such as dynamic signature verification, speaker verification, and keystroke dynamics. Biometric authentication technologies such as face, finger, hand, iris, and speaker recognition are commercially available today and are already coming into wide use.

3. MOBILE EDGE COMPUTING DEVICES (MECD)

This chapter will focus on designing the mobile gateway for USN. As discussed in Section 1.1, USN consists of a huge number of end nodes that generate a considerably large amount of sensing data. Accordingly, the interaction of the network is huge. Furthermore, the mobility of nodes makes this problem more complicated. Therefore, scalability is a critical problem in USN.

3.1. Motivation

In the case of USN, we observed that end nodes installed on a moving object such as a person or vehicle are moving together and collaborating with each other. A mobile gateway needs to be dedicated to one object can manage a local network of these end nodes in a distributed manner. Such architectures scale well with increasing network size and help to protect a local network from attacks outside of this network. For example, a person wearing tens of health sensors on his body can have a mobile gateway that manages all these sensor nodes and data generated by them. This local network is managed seperately from external networks so that it will be more scalable and secure.

In a traditional wireless sensor network, the role of gateways is connecting end nodes to remote servers. The gateway typically forwards raw data to a server that processes data from various gateways. With rapid advance in computing and networking technologies, the mobile gateway becomes increasingly more poweful and it can perform increasingly more local data processing. Through the local data processing, the gateway sends only necessary data to the server in order to reduce the amount of data sent from an end node to the server.

In addition to the collaboration among sensor nodes within one object, there can be interaction across objects for collaboration. In USN, however, most of collaboration occurs locally. In other words, while the density of collaboration within an object remains same, the collaboration between objects will decrease when one moves away [21]. Although there is still the need of communicating with the remote users/servers that are located far away, local interaction/collaboration has preponderance. Instead of passing data through the server, direct communication with currently neighboring objects can decrease the number of remote communications and also reduce the response time.

Therefore, we design the mobile gateway called a *Mobile Edge Computing Device* (MECD) for USN. An MECD is dedicated to a moving object in order to manage a local network of end nodes that are associated with that object and process data locally instead of the remote server. The examples of local data processing are alert generation, event detection, and database management that are required for USN applications. Using local data processing of mobile gateways and distributed network management, we can improve scalability and security.

3.2. Functional Architecture of MECD

An MECD uses a component-based architecture because the modularity can help to upgrade the system easily. MECDs must support the major functionalities of USN applications: 1) monitoring of conditions, 2) tracking of mobile objects, and 3) detection of events.

As shown in Fig. 7, an MECD consists of the following various components:

- *Monitoring Process*: gathers data from several end nodes (sensors and RFID devices) as well as neighboring MECDs and expresses them in a summary form. These data will be sent to the Alerting Process and the Database Management components.

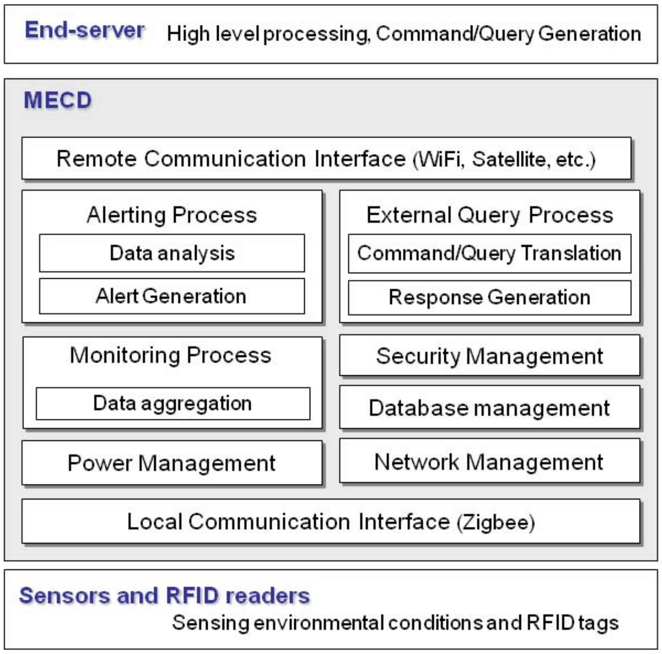


Fig. 7. Block Diagram for MECDs.

- *Alerting Process*: analyzes data gathered by the Monitoring Process component and generates an alert if there is an abnormal event or extreme condition. The alert will be sent to the Local or Remote Communication Interface components based on its type.
- *External Query Process*: translates a query command into a proper sensor query. Based on the query type, the query will be sent to the other components such as Local Communication Interface, Database Management, Network Management, or Power Management. When the response generation module receives a response, it generates a message in an appropriate format and then forwards the message to the Remote Communication Interface component.
- *Remote Communication Interface*: links components of the MECD with the remote server.

- *Local Communication Interface*: connects components of the MECD to sensor nodes, RFID devices, and neighboring MECDs.
- *Network Management*: provides tools to manage networks such as network configuration, routing, neighbor discovering, and bandwidth management.
- *Database Management*: helps to access the local database.
- *Power Management*: manages the sleeping mode and frequency of the periodic reporting on the basis of end nodes' battery status.
- *Security Management*: performs security protocols for key management, authentication, and encryption/decryption.

3.3. Prototype Implementation of MECD

To implement the MECD prototype, we use a CrossBow Stargate gateway [24]. The Stargate is a powerful single-board embedded Linux computer designed for wireless sensor networking applications. The Stargate has low power consumption and various interfaces such as USB, PCMCIA, Compact Flash (CF), and a 51-pin connector. Through these interfaces, various communication and processing capabilities can be supported.

Fig. 8 shows our configuration of the Stargate. A CrossBow MicaZ mote [25] is connected via a 51-pin connector for 2.4-GHz Zigbee communication, and the 802.11 CF card provides remote WiFi access. The Stargate is loaded with a Postgres SQL database system and a USB memory card to store the sensed data. Ethernet might be used for maintenance, for example, updating a program or dumping data.

As shown in Fig. 9 indicates, our MECD prototype was packaged with waterproof materials to ensure survivability in harsh environments. In the box, there is a Stargate and a DC/DC converter that converts the input power to 5 V for the Stargate.

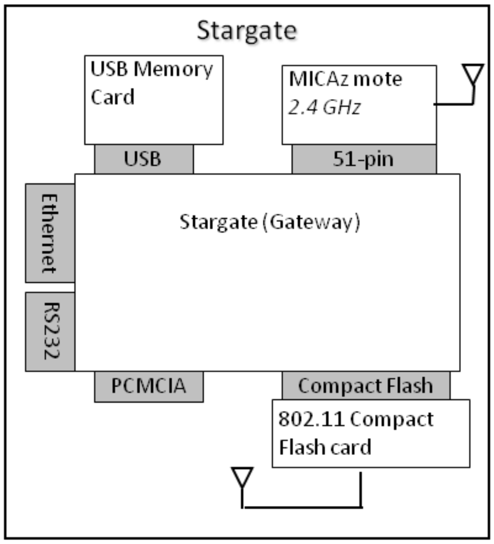


Fig. 8. Configuration of the MECD Implementation.

Fig. 10 describes the software components of our MECD prototype and the data flow between them. All software components were written in C.

With our configuration, the MECD communicates with the server through the 802.11 card driver. If the GPRS modem driver is installed, the MECD can also use it for remote communication. Socket server daemons help to establish connection with other devices.

Once the query is received from the server, the query manager component performs the External Query Process described in the previous Section 3.2. On the basis of the query type, the query is forwarded to the internal network abstract component or the database management component. The internal network abstract component is responsible for translating a query into an appropriate format as part of the External Query Process. The formatted sensor query is then sent out to the sensor network. In our configuration, the 51-pin serial port on the Stargate connects to the MicaZ mote. Therefore, the serial forwarder provides a bridge between the serial port and the network.

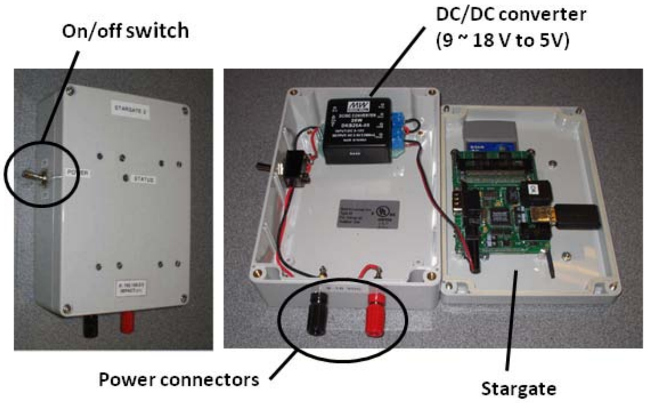


Fig. 9.   Waterproof Package of the MECD implementation.

We use the C-based serial forwarder, sf, provided by Crossbow. The data aggregation component performs the Monitoring Process. Collected data will be sent to the database management to be stored along with the report components. The report component and the external network abstraction component perform the Alerting Process. Radio management is part of the Network Management and determines the outgoing wireless interface dynamically when there are several available communication ways (e.g., 802.11, Zigbee, or GPS). The decision can be based on resource availability, efficiency, and network condition. We did not implement the radio management component at this stage because our prototype uses only 802.11 WiFi for remote communication. However, the future solution will include this functionality.

3.4. Case Study: Intelligent Container System

This section discusses the actual use of our MECD design in an intelligent container application. Our intelligent container system is a comprehensive solution to provide end-to-end visibility of cargo containers for homeland security as well as global supply chain management.

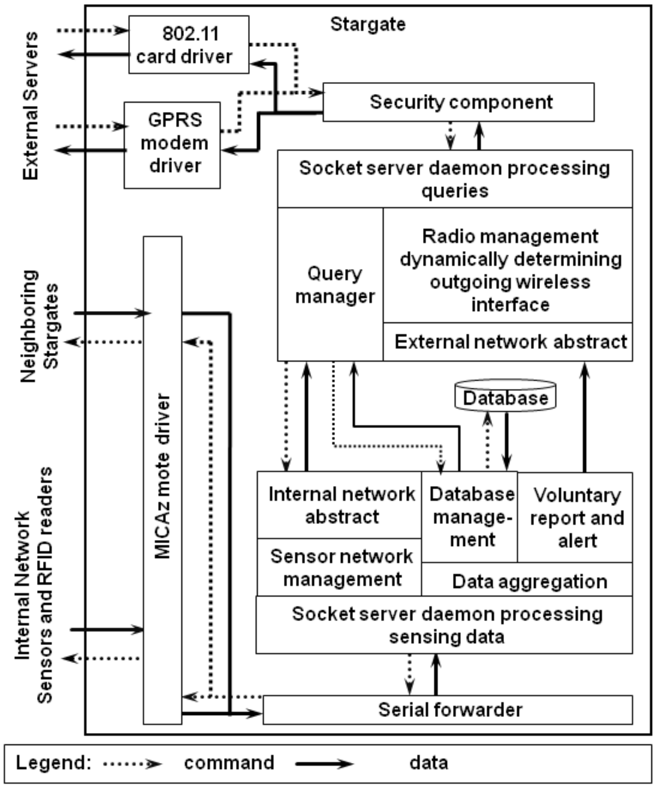

External Servers
Neighboring Stargates
Internal Network Sensors and RFID readers
Stargate
802.11 card driver
GPRS modem driver
Security component
Socket server daemon processing queries
Query manager
Radio management dynamically determining outgoing wireless interface
External network abstract
Database
MICAz mote driver
Internal network abstract
Database manage-ment
Voluntary report and alert
Sensor network management
Data aggregation
Socket server daemon processing sensing data
Serial forwarder
Legend: command data

Fig. 10. Software Components and Data Flow for our MECD.

3.4.1. Functional Requirements

An intelligent container requires the following functionalities:

- *Container Integrity*: is essential for homeland security and supply chain. To ensure the container integrity, a container is sealed during transit. The system should be able to detect an unauthorized door opening action and a breach on any side of a container.
- *Tracking*: can be carried out in multiple levels such as containers, packages, or items. The system must be able to support real-time tracking.
- *Monitoring*: of containers' condition must be provided because abnormal conditions such as extremely high temperature or humidity might cause damage to containers or packages.
- *History of Journey*: must be maintained during a journey and can be any information, including when, who, how, and what.

For more details, Section A.1 includes the requirements for homeland security driven by government regulation.

Supporting these functional requirements by an MECD is straightforward. The Monitoring Process component provides tracking and monitoring functionalities, and the Alerting Process component supports alerts against container integrity. History of journey is managed by the Database Management component. The other components are also used for managing and helping these three components.

3.4.2. Test-bed Configuration

Fig. 11 depicts the configuration of our test-bed. Within the container, we deployed several CrossBow MicaZ [25] and TelosB [26] motes. An MECD and an RFID Reader-Mote module were installed on the container's door. These devices inside the containder

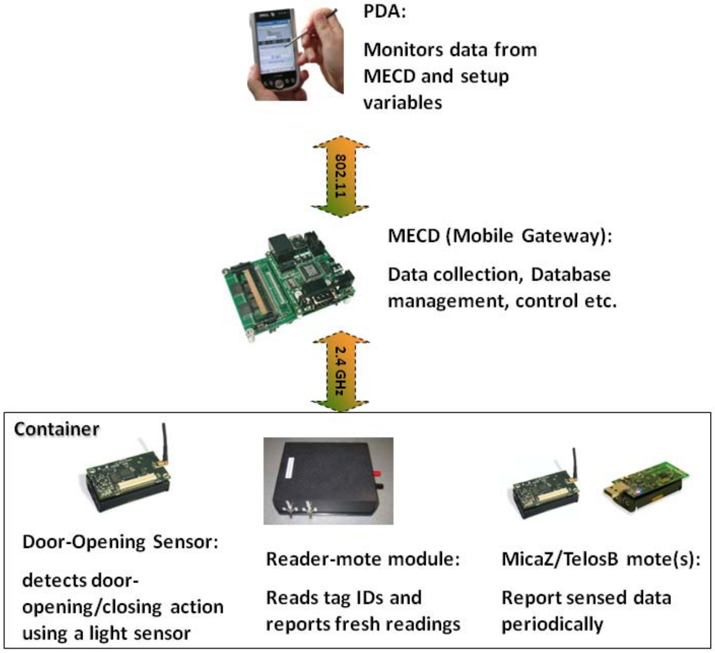

PDA:
Monitors data from MECD and setup variables
802.11
MECD (Mobile Gateway):
Data collection, Database management, control etc.
2.4 GHz
Container
Door-Opening Sensor:
detects door-opening/closing action using a light sensor
Reader-mote module:
Reads tag IDs and reports fresh readings
MicaZ/TelosB mote(s):
Report sensed data periodically

Fig. 11. Test-bed Configuration for the Intelligent Container System

were connected via 2.4-GHz Zigbee communication. Zigbee is used for networking local gateways because of its low cost, low power, and wireless mesh-networking characteristics [27]. A GUI was implemented on a PDA for remote control and monitoring. The external network between the MECD and the remote servers (PDA) used 802.11 WiFi communication.

3.4.3. Sensor Implementation

When a mote in the container is turned on, it broadcasts a beacon to find the corresponding MECD. At this implementation stage, we assumed that there was only one MECD during this initialization process, but the future implementation needs to provide a way to find an appropriate MECD if the multiple MECDs can hear this beacon message. Possible solutions are choosing the closest MECD or using a pre-deployed key to find an appropriate MECD. Once an MECD receives the beacon, it performs the handshaking process, including synchronization, database update, and ID assignment.

3.4.4. Door Opening/Closing Detection

To save battery, our intelligent container system only triggers RFID readings when the container door is open. To detect the door opening/closing action, we placed one Crossbow MicaZ mote attached to Crossbow MTS300 sensor board [28]. This sensor senses light in the container and notifies the MECD of a door opening or closing action when the value is over or under the threshold (300 in hexadecimal), respectively. When the MECD receives this event notification, it sends a *Start Reading* or *Stop Readning* command to RFID readers associated with that container. For more accurate detection, we can combine the information from light sensors with an electronic seal (E-seal) in the future. In this paper, we call it the door-opening sensor.

3.4.5. Reader-Mote Module Implementation

For the Reader-Mote module implementation, we used the SkyeTek UHF M9 reader [29], which is a small, low power, cost-efficient, and globally compliant ultra-high frequency (UHF) reader. The M9 RFID reader was integrated with the MicaZ mote to provide Zigbee communication as well as the computation capability.

We connected the RS-232 serial port of the M9 reader and the UART of the MicaZ mote through a converter cable shown in Fig. 12. This converter cable provides two-way communication and voltage conversion between the 5-V M9 RFID reader and the 3-V MicaZ mote. The MicaZ mote receives an external command, such as Setup, Start Reading, or Stop Reading, from the MECD. The mote then translates it into an appropriate command for the M9 reader and forwards a formatted command through the UART. Our Reader-Mote module was also packaged with waterproof materials as

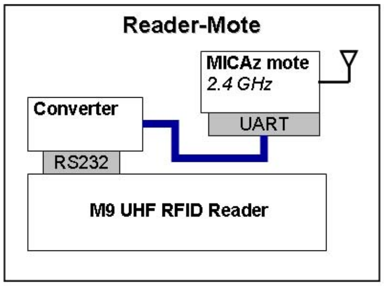


Fig. 12. Configuration of the Reader-Mote implementation.

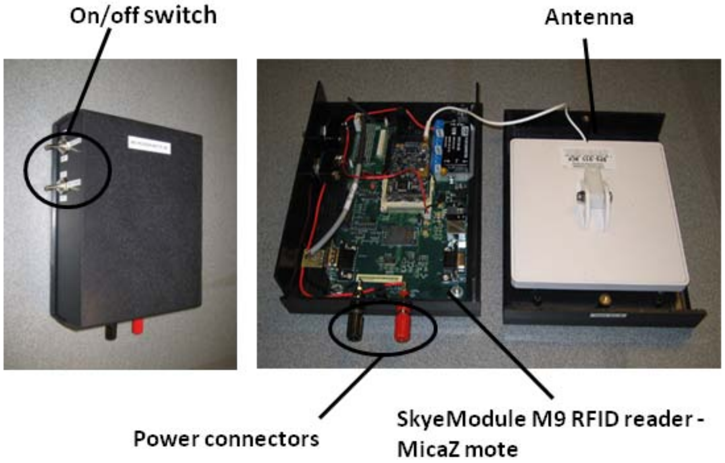


Fig. 13. Waterproof Package of the Reader-Mote implementation.

shown in Fig. 13. In the package, there is a SkyeTek M9 reader and a small antenna that is a directional 902 - 928-MHz antenna with a 6-dBi gain.

Fig. 14 depicts the operation process of the RFID Reader-Mote module. During the setup process, the administrator sends a command to change the RF frequency and the output power of the RFID reader on the basis of the current location information. This ensures compliance with regulated reading power levels and frequency bands in different countries. Table I summarizes the regulations of the power level and frequency bands [30]. The power is expressed either as EIRP (Effective Isotropic Radiated Power) or ERP (Effective Radiated Power). Here, the European regulated power level of 2

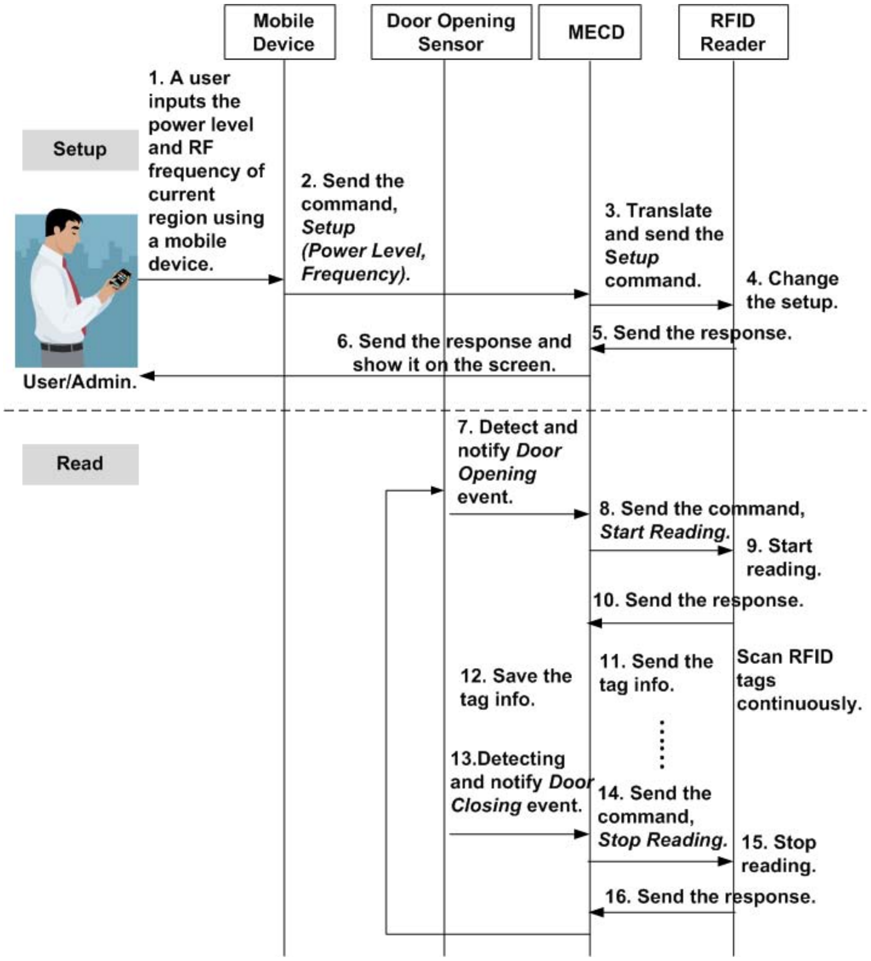

Mobile Device
Door Opening Sensor
MECD
RFID Reader
Setup
1. A user inputs the power level and RF frequency of current region using a mobile device.
2. Send the command, *Setup (Power Level, Frequency).*
3. Translate and send the *Setup* command.
4. Change the setup.
5. Send the response.
6. Send the response and show it on the screen.
User/Admin.
Read
7. Detect and notify *Door Opening* event.
8. Send the command, *Start Reading.*
9. Start reading.
10. Send the response.
11. Send the tag info.
Scan RFID tags continuously.
12. Save the tag info.
13.Detecting and notify *Door Closing* event.
14. Send the command, *Stop Reading.*
15. Stop reading.
16. Send the response.

Fig. 14. Process Flow of RFID Readings.

Table I. Government regulations for UHF RFID.

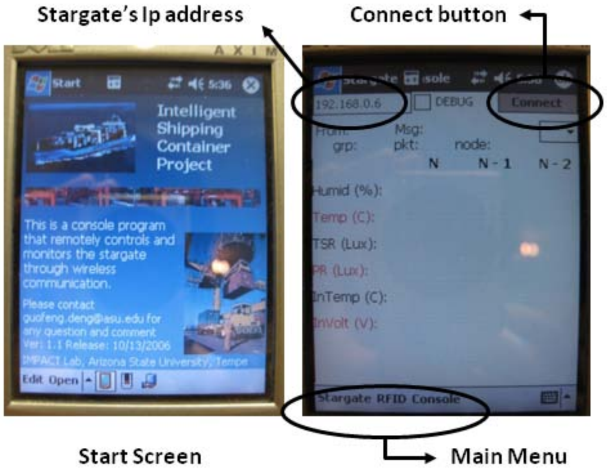
Region	Power Levels	Frequency Bands
Singapore	0.5 Watts ERP	866-869 MHz
Taiwan	0.5 Watts ERP	922-928 MHz
Philippines	0.5 Watts ERP	918-920 MHz
Europe, South Africa	2 Watts ERP	865.6-867.6 MHz
China	2 Watts ERP	840.5-844.5 MHz
U.S	4 Watts EIRP	902-928 MHz
Australia	4 Watts EIRP	920-926 MHz
New Zealand	4 Watts EIRP	864-868 MHz
Japan	4 Watts EIRP	952-954 MHz
South Korea	4 Watts EIRP	908.5-910 MHz

Watts ERP is equivalent to 3.28 Watts EIRP [31]. Ideally, we expect this process to be automatic, and the future MECD implementation will use a GPS to obtain the current location.

As mentioned in Section 3.4.4, the MECD sends *Start Reading* or *Stop Reading* commands to the Reader-Mote module when it receives the message of *Door Opening* or *Door Closing* events from the sensor attached on the container door. After a reading is triggered, the M9 RFID reader immediately passes the RFID Tag information to the MicaZ mote without any freshness checking. On the MicaZ mote of the Reader-Mote module, the duplicated IDs are eliminated, and only fresh readings are sent out to the MECD.

3.4.6. GUI Implementation

For the remote control, we implemented a GUI program called SGConsole using Visual C#. Fig. 15 is the display of the SGConsole program running on a PDA.



This is a console program that remotely controls and monitors the stargate through wireless communication.

Please contact guofeng.deng@asu.edu for any question and comment
Ver: 1.1 Release: 10/13/2006
IMPACT Lab, Arizona State University, Tempe

Fig. 15. PDA's GUI for Remote Control.

To access a particular container, a user types the IP address of the MECD and then presses the *Connect* button on the upper right corner of the screen. Once the connection is established, data are displayed on the screen for remote monitoring.

At the bottom, there is the main menu: *Stargate*, *RFID*, and *Console*. The Stargate menu includes the sub-menus: *Set Time*, *Shutdown*, and *Status*. When a Stargate (MECD) is turned on, the time is initialized by the *Set Time* sub-menu. The PDA transmits the timestamp to the MECD and then receives feedback from the MECD. The *Shutdown* sub-menu is used for shutting down the Stargate gracefully. The *Status* sub-menu displays its current time, running processes, and free disk space. As mentioned, the Reader-Mote module is turned on by a door-opening event. However, a user can also turn it on manually by the *RFID* menu. It also supports to refresh the cache in the Reader-Mote module. The *Console* menu is used for clearing the screen and exit.

3.4.7. Deployment

The test-bed was tested first in a stand-alone container over several months and then in a 33 stacked container configuration to assess the inter-network interference

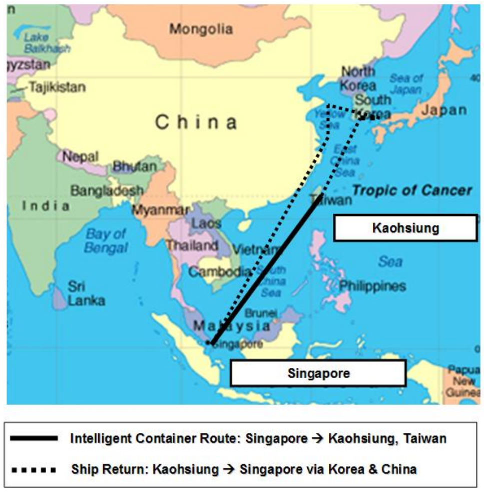


Fig. 16. Route for the Test from Singapore to Taiwan.

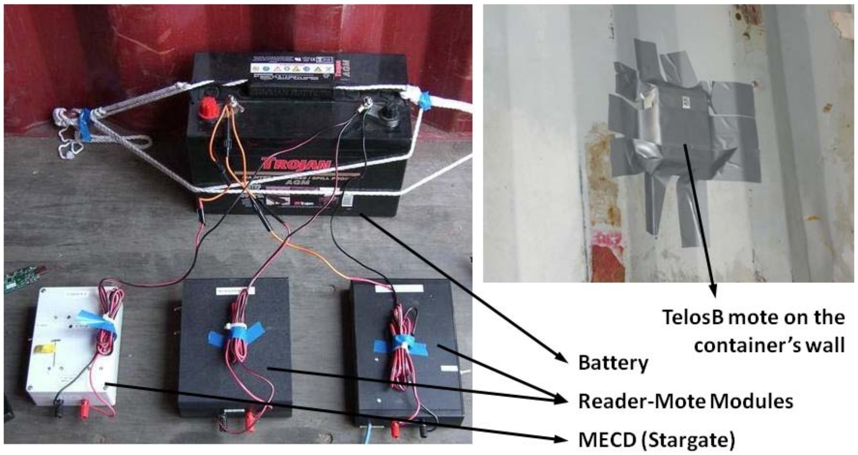


Fig. 17. Actual Deployment in the Container.

and network communication strength. Section A.2 presents the experimental results in details. Finally, we deployed our system for a multi-day shipment from Singapore to Taiwan shown in Fig. 16.

Fig. 17 shows the installation of our equipment in a container. For this test, an administrator sets the power level and frequency to the regulation of Singapore or Taiwan using the PDA before opening a container door. The regulated power level is 0.5 W ERP in both countries. The frequency is 866-869 MHz in Singapore and 922-928 MHz in Taiwan. With 0.5 W ERP, the average read range of our Reader-Mote module was approximately 18 in. Therefore, we need to use multiple Reader-Mote modules to expand the coverage. Each Reader-Mote module sends fresh readings to MECD, but there is a chance of multiple readers reading the same RFID tag. To avoid duplication, the MECD will perform a freshness check again.

Before we deployed our system, we examined the energy consumption of the Stargate and the RFID reader. The Stargate with MicaZ and AmbiCom CF WiFi card attached drew 406 mA of current, consuming a power of 1827 mW. The M9 RFID reader on single reading drew 270 mA of current, consuming a power of 1215 mW, while a continuous reading mode drew 615 mA of current, consuming a power of 2767.50 mW. To support the shipment from Singapore to Taiwan, the Stargate (MECD) and Reader-Mote modules were powered by a car-size battery. These modules were mounted on the container's wall, similar to the TelosB mote in Fig. 17.

Four TelosB motes were mounted on the four corners of the container; they measured temperature and humidity every 5 min. The MECD stored all the collected data such as temperature, humidity, and signal strength in its local database; these data were dumped from the flash memory after shipment.

We successfully tested our single container configuration. However, we found that the connectivity between a gateway and a server can be very low because of the physical constraints (i.e. due to high interference from metal and dense materials in and around the containers). In addition, the high temperature reduces the RF (radio frequency) signal strength. Fig. 18 shows that the signal strength of two nodes in a container decreases with an increase in the temperature. As shown in Fig. 18, the variation of temperature is high, approximately 20 degrees Celsius within a day. Therefore, we need

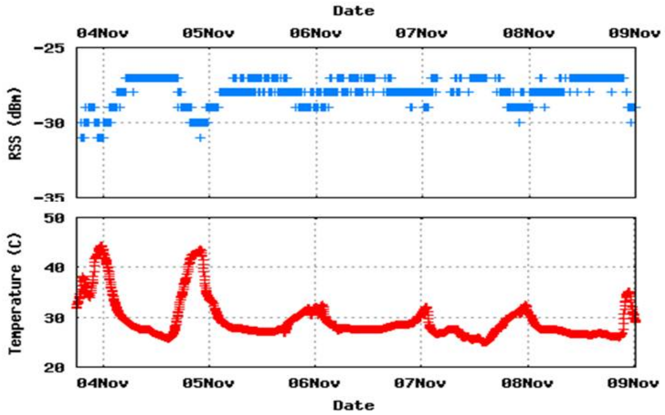


Fig. 18. Temperature and Signal Strength Changes during Shipment.

to consider these two issues for the system design of intelligent containers.

3.5. Related Work

In this section, we present a brief discussion on the recent works on mobile gateways.

Chen and Ma [32] addressed the problems of the static gateways in a wireless sensor network. One of the key problems is that sensor nodes close to the gateway will die quickly because they forward data from other nodes to the gateway. Further, the network connectivity may not be guaranteed because of obstacles or sensor node failures. To solve these problems, they have introduced an additional layer of mobile nodes such as mobile phones, PDAs, and laptops with multiple communication capabilities.

Recently, researchers have paid attention to the mobile gateway, but the mobility introduces new challenges such as re-routing and energy efficiency. In [33], Akkaya and Younis have proposed an energy-aware routing scheme for mobile gateways. This scheme aims to reduce the frequency of re-routing which in turn would reduce the overhead. In this scheme, re-routing is triggered only when the distance of the gateway from the last hop nodes becomes unacceptably large. Otherwise, the last hop nodes increase their radio ranges to reach the gateway.

Shakya et al. have focused on maximizing the lifetime of a sensor network [34]. To balance the lifetime among nodes, they have proposed the distribution of the role of forwarders among all N-hop neighbors of the gateway. As the gateway moves, it performs a local reconfiguration of an N-hop neighborhood. Because of the local reconfiguration, the re-routing cost can be reduced.

Many researchers have investigated various mechanisms to enhance energy efficiency and network connectivity for a wireless sensor network, but we will focus on the needs of mobile gateways for a USN to handle a tremendous amount of data generated by a huge number of sensor nodes.

4. MESH-NETWORKED MECDS

The gateway is responsible for connecting the end nodes to the remote server in sensor networks. However, the mobile gateway in USN has a problem to communicate with the server in some cases. For example, as shown in Fig. 19, containers are stacked very closely and thus a container stacked at the bottom will not have a direct connection to the server because of the physical constraints (i.e. due to high interference from metal and dense materials in and around the containers). Moreover, containers can be in a harsh environment (e.g., sea and desert) and Fig. 18 in Section 3.4.7 shows intense temperature changes during shipment over sea. High temperature reduces signal strength of RF signals and consequently will have a negative effect on connectivity between a gateway and a server.

In order to solve this problem of *unreachability*, we propose the creation of a *dynamic mesh network* among neighboring gateways so that they can collaborate in sending sensed data to the server. This mesh-networking approach can support reliable connectivity by redundant paths between the nodes.

4.1. System Design

This section discusses our system design and the system requirements of this mesh networking approach.

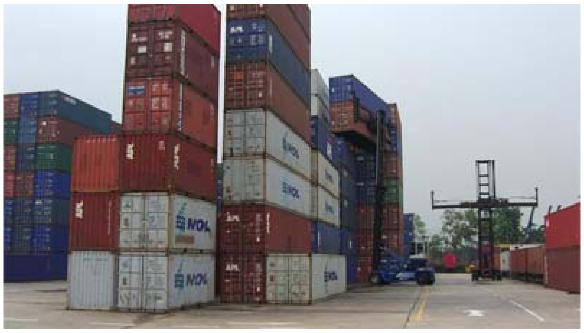


Fig. 19. Cargo Containers stacked in a Container Yard.

4.1.1. System Architecture

This section presents our *hierarchical* system architecture that is designed to support different levels of collaborations in an efficient way. To design USN applications, we must consider the following requirements.

- *Scalability*: In USN, there are a considerable number of devices especially end nodes. The system should be able to handle an increasing amount of data generated by a huge number of end nodes.
- *Reliability*: The system can be deployed in harsh environments and therefore nodes or links can fail. In such harsh environments, however, an end node should be able to send a message to a server even though there are some node or link failures. In other words, end-to-end communication should be reliable.

Under these requirements, we design a 3-level hierarchical network as shown in Fig. 20.

The three levels are as follows:

- *Server-level*: Servers collect data from end nodes and send query commands to them through MECDs. Servers are connected so that they can collaborate on the basis of the interests or needs of applications.
- *MECD-level*: An MECD performs two types of operations. First, the MECD gathers data sensed by end nodes and send high-level data to the server. On the other hand, the MECD receives a command from the server, interprets it, and sends a formatted command to the end nodes. In order to support these operations efficiently, MECDs create a dynamic mesh network with neighboring MECDs and collaborate via this network.

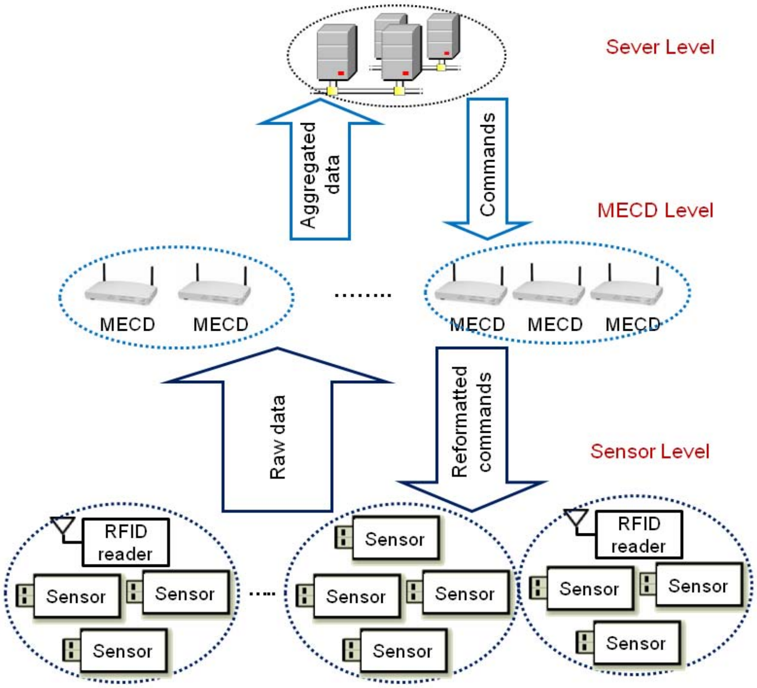


Fig. 20. Hierarchical Network Structure.

- *Sensor-level*: End nodes can be sensors and RFID devices with wireless communication capability. They monitor surroundings or objects and send sensing data to the MECD.

This 3-level network is designed to support multi-level collaboration. Because all networks are managed in a distributed manner, this hierarchical structure can reduce the effects of the growth in one network on other level networks. Further, the system is *efficient* in terms of bandwidth and energy because MECDs can reduce the amount of data to be transmitted to the server. The system can be easily enlarged by adding a few MECDs although the number of end nodes increases considerably. Therefore, this hierarchical architecture can deliver *scalability* to the system. In our architecture, sensor-level and MECD-level networks use mesh networking. Because of the redundant paths in mesh networks, *reliability* can be improved. In addition, the MECD has multiple communication capabilities such as Zigbee, Bluetooth, WiFi, satellite, and cellular networks to communicate with the server. It selects one of the available networks *dynamically* on the basis of not only the system requirements (e.g., transmission range and bandwidth) but also the network conditions. Such alternative networks or paths can help *reliable* communication from an end node to a server.

4.1.2. System Requirements of MECD-level Networks

In this section, we will focus on the MECD-level network proposed in this paper. The requirements of the MECD-level network are as follows:

- *Server reachability*: As mentioned at the beginning of Chapter 4, some MECDs (e.g., the container's MECD stacked at the bottom) are not able to communicate with the server directly. The MECD-level network is used for helping communication between the server and MECDs that do not have a direct connection. Therefore, this MECD-level network must support the ability to reach the server from any MECD in the system.
- *Network latency*: In this study, our targeted applications are real-time tracking and monitoring applications. However, this MECD-level network can introduce additional delay to transmit a message via multi-hop routes. The increase in the delay created by the MECD-level network should be acceptable.
- *Energy efficiency*: Similarly, MECDs will consume additional energy to route messages sent by other neighboring MECDs. Since MECDs have limited energy, the total power consumption of this network is low.

In order to meet these requirements, there are issues that need to be considered:

- *Network size/density*: The size and density of MECD-level networks depend on applications and situations. Because the network size and density significantly affect the network latency and energy consumption, they are important factors with respect to the evaluation of the MECD-level network.

- *Temperature*: Bannister *et al.* [35] found that the signal strength between two nodes decreases as the temperature increases. The signal strength affects the server reachability, network latency, and energy consumption of the network. Therefore, the temperature condition of the real environments should be considered.

4.2. Evaluation

In Section 3.4.7, we showed that our system was successfully tested in a single container configuration. However, testing the MECD-level network in an actual environment is challenging because there are many factors to be considered. First, the density of MECD-level networks depends on the size of the cargo containers. The network density has a significant effect on connectivity between two MECDs. Therefore, to evaluate the server reachability, we need to consider different container sizes. Further, the size of the MECD-level networks depends on the capacity of ships, trains, and container yards. Obviously, a large network size increases the network latency as well as the total energy consumption of the MECD-level network. Finally, the temperature that affects the server reachability, network latency, and total energy consumption varies considerably during the shipment.

In this section, we will discuss simulation for the MECD-level network with different container sizes, network sizes, and temperatures.

4.2.1. Simulation Setup

To set up the network of containers [36], we use the International Standards Organization (ISO) standards. There are five common ISO standard lengths: 20 ft. (6.1 m),

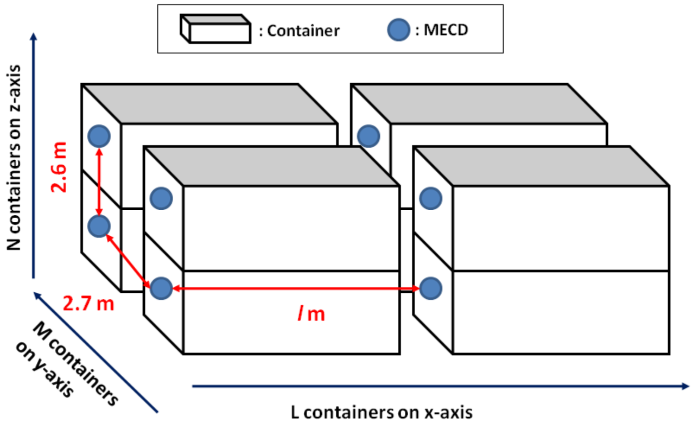


Fig. 21. Simulation Setup for ISO Containers.

40 ft. (12.2 m), 45 ft. (13.7 m), 48 ft. (14.6 m), and 53 ft. (16.2 m) [36]. The height of the containers varies from 4.25 ft. to 9.5 ft. However, the typical container height is 8.5 ft. (2.6 m) [37] [38]. The standard width of containers used in international commerce is 8 ft. (2.44 m) [37] [38].

Fig. 21 shows the network setup for the simulation. We assume that the MECD is attached at the center of each container door. In this simulation, we use the fixed values of 8 ft. for the container width and 8.5 ft. for the container height. Typically, the space between the containers is less than 1 ft. [39]. Therefore, we assume that the distances between the MECDs are 2.7 m (approximately, 8.86 ft. = 8 ft. wide + 0.86 ft. space) in width and 2.6 m in height. For the length, we consider 5 common ISO standard lengths and l denotes the distance between the MECDs in length (the container length + the space) in meters.

L, M, and N are the number of containers on the x-axis, y-axis, and z-axis, respectively. Typically, up to six shipping containers are stacked on top of one another [39]. Therefore, N is fixed as 6 for simplicity. L and M depend on the environment where the containers are stacked (e.g., container yard, ship, train, and port).

4.2.2. Communication Model

In this simulation, we assume that all MECDs have same communication and processing capability. One MECD can directly communicate with other MECDs within the maximum communication range, but it cannot directly communicate with MECDs out of this range.

First, we estimate the maximum communication range between two MECDs using the *Log-Distance Path Loss* model. The Log-Distance Path Loss model is a popular radio propagation model to predict the path loss on a link with respect to distance and environment [40]. The received power at distance d is expressed as follows:

$$P_r(d) = P_r(0) - 10n_p \log(d/d_0) + X_\sigma, \tag{4.1}$$

where $P_r(0)$ is the received power measured at the reference distance d_0 and n_p is the path loss exponent. X_σ is a zero-mean normally distributed random variable with standard deviation σ.

The temperature loss in dB is defined as follows:

$$T_L(T) = 0.1996(T - 25), \tag{4.2}$$

where T is the temperature in Celsius with the range 25°C $\leq$ T $\leq$ 65°C [35]. Using Equation 4.1 and 4.2, we can define the maximum communication range as the maximum value d that satisfies the inequality:

$$P_r(d) - T_L(T) \geq P_s, \tag{4.3}$$

where P_s is the radio sensitivity [35].

In this simulation, we use d_0 = 1 m, $P_r(0)$ = -45 dBm, σ = 3 dBm, and P_s = -94 dBm that are widely used for wireless sensor networks [35]. The parameter, n_p, depends on the environment, and we use the value of 3.3 for metal materials of cargo containers [41].

4.2.3. Energy Consumption Model

The energy to transmit a 1-bit message to a distance of 100 m has been found to be equal to the energy required to process 3000 instructions [42]. Therefore, in this simulation, the energy required for processing data is mostly ignored, and only the energy consumption for communication is considered.

We use a simple radio model called the *first-order radio model* to compute the total energy consumption of packet transmission and reception over distance [43]. In this model, it is assumed that the energy consumption to run the transmitter circuitry, $E_{Tx-elec}$, and receiver circuitry, $E_{Rx-elec}$, are the same. When the distance is d, a d^2 energy loss is assumed for channel transmission. The energy consumption in transmitting a k-bit data packet to a distance d is defined as follows:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2, \tag{4.4}$$

where E_{elec} is the energy consumption to run the transmitter or receiver circuitry and ε_{amp} is the energy consumed by the transmitting amplifier. We used E_{elec} = 50 nJ/bit and ε_{amp} = 100 pJ/nit/m^2; these are the widely used values for wireless networks [42] [43].

The energy consumption in receiving a k-bit data packet is defined as follows:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} \times k. \tag{4.5}$$

Using equations 4.4 and 4.5, we define the total energy consumption to transmit a k-bit data packet over n hops:

$$E_{total}(k,n) = n \times (E_{Tx}(k,d) + E_{Rx}(k)) = n \times (2k \times E_{elec} + \varepsilon_{amp} \times k \times d^2). \quad (4.6)$$

4.2.4. Metrics and Measurement Methods

In this section, we will discuss how to measure the server reachability, network latency, and energy efficiency.

In the case of the intelligent container system, the container at the top of the stack has a clear line-of-sight. Therefore, we assume that the MECD located on the top corner can communicate with the server directly. This MECD called the *forwarder* is responsible for forwarding the packets from the other MECDs to the server.

1. *Server Reachability*

   In the simulation, we first compute the location of each MECD with a given length (l). Using Equations 4.1, 4.2, and 4.3, we calculate the maximum communication range with the given network density and temperatures. On the basis of that range, we determine whether an MECD has a direct link with other MECDs. In other words, we determine whether two MECDs are within that range.

   Using the link information, we compute the connectivity and the server reachability for all MECDs in the network.

   Definition 4.2.1 (Connectivity (C_i)) *The connectivity represents the ability to send a packet from a specific MECD to the forwarder. The connectivity of MECD i, C_i is 1 if there is at least one possible path between MECD i and the forwarder. Otherwise, C_i is 0. Note that a path can be multiple hops.*

Definition 4.2.2 (Server Reachability(SR)) *The server reachability, SR, denotes the ratio of the number of MECDs that can reach the server through the MECD-level network to the total number of MECDs in the network. The server reachability, SR, is defined as follows:*

$$SR = \frac{\sum_{i \in S} Ci}{L \times M \times N} \times 100, \tag{4.7}$$

where S is the set of MECDs in the network. Note that $L \times M \times N$ is the total number of MECDs in the network.

As discussed in Section 4.2, the temperature and the network density have significant effects on server reachability. We use T = 25°C, 45°C, and 65°C considering temperature changes during the actual shipment. In the intelligent container scenario, the network density changes with a change in the distance between the MECDs with respect to the length, l. Considering the ISO container lengths and the space between MECDs, we use the range of 5 m $\leq l \leq$ 30 m. However, the network density remains same with a change in the total number of MECDs. Therefore, L, M, and N are fixed as 6 for calculating SR.

2. *Network Latency*

   To evaluate network latency, we measure the *average path length* at different values of T, $L \times M \times N$, and l. In this simulation, the average path length is defined as the average number of hops along the shortest paths between the forwarder and all other MECDs in the network. The average path length is an important factor to evaluate the performance of the MECD-level networks because the network latency can be expressed as a function of the path length.

On the basis of the link information between MECDs, we find the shortest paths between the forwarder and all the other MECDs and then calculate the average path length.

As mentioned in Section 4.2.1, N is fixed as 6 for simplicity. L and M are changed from 6 to 20 to see the effects of these changes on the average path length at T = 25°C, 45°C, and 65°C. Since 45 ft. long containers are not commonly used in international commerce [37] [38], we fix l = 6.4 m, 12.5 m, 14.94 m, and 16.46 m (20 ft., 40 ft., 48 ft. and 53 ft. with 1 ft. space).

3. *Energy Efficiency*

   To evaluate energy efficiency, we measure the total energy consumption at different values of T, $L \times M \times N$, and l. The values of T, $L \times M \times N$, and l are same as the simulation of the network latency discussed above.

   In this simulation, we assume that every node in the network sends a packet to the forwarder via its shortest path. The packet size, $k = 20$ bytes is the same as that of our test-bed system. Considering the values of parameters in Section 4.2.2, we estimate the ideal maximum transmission range to be approximately 50 m in IEEE 802.15.4 Zigbee [44]. Therefore, we use $d = 50$ m in equation 4.6.

   Even though the *ideal* maximum transmission range at these parameters is approximately 50 m, the *actual* maximum communication range can be reduced by increasing the temperature. We have already calculated the actual maximum communication range and the shortest path lengths from each node to the forwarder at T = 25°C, 45°C, and 65°C for the simulation of network latency discussed above. Therefore, we use $n =$ the shortest path length of each node for 4.6. Using 4.6, we

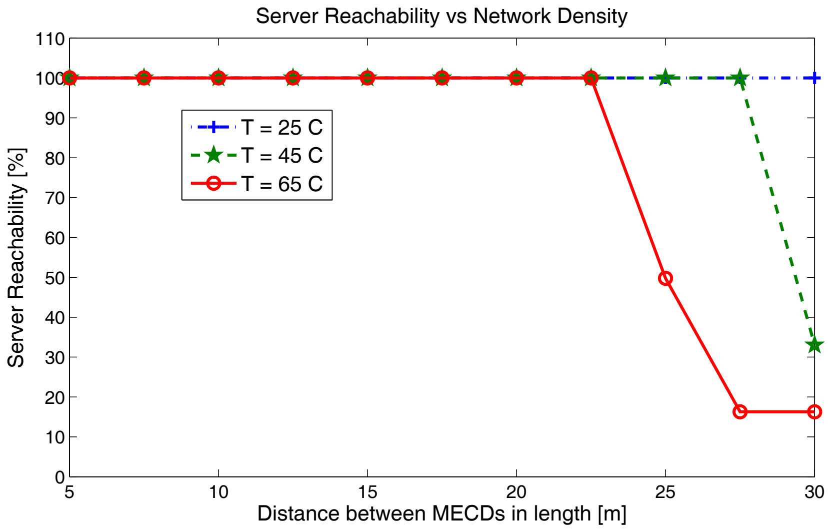


Fig. 22. Server Reachability with Different Distances between MECDs in Length (l) and Temperature (T).

compute E_{total} for every MECD and sum them for the total energy consumption of the MECD-level network.

4.2.5. Simulation Results

Fig. 22 shows SR with the range 5 m $\leq l \leq$ 30 m and T = 25°C, 45°C, 65°C. At T = 25°C, SR is 100%, but at T = 45°C and 65°C, SR is less than 100% after a certain distance ($l \geq$ 28 m with T = 45°C and $l \geq$ 22.5 m with T = 65°C). However, ISO standard container lengths with 1 ft. space are less than 16.5 m. Therefore, we can conclude that *the MECD-level network can provide 100% server reachability for ISO standard containers within the range 25°C $\leq$ T $\leq$ 65°C when the containers are closely stacked*.

For the simulation in this dissertation, we set the path loss exponent, n_p to the value of 3.3 for metal materials as mentioned in Section 4.2.2. However, in order to see the effects of n_p values on SR, we calculated SR with different n_p values and T =

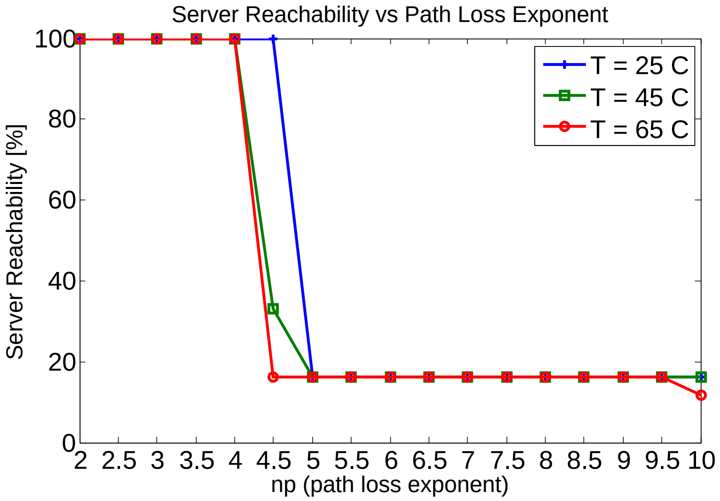


Fig. 23. Server Reachability with Different Path Loss Exponents (n_p) and Temperatures (T) when Distance between MECDs in Length (l) = 16.46m.

25°C, 45°C, 65°C. This simulation used l = 16.49m as the longest length of common ISO containers. Fig. 23 show that SR becomes less than 100% when $n_p \geq 4.5$. However, n_p is 2 for the free space and typically between 3 and 4 [45]. In conclusion, the complete server reachability for ISO standard containers within the range 25°C ≤ T ≤ 65°C can be achieved.

Fig. 24, 25, and 26 present the average path lengths with different values of $L \times M \times N$ and l when T is 25°C, 45°C, or 65°C, respectively. From these figures, we can see that the average path length gradually increases when the number of MECDs increases considerably. Further, to conclude, *the MECD-level network will produce a small amount of additional delays and is scalable to a large size of MECD networks.*

Fig. 27, 28, and 29 show the total energy consumption at different values of $L \times M \times N$ and l and l when T is 25°C, 45°C, or 65°C, respectively. From these figures, we can see that the total energy consumption gradually increases when the number of MECDs increases considerably. Further, to conclude, *the MECD-level network will consume a*

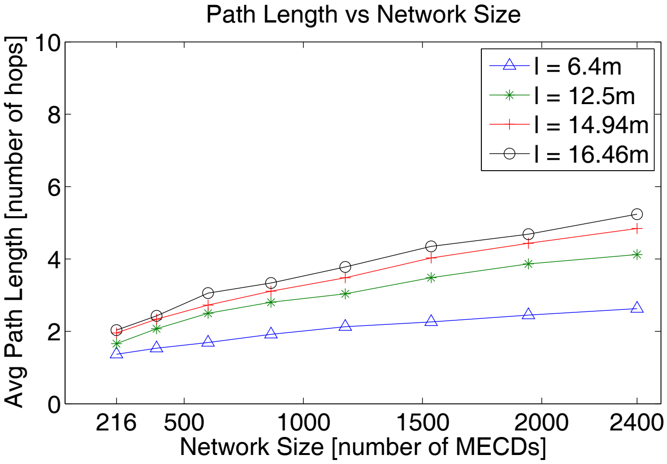


Fig. 24. Average Path Length with Temperature (T) = 25°C, Different Network Sizes ($L \times M \times N$) and Distances between MECDs in Length (l).

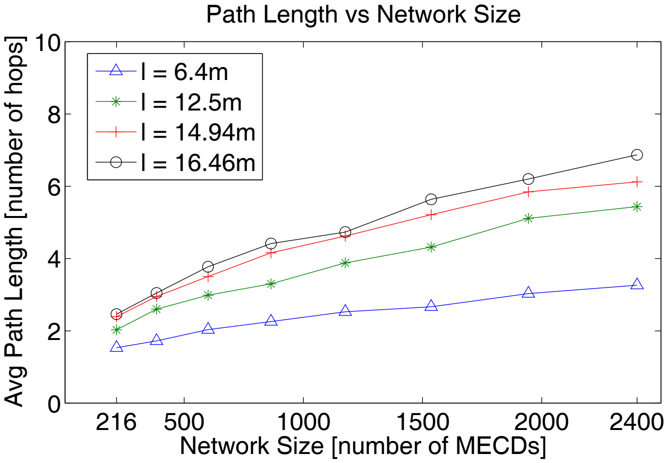


Fig. 25. Average Path Length with Temperature (T) = 45°C, Different Network Sizes ($L \times M \times N$) and Distances between MECDs in Length (l).

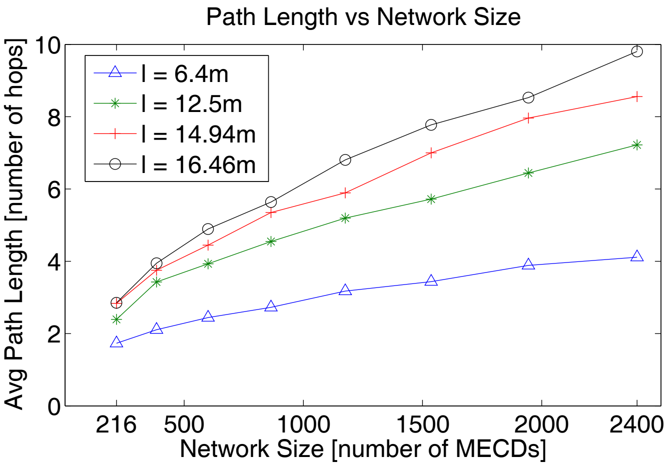


Fig. 26. Average Path Length with Temperature (T) = 65°C, Different Network Sizes ($L \times M \times N$) and Distances between MECDs in Length (l).

small amount of additional power and is scalable to a large size of MECD networks.

For the simulation above, we used $Pr(0)$ = -45 dBm. For MicaZ motes, the received signal power at distance 1m, $Pr(0)$ can be calculated as follows [46]:

$$P_r(0) = P_t - 40.2, \tag{4.8}$$

where P_t is the transmission power in dBm. Because the transmission power of MicaZ is between 0 dBm and -25 dBm [25], -65.2 dBm $\leq Pr(0) \leq$ -40.2 dBm.

This dissertation used $Pr(0)$ = -45 dBm which is almost the maximum transmission power of MicaZ for the simulation because this number is a typical value for wireless sensor networks. However, from the results shown in Fig. 24, 25, and 26, we found that the average path lengths were usually small. Especially, with the small distance and network size, the average path lengths were about 1. That means the most of MECDs are within the communication range of the forwarder with $Pr(0)$ = -45 dBm.

Due to this observation, we found that the maximum transmission power may not need for the complete SR. Therefore, we tried to reduce the transmission power

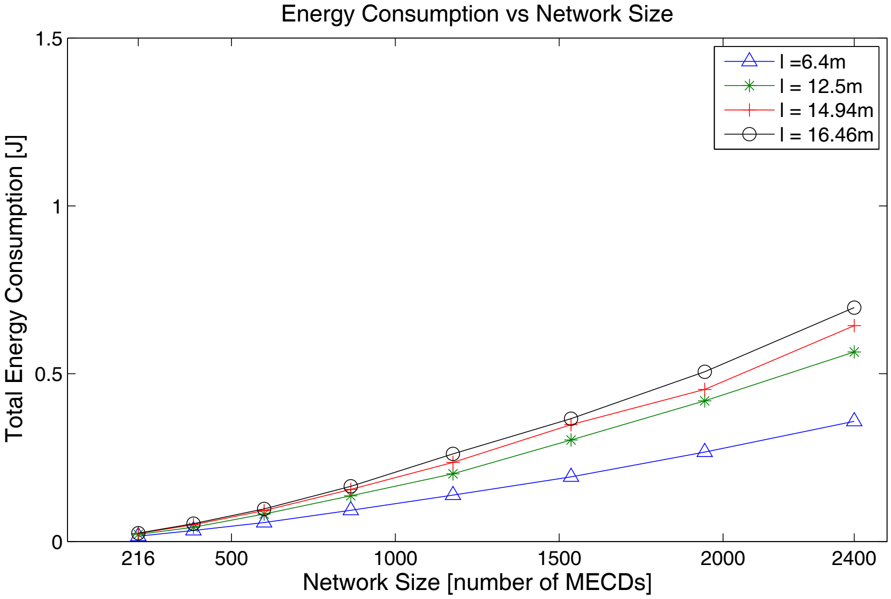


Fig. 27. Total Energy Consumption with Temperature (T) = 25°C, Different Network Sizes ($L \times M \times N$) and Distances between MECDs in Length (l).

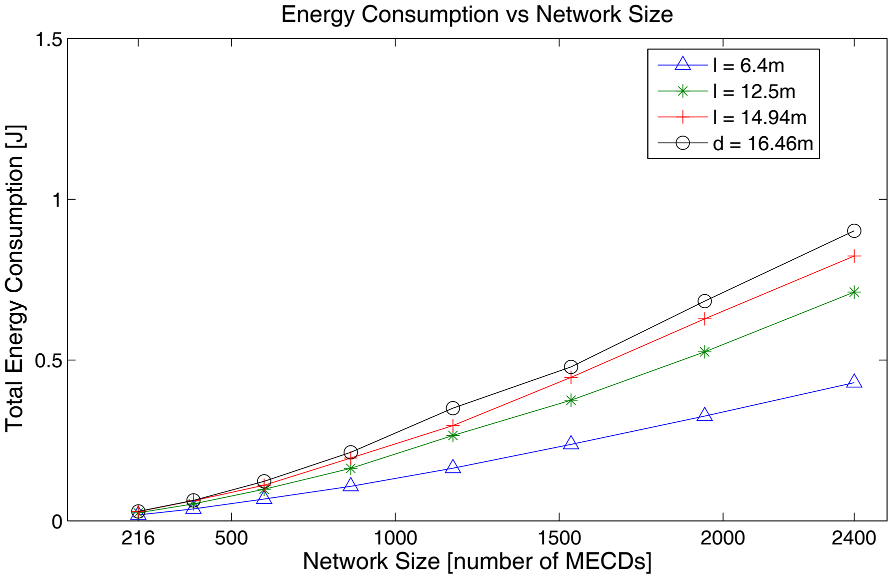


Fig. 28. Total Energy Consumption with Temperature (T) = 45°C, Different Network Sizes ($L \times M \times N$) and Distances between MECDs in Length (l).

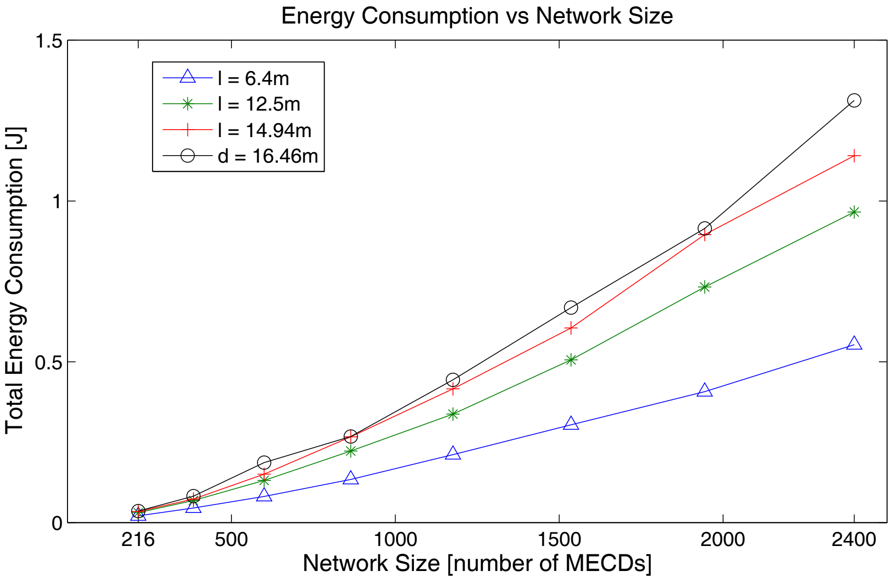


Fig. 29. Total Energy Consumption with Temperature (T) = 65°C, Different Network Sizes $(L \times M \times N)$ and Distances between MECDs in Length (l).

of MECDs to communicate with the forwarder while SR is still 100%. We calculated SR with every 2.5 dBm of $Pr(0)$ from -65.2 dBm to -40.2 dBm. Fig. 30 shows the minimum transmssion power to support the complete SR with different temperatures and distances between MECDs in length. The results indicate that 100% SR can be achieved with a lower transmission power than the maximum transmission power. Specifically, when $l =$ 6.4 m, the minimum transmission power of MicaZ is enough to support the complete SR for 25°C $\leq$ T $\leq$ 65°C. Even with $l = 16.46$ m and T = 65°C (the worst case), -12.5 dBm can support 100% SR and the maximum transmission power (0 dBm) is not needed.

However, there is a trade-off between the energy consumption and the network latency. To see the effects of the transmission power of one node on the network latency, we calculated the average path legnths for different transmission powers with T = 25 °C, 45 °C, and 65 °C and different container lenghts ($l = 6.4$ m, 12.5 m, 14.96 m, and 16.46 m). Fig. 31, 32, 33, and 34 show the results when the number of MECDs in the

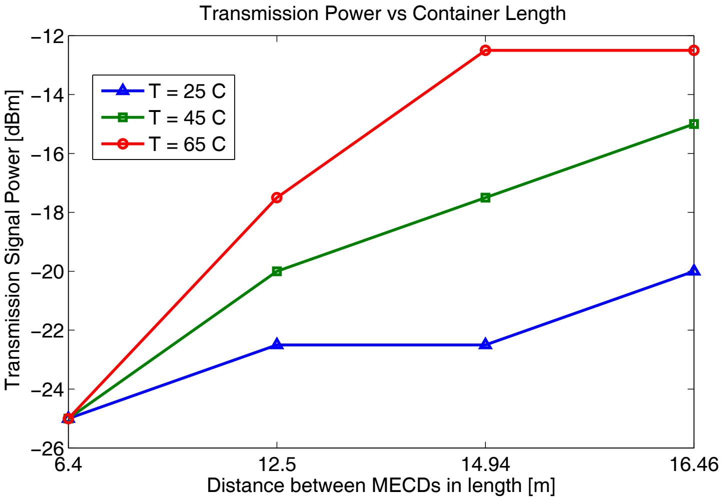


Fig. 30. Minimum Transmission Power for Complete Server Reachability with Different Temperatures (T) and Distances between MECDs in Length (l).

network is 216. As shown in these figures, the average path lengths could increase as the transmission power decreases.

Although a sender can save the energy to transmite a packet by decreasing the transmission power, the overall energy consumption can increase. This is because the intermediate MECDs spend additional transmission and reception power. Fig. 35, 36, 37, and 38 show the trend in the total energy consumption with the changes of the transmission power. As shown in these figures, the total energy consumption decreases as the transmission power decreases until the certain transmission power level (e.g., -20 dBm with T = 25 °C and $l = 6.4$ m). However, after that level, the total energy consumption starts increasing even though the transmission power decreases. Therefore, we need to dynamically adjust the transmission power level based on the temperature and container lengths.

4.3. Related Work

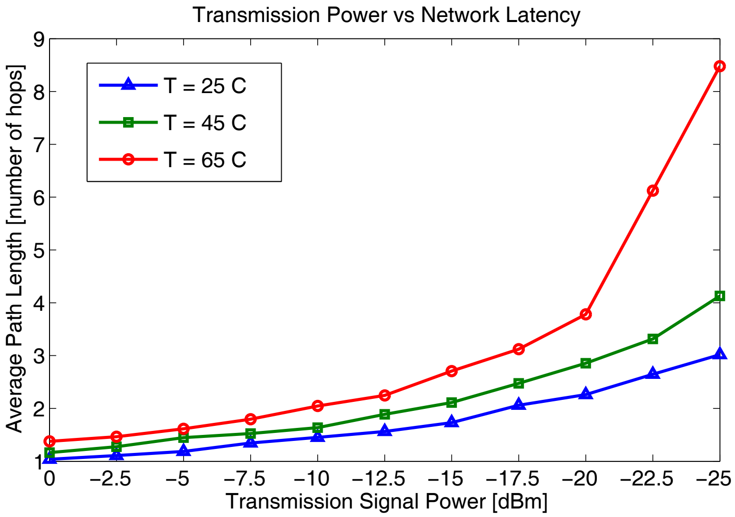


Fig. 31. Average Path Length with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 6.4 m.

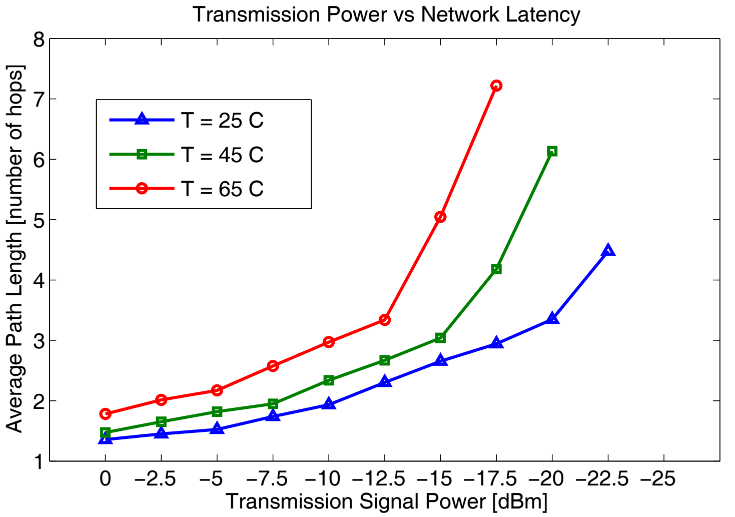


Fig. 32. Average Path Length with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 12.5 m.

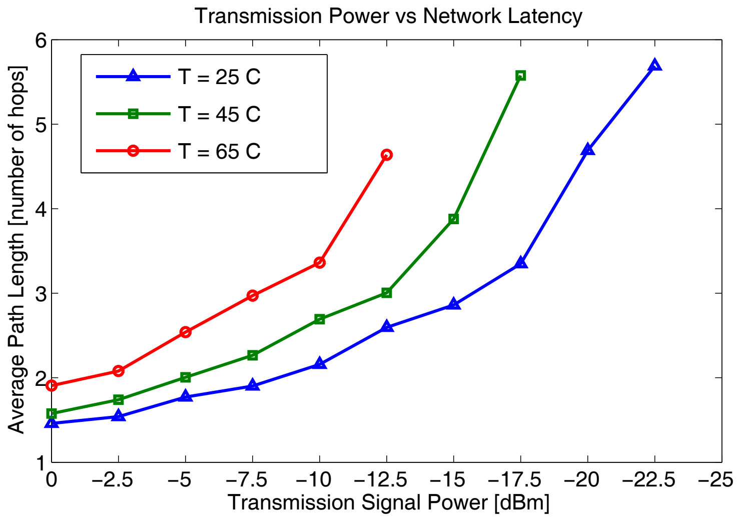


Fig. 33. Average Path Length with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 14.94 m.

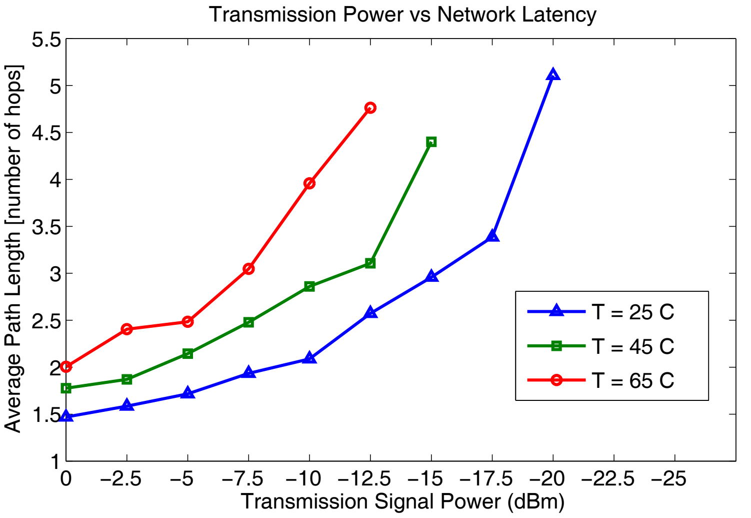


Fig. 34. Average Path Length with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 16.46 m.

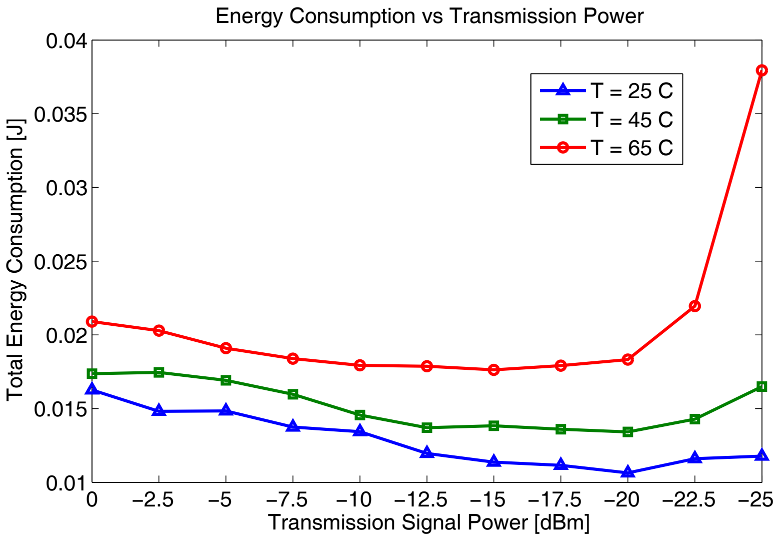


Fig. 35. Total Energy Consumption with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 6.4 m.

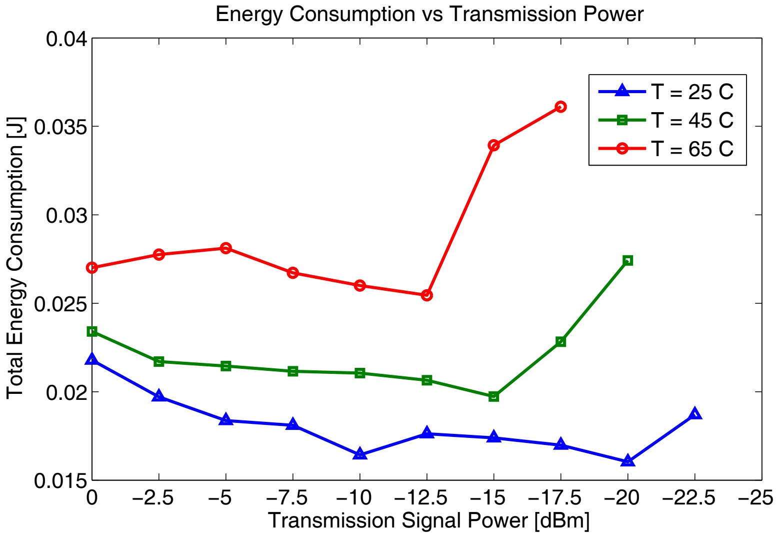


Fig. 36. Total Energy Consumption with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 12.5 m.

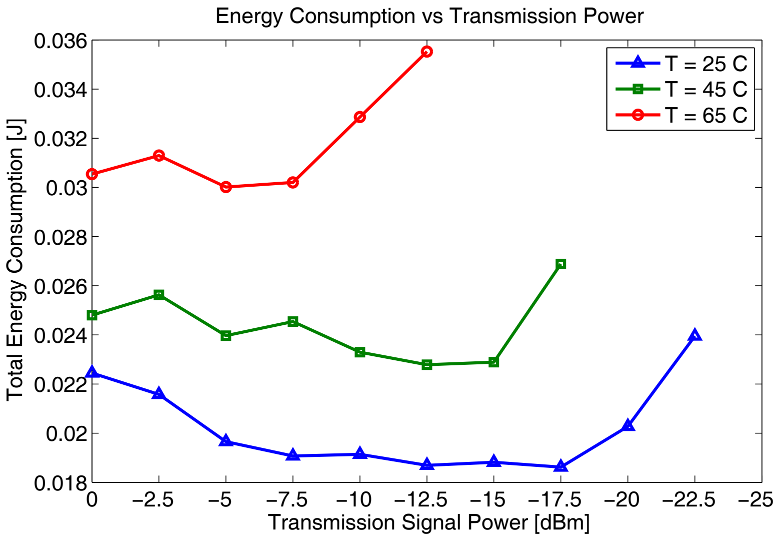


Fig. 37. Total Energy Consumption with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 14.94 m.

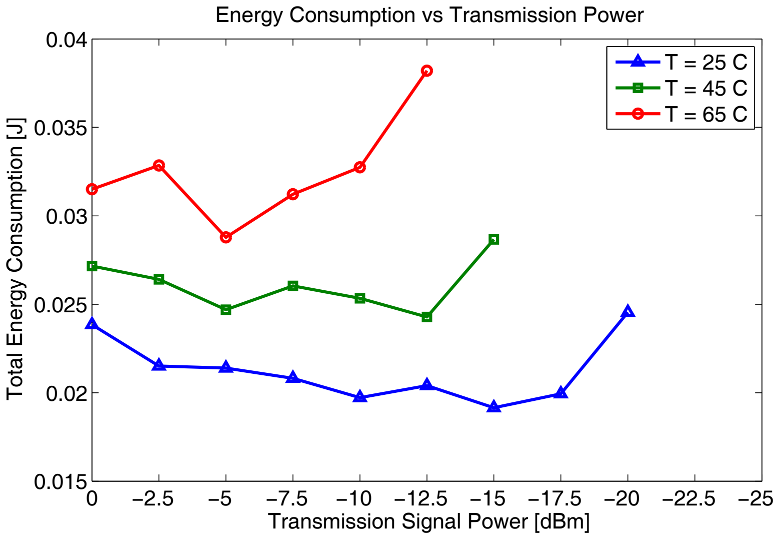


Fig. 38. Total Energy Consumption with Different Temperatures (T) and Transmission Powers when Distances between MECDs in Length (l) = 16.46 m.

In this section, we will briefly discuss the existing intelligent container systems targeted in our test and simulation.

Today, there are more than 20 million cargo containers moving around the world each day [47]. Cargo containers transport 90% of the world's trade and more than 10 million cargo containers enter U.S. ports each year. However, some reports have stated that only 5 percent can be inspected with today's capability [48]. Since the events of 9/11, numerous government regulations, initiatives, and mandates have emerged for security of cargo containers, including: Automated Targeting System (ATS) [49], Customs Trade Partnership Against Terrorism (C-TPAT) [50], Container Security Initiative (CSI) [51], and Smart & Secure Trade lanes (SST) [52].

Therefore, there have been several efforts by the government, industry and academia to develop the intelligent container system that remotely monitors the container's contents along with its condition and location using advanced technologies.

L-3 Communications [53] in partnership with the Department of Homeland Security (DHS) has developed a system to ensure the container's integrity in transit using sensors and fusion techniques.

ODIN Technologies [54] has introduced an RFID solution called S.M.A.R.T. (Secure Material Accounted in Real Time) for the asset tracking of a supply chain. It reads RFID tags of items inside the container and transmits data to the server. Savi Technology [55] and AVANTE Technology [56] have used the RFID technology for asset tracking and have coupled RFID with GPS to support real-time locating systems (RTLS).

Jedermann et al. [57] have proposed a combination of RFID with sensor networks for the advanced tracking and monitoring of a supply chain.

5. AUTHENTICATION FOR ENS

Authentication is one of the common security services as discussed in Section 2.2.2. This process is necessary in ENS because it allows verification of the user's identity before access to resources and services via networks can be granted. For example, considering the example scenarios in Section 1.2.1, most devices are connected over wireless networks. Wireless networks are more vulnerable than wired networks because of the broadcast nature of wireless communications. Therefore, we need to protect the sensitive data transmitted. In the case of smart space, the data transmitted between devices can be confidential documents or personal contact information. Suppose a neighbor has a powerful antenna and sits within the communication range. Then, he can easily eavesdrop and learn the secret data. More serious forms of attacks are active attacks that attempt to modify the data or access resources. For example, a thief could turn off a security alarm in the house to enter. Obviously, a legitimate user does not want to allow other people to hear his communication, and access his resources. Therefore, the ENS requires an authentication process before communication, collaboration and access grant.

5.1. Motivation

As discussed in Section 2.2.3, the existing authentication mechanisms for traditional computing environments are based on the user input (e.g., password or PIN), the trusted-third party (e.g., public key infrastructure or a key distribution centre), or information derived from a personal characteristic (e.g., biometrics). In the ENS, however, these traditional authentication mechanisms are not appropriate and the authentication becomes a challenging problem because mutually unknown devices/users must verify each other's identity.

5.1.1. Requirements

Authentication for ENS should meet the following requirements.

- *Scalability*: In ENSs, the number of devices will increase tremendously. The traditional authentication techniques generally assume that there is a pre-established secret such as a password, key, or biometric information. This assumption is not applicable to ENS because of the possibility of communicating with strangers. It is impossible to manage a pre-established secret with all other devices in the world. Therefore, the traditional mechanisms will not scale to the huge computing environment [58].
- *Invisibility*: The ideal solution for the pervasive computing should be transparent to users [59]. Thus, the authentication should be an automatic and seamless process running in the background with minimal user interaction. The traditional ways to get authenticated that involve human interactions such as presenting passwords or biometrics (e.g. fingerprint or voice) are not suitable.
- *Autonomy*: In ENS, there is no fixed infrastructure. The authentication scheme through the trusted third party or centralized server is not feasible for this environment [60]. The authentication process for ENS should be autonomous.
- *Heterogeneity*: ENS consists of a wide range of devices from sensors to PCs and they have different capabilities in terms of computation, communication and other hardware resources (e.g. sensors). The authentication mechanism should be flexible to handle these heterogeneous devices.
- *Resource Constraints*: One of the characteristics of ENSs is resource constraint in terms of computation, communication, and energy. Most of devices do not have capability to perform expensive algorithms such as asymmetric algorithms or

authentication protocols including several message transmissions. Therefore, the authentication process should be efficient in terms of computation, communication, and energy.

5.1.2. Location-based Authentication

In many ENS applications, users/devices commonly form social groups to collaborate with others [61]. The group membership is changed based on some social context, such as the user's location, identity or interest. Currently, the location is the most widely-used type of context for various applications. There are many examples of location-based services such as location-based billing, location-based authentication, resource tracking, and location-based mobile advertising.

Considering the proposed architecture in Section 4.1.1, an MECD can manage a shared key with each end node in its sensor-level network and use these pre-shared keys for authentication. However, authenticating other neighboring MECDs for collaboration becomes problematic. MECDs can move anywhere and meet any unknown MECD that they have never seen before. Maintaining shared keys with all MECDs in ubiquitous environments is not possible.

To solve this problem, we can take the localized collaboration characteristic of MECD-level networks. In other words, the current location information can be used to authenticate a device in order to join a local network and collaborate with other devices. In this work, we focus on the *Location-based Authentication* problem which is the process to prove one's identity and authenticity by detecting its current location [62].

One of the challenging problems in location-based authentication is how to determine the location of a node. In [62], GPS technique is used to determine the node's physical location. However, putting GPS receivers in every node or manually configuring locations is not cost effective. Also, it is difficult to use in indoor environment.

There are many *localization* (also called *location determination*) schemes that have been proposed in the literature to allow nodes to estimate their locations using information transmitted by a set of reference nodes that know their own locations. Localization techniques can have different approaches: *distance bounding, in-region*, and *absolute location* [63]. The absolute location approach is used to determine the entity's absolute location. The distance bounding approach determines whether the entity is closer to the verifier than some distance or not. In the in-region approach, it determines whether the entity is inside a certain delimited region or not.

Most of localization research has been on the assumption of an absence of malicious nodes in a designated area. However, the truthfulness of the location information is a critical aspect [64]. There have been some works recently on *secure location verification* [65] [66] [67] [68] [69]. Secure location verification techniques can be categorized into two categories: range-dependent and range-independent techniques [70] [63]. Range-dependent schemes measure physical properties of the exchanged signals such as time of arrival (TOA) [68], the angle of arrival (AOA) [69], and received signal strength (RSS) [65]. On the other hand, range-independent schemes use other characteristics such as the existence of beacon signals [67] and physical channels [66]. These approaches do not require any measurements to be taken. Usually, range-dependent techniques provide more accurate location estimation, but are more expensive than range-independent techniques. Therefore, range-dependent techniques are used for absolute location approaches while range-independent techniques are generally used for the in-region or distance bounding location estimation.

The existing schemes require reference nodes which already know their location, pre-established trust relationships with reference nodes, or additional hardware (e.g., a directional antenna). However, the assumption of the existence of trusted reference nodes is not applicable to a pervasive computing environment. The need of additional hardware is also inappropriate because of the cost constraint of embedded devices. Therefore, we need a new solution where the entity's location is securely verified for ENSs.

5.2. Goals and Assumptions

In this work, we focus on applications which require authentication based on the user's current location. Especially, in our targeted applications described in Section 1.2.1, a device should be within a particular area (e.g., a room, office, classroom, ship, port, or container yard) in order to join a local network and collaborate with other devices. Therefore, we use the in-region approach to solve the *location-based authentication* problem.

5.2.1. Problem Definition and Assumptions

We define a self-organizing region-based authentication problem as follows:

Definition 5.2.1 (Self-organizing Region-based Authentication Problem) *A verifier v automatically authenticates a requester r when they are in a region R of interest. The region, R may be a room, a house, a building, a ship, a yard or other physical areas.*

Through this work, we assume that the requester claims that they present in a particular region of interest R rather than a particular point of location. This region, R must have some sort of physical control to restrict people into this area. In other words, only authorized people are allowed to enter this area. We assume that v verifies local region claims rather than all region claims. If v is currently in the region $R1$, then it must be able to verify whether r is in $R1$ or not. However, verifying other locations is not required.

Definition 5.2.2 (Co-located) *When v and r are present in the same region R, we call them* co-located*.*

The verifier v and the requester r are mobile nodes with wireless communication capability. If r sends a message to v, v must be able to receive it and then v and s both must behave according to the protocol. v will accept r if they are co-located and r behaves according to the protocol. Otherwise, v will reject r's claim.

We assume r and v are well-synchronized. There are several time synchronization algorithms available in the literature [71]. In [71] [72], 1 μsec high-precision clock can be achieved using small wireless sensors. We assume all devices use this synchronization technique to support an approximate 1 μsec high-precision clock.

The authentication must be an *automatic* process without any help. In other words, there must be no human involvement so that a user can concentrate on his or her work. Furthermore, r and p do not need any pre-shared information or pre-established trust relationship.

5.2.2. Threat Model

By this assumption, the adversary should not actually have any presence in the region R. However, all nodes may be malicious. In this work, we consider the active adversary model. The attacker can capture, record, intercept, replay or insert any message in the wireless communication medium using a powerful antenna. Therefore, the attacker can record the previous communications and replay valid packets to pair with other legitimate nodes as *replay attack*. In addition, there is the attacker *guessing* valid packets in order to get authenticated. The attacker can also try to *impersonate* a legitimate node by intercepting all messages between legitimate nodes and injecting a new message. It is known as a *man-in-the-middle attack*.

5.3. Self-organizing Authentication using Environmental Sounds

In this work, we suggest taking the *cyber-physical approach* in order to solve the self-organizing region-based authentication problem. Recently, researchers have taken a cyber-physical approach to security [73] and the *cyber-physical security solution* is the integration of the security solution with the physical processes. In this approach, a secret is generated from physical environments and used to verify the identity of a device/user.

Sensing information such as light, temperature, WiFi signal strength, and sound can be a candidate for this approach. In this work, however, we propose to use environmental sounds to take the advantages of the following facts.

- Environmental sound is physical context which is not input by a user.
- Devices within the particular region hear a similar environmental sound.
- A microphone is very cheap and many contemporary devices in embedded network systems are equipped with a microphone.
- Environmental sound is produced by random events in any physical location in any frequency range.

The hypothesis is that *if we can find an acoustic feature(s) which is only known by co-located devices at a given time, it can be used for authentication.*

5.3.1. Research Questions

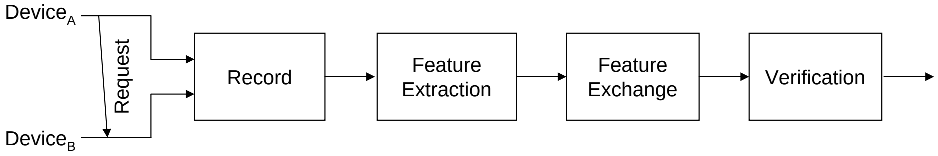


Fig. 39. Overall Authentication Procedure using Acoustic Features.

To identify research questions for this problem, we classify an overall procedure that is necessary for an acoustic feature-based authentication technique. Fig. 39 shows how our authentication process works. There are four phases; *Record, Feature Extraction, Feature Exchange* and *Verification.* Suppose $Device_A$ wants to initialize communication with $Device_B$. First, $Device_A$ sends the request. Then, both $Device_A$ and $Device_B$ record sound and extract features from recorded sound. After feature extraction, $Device_A$ and $Device_B$ exchange the features and compare them. If features are similar, authentication is done successfully. Otherwise, authentication fails.

1. **Record Phase**: As we discussed in Section 5.2.1, we assume that $Device_A$ and $Device_B$ are synchronized well (within approximate 1 μsec). Thus, a synchronization problem is not our concern.

   The sampling rate is one of the critical factors in the record phase. By the Nyquist theorem, we must have at least two samples in each cycle: one measuring the positive part of the wave and one measuring the negative part. More than two samples per cycle increases the amplitude accuracy, but less than two samples will cause the frequency of the wave to be completely missed. Because audio data consists of time series which are usually quite large, storing and computing them is costly, especially for resource-limited embedded devices. Obviously, *there is the trade-off between accuracy and efficiency.*

Sampling duration is another important factor in this phase. Apparently, longer sampling duration can improve the accuracy of this technique, but cause delay. In pervasive applications, however, authentication should not interrupt users or the other services. There is also the trade-off between accuracy and delay. Thus, the experimental study is necessary to determine appropriate sampling rate and duration.

2. **Feature Extraction Phase**: The feature extraction phase analyzes the incoming waveform and extracts certain features from it. The acoustic feature should support the following requirements to be used for the region-based authentication problem.

   - *Distinctiveness*: The acoustic feature(s) should be able to distinguish co-located devices from non co-located devices. In other words, when $Device_A$ and $Device_B$ are co-located, they should share the common feature(s). But, if $Device_A$ and $Device_B$ are not co-located, the feature(s) derived by $Device_A$ should differ from $Device_B$.
   - *Randomness and Time variance*: The acoustic feature(s) should be random so that no one can predict it. Furthermore, the acoustic feature(s) should vary over time to avoid reusability.

   There have been many studies on the acoustic feature extraction in speech recognition and classification. However, the algorithm should be inexpensive in terms of computation and energy for resource-limited mobile devices in ENS.

3. **Feature Exchange Phase**: The next question is how to exchange features securely against possible attacks defined in Section 5.2.2. If two co-located devices extract the exact same feature with high randomness and time-variance, it can be

directly sent because it can not be re-used by an attacker. However, it is very difficult to get identical acoustic features that are perfectly random and time-variant. Therefore, we need a secure way to exchange acoustic features even if they are not perfectly identical, random, and time-variant.

4. **Verification Phase**: After exchanging extracted features, two devices will verify whether they are co-located or not based on the similarity of acoustic features. The question on this phase is how to measure similarity. The requirement for this verification process is that it should be able to reject a request of non co-located devices so that only devices within the same region can be authenticated by each other.

5.3.2. Experimental Study: Acoustic Feature Extraction

Feature extraction can be categorized into two types: time-domain (temporal) and frequency-domain (spectral) feature extraction. Time-domain features are calculated directly from the original sound waveform. Frequency-domain feature extraction first transforms the sound waveform to the frequency domain using Fourier transform and then calculates the features from transformed signal [74].

The Zero Crossing Rate (ZCR) and Short Time Energy (STE) are the most widely used time domain features. ZCR is a number of zero-crossing in a frame. A zero-crossing occurs when adjacent samples have different signs [75]. The ZCR is defined for the single frame as

$$ZCR_n = \frac{1}{2N} \sum_{m=-\infty}^{\infty} |sgn(x(m)) - sgn(x(m-1))| w(n-m), \tag{5.1}$$

where $x(m)$ is a discrete audio signal, n is the time index of ZCR, and $w(m)$ is the window of length N.

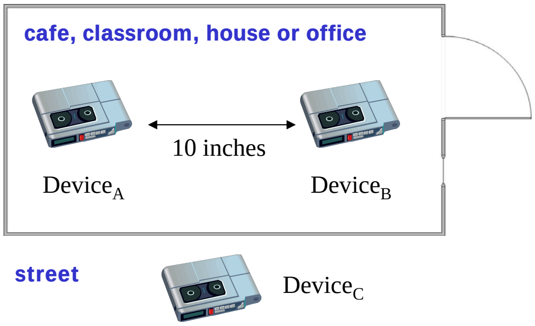


Fig. 40. Experimental Setup.

$$sgn(x(m)) = \begin{cases} 1, & if x \geq 0 \\ -1, & otherwise \end{cases}$$

STE is defined as:

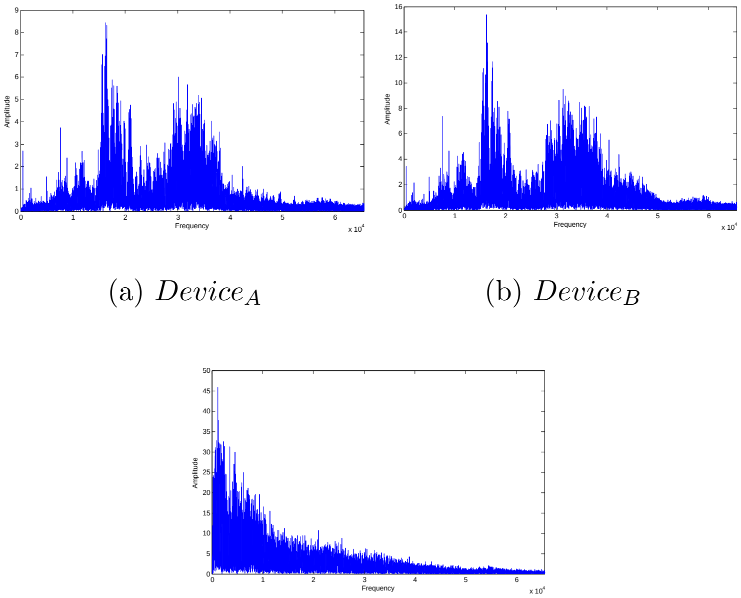
$$E_n = \frac{1}{N} \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2, \tag{5.2}$$

Generally, the amplitude of unvoiced speech segments is much lower than the amplitude of voiced segments. The energy of the speech signal provides a representation that reflects these amplitude variations [76].

For frequency-domain analysis, Discrete Fourier Transform (DFT) is widely used in signal processing to analyze the frequencies contained in a sampled signal. The DFT function implements the transformation of the feature vectors of length N by:

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}, \tag{5.3}$$

where x is a discrete audio signal and $\omega_N = e^{(-2\pi i)/N}$ is an Nth root of unity. DFT can be computed efficiently in practice using a Fast Fourier Transform (FFT) algorithm. An N-point FFT can produce $\frac{N}{2}$ unique features because there is even symmetry around the center point [77].



(a) $Device_A$

(b) $Device_B$

(c) $Device_C$

Fig. 41. 2^{17}-point FFT Results Recorded in the House.

To investigate spectral information in different situations, we collected data in four environments, 1) cafe, 2) cassroom, 3) house and 4) office. Two recorders ($Device_A$ and $Device_B$) were located within 10 inches of each other and another ($Device_C$) was located outside shown as Fig. 40. Sounds were recorded for 10 seconds at an 8 kHz sampling rate.

Fig. 41 shows the results of 2^{17}-point FFT recorded in the house. In Fig. 41(a) and 41(b), we can see that there are similar patterns in the results of 2^{17}-point FFT between co-located devices. However, when a device is not co-located, the patterns are different as shown in Fig. 41(c). Specifically, the peaks occur at similar frequencies and they can be simply used as acoustic features.

To compare these three acoustic feature extraction algorithms, we calculated STE, ZCR, and the peak value of frequency-domain signal (performed FFT) on a 1-sec window

Table II. Feature Match Rates in Different Environments for Co-located Devices.

	Cafe	Classroom	House	Office
Short Time Energy (STE)	0	0	0	0
Zero Crossing Rate (ZCR)	0	0.00301	0.00053	0.00736
Peak on Fast Fourier Transform (FFT)	0.8666	0.8875	0.4978	0.5173

Table III. Feature Match Rates in Different Environments for Non Co-located Devices.

	Cafe	Classroom	House	Office
Short Time Energy (STE)	0	0	0	0
Zero Crossing Rate (ZCR)	0.3903	0	0	0
Peak on Fast Fourier Transform (FFT)	0.0064	0.0019	0.0011	0.0084

for different cases. Then, we checked if two devices generated the same value for each acoustic feature (STE, ZCR, and peak on FFT). Table II and III show the rate of matching two features of co-located devices ($Device_A$ and $Device_B$) and non co-located devices ($Device_A$ and $Device_C$), respectively.

The experimental results show that two co-located devices can not have the same STE features. For ZCR, most of the cases provides very small match rates even though the two devices are co-located. Therefore, these two features can not be used to determined whether two devices are co-located or not. On the other hand, the peak of FFT provides approximately 0.8 for cafe and class cases and 0.5 for house and office cases. Although non co-located devices can have the same feature, the rates are very small (less than 0.01). Therefore, we can conclude that the frequency domain features are more appropriate for this problem.

5.3.3. Experimental Study: Frequency Domain Acoustic Feature Extraction

When the number of FFT points increases, the resolution of the spectrum obtained also increases. At the same time, more computations and time are needed. Because of the limited resources of ENS, we need to examine the trade-off between accuracy and computational cost.

First, we compare correlation of two recorded audio signals at different lengths of FFT in order to see the effect of the length of FFT on accuracy. For each type of environments (cafe, classroom, office, and house), three recorders recorded sound for 10 seconds at 8 kHz. Note that $Device_A$ and $Device_B$ are located within 10 inches of each other and $Device_C$ is located outside of the region. We perform the fft function on each recorded sound with different points and then perform the corrcoef function of two signals in Matlab. We repeated this process about 40 times.

The corrcoef function in Matlab returns *cross-correlation* which is a measure of the similarity of two signals. The cross-correlation coefficients, $crosscorr_k$ can be defined as a function of the time-lag k [78].

$$crosscorr_k = \frac{\sum_{t=1}^{N-k}(x_t - \overline{x})(y_{t+k} - \overline{y})}{\sqrt{\sum_{t=1}^{N-1}(x_t - \overline{x})^2}\sqrt{\sum_{t=1}^{N-1}(y_t - \overline{y})^2}}, \tag{5.4}$$

where x_t are y_t are signals obtained at time t, and t is from 1 to N. $\overline{x}$ and $\overline{y}$ are the mean of x and y values respectively. The corrcoef in Matlab is the zeroth lag function. $crosscorr_k$ will lie in the range [-1, 1], with 1 indicating perfect correlation and -1 indicating perfect anti-correlation (the inverse of one of the series). Therefore, in our work, the value of cross-correlation coefficients is close to 1 when two devices are co-located. On the other side, when the cross-correlation coefficients become close to 0, they are not co-related. For this work, we do not consider the negative correlation as co-location.

We compare the cross-correlation coefficients of two cases: 1) $Device_A$ and $Device_B$ (co-located) and 2) $Device_A$ and $Device_C$ (not co-located). To be able to distinguish two cases, the cross-correlation coefficients should not be overlapped. Fig. 42, 43, 44 and

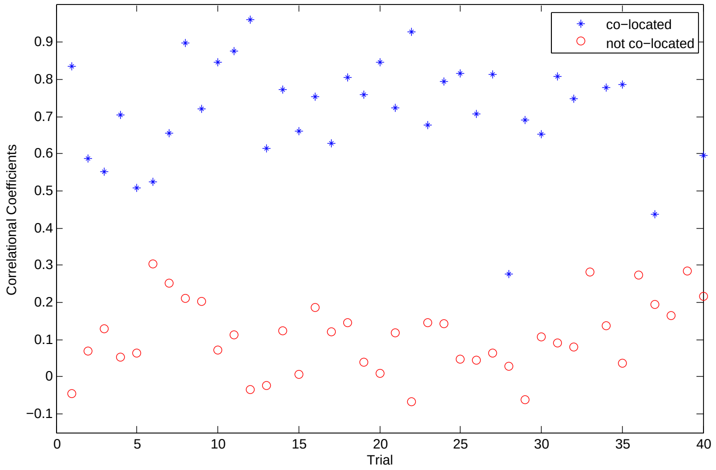


Fig. 42. Correlational Coefficients of Sound Recorded at a Cafe with 2^8-point FFT.

45 show the correlational coefficients of 2^8-point FFT feature extraction with 256 Hz sampling rate. The lines with stars and the circles indicate the correlational coefficients between $Device_A$ and $Device_B$ (co-located) and between $Device_A$ and $Device_C$ (not co-located), respectively. In Fig. 43 and 45, there is no overlap between two cases. However, Fig. 42 and 44 have overlap between them. That means they are not distinguishable.

To determine whether two cases are distinguishable or not, we find the maximum value of the correlational coeffients for devices that are not co-located. We then check if the correlational coefficient of co-located devices is below the threshold. If so, that means there is overlapping and the two cases are not distinguishable. After checking all 40 trials, we calculate the percentages of the trials that the correlational coeffcient of co-located devices is below the maximum value of the correlational coefficients of non co-located devices. Table IV shows the results with 2^8-point FFT, the 2^{13}-point and 2^{17}-point FFT. Obviously, the larger FFT points provide the better distinctiveness of the two cases.

To see the effects of the length of the FFT function on computational cost, we

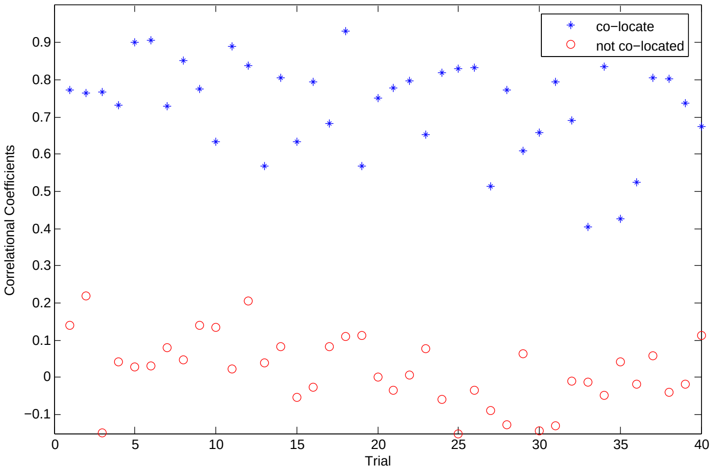


Fig. 43. Correlational Coefficients of Sound Recorded at a Classroom with 2^8-point FFT.

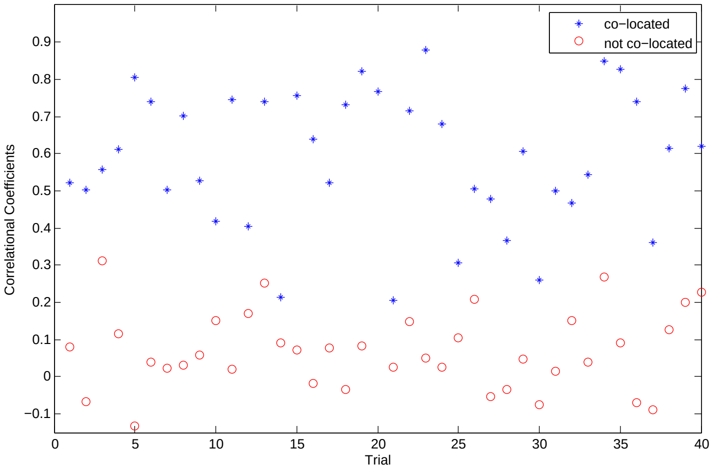


Fig. 44. Correlational Coefficients of Sound Recorded at a House with 2^8-point FFT.

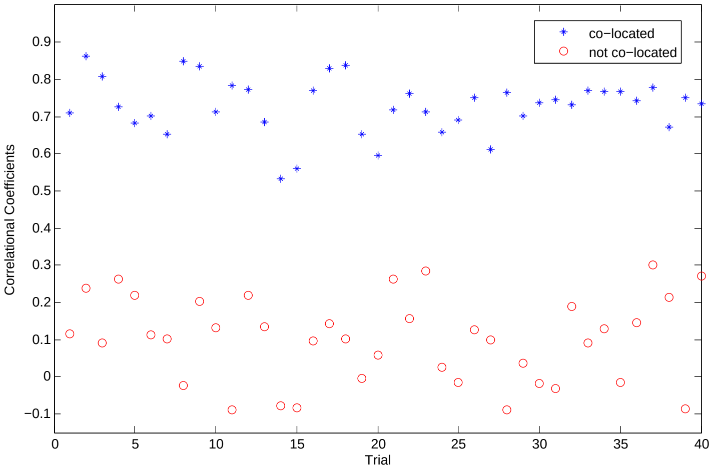


Fig. 45. Correlational Coefficients of Sound Recorded at an Office with 2^8-point FFT.

Table IV. The Percentages of the Trials that the Correlational Coeffcient of Co-located Devices is Below the Maximum Value of the Correlational Coefficients of Non Co-located Devices with Different Lengths of FFT.

Length of FFT function	Cafe	Classroom	House	Office
2^8-point FFT	0%	2.5%	10%	0%
2^{13}-point FFT	0%	0%	5%	0%
2^{17}-point FFT	0%	0%	0%	0%

compare the execution time of 2^6-point, 2^8-point and 2^9-point FFT functions on Mica2 and TelosB motes. Bhatti et al. [79] provide the comparision of the execution times of 2^6-point and 2^9-point FFT functions on a Mica2 mote. Neuman and Gaura [82] include the measurement of energy consumption of the 2^7-point FFT function on a Mica2 mote. In [80], they implemented the 2^8-point FFT function on a TelosB mote and measured the execution time and energy consumption. A Mica2 mote has 4 kB of RAM provided by the Atmel Mega 128 chip and a TelosB mote has 10 kB of RAM provided by the TI MSP430 chip. Table V provides the details of specifications of Mica2 and TelosB motes [26] [81] [82]. Table VI shows the summary of the measurement of the different lengths of FFT functions. A typical 1.5V AA battery has a capacity of about

Table V. Mica2 and TelosB Mote Designs.

	Mica2 [81]	TelosB [26]
Size	32 × 57	32 × 65
CPU	8 MHz Atmel Mega 128	8 MHz TI MSP430
Program Flash Memory	128 kB	48 kB
RAM	4 kB	10 kB

Table VI. Execution Time and Energy Consumption of the Different Lengths of FFT on Mica2 and TelosB Motes.

The length of FFT	Platform	Execution Time	Total Energy Cost
2^6-point FFT	Mica2	56 msec [79]	
2^7-point FFT	Mica2		930 μJ [82]
2^8-point FFT	TelosB with Radio-off	440 msec	1.32 mJ [80]
2^8-point FFT	TelosB with Radio-on	440 msec	25.81 mJ [80]
2^9-point FFT	Mica2	30 sec [79]	

2600 mAh [80]. With two AA batteries, a TelosB mote can compute a 256-point FFT function about 313 times with radio-on mode and about 6136 times with radio-off mode. A 2^9-point FFT takes 30 sec which produces a large delay additionally. In conclusion, FFT computation is very expensive to perform on mobile devices. Therefore, we may need to use a small number of FFT points even though they do not support the perfect distinctiveness.

The next question is whether frequency-domain features are random and time-variant. This requirement is important because randomness and time-variance make sure that the set of features is unpredictable by attackers. We perform randomness tests using one of the popular randomness testing programs, "ENT" [83]. ENT performs a variety of tests on the stream of bytes as an input file and produces output as follows [83]:

- *Entropy* is a measure of the uncertainty associated with a variable, expressed as a number of bits per character. For a random variable X with n possible values

$x_1, \ldots, x_n$, the entropy $H(X)$ is defined as

$$H(X) = \sum_{i=1}^{n} p(x_i) log_b \frac{1}{P(X_i)}, \tag{5.5}$$

where $p(x_i)$ is the probability of x_i [84]. Common values of b are 2. The measure should be maximal if all the values of X are equally likely (all possibilities, $p(x_i)$ are equal). In the ENT program, the entropy should be 8 bits per byte when the inputs are random.

- *Chi-square Test* is one of the most commonly used tests for the randomness of data. The ENT program interprets the percentage as the degree to which the sequence tested is suspected of being non-random. If the percentage is greater than 99% or less than 1%, the sequence is almost certainly "not random". If the percentage is between 99% and 95% or between 1% and 5%, the sequence is "suspect". Percentages between 90% and 95% and 5% and 10% indicate the sequence is "almost suspect" [83].

- *Arithmetic Mean* is simply the result of summing all the bytes in the file and dividing by the file length. If the data are close to random, this should be about 127.5. If the mean departs from this value, the values are consistently high or low [83].

- *Monte Carlo Value for Pi* Monte Carlo methods are also generally used as a randomness test. In this program, each successive sequence of six bytes is used as 24 bit X and Y co-ordinates within a square. If the (X, Y) point is inside a circle inscribed within the square, it is considered a "hit". With large streams, the percentage of hits called Pi will approach the correct value of π if the sequence is close to random [83] [85].

Table VII. Results of Randomness Test using ENT Program.

	Cafe	Classroom	House	Office
Entropy	7.57	7.58	7.52	7.54
Chi-square Test (the number of success)	0	0	0	0
Arithmetic Mean	126.28	125.75	128.30	127.10
Monte Carlo Value for Pi	3.12	3.13	3.12	3.14
Serial Correlation Coefficient	0.70	0.43	0.81	1.53

- *Serial Correlation Coefficient* measures the correlation of successive bytes in the file. For random sequences, this value will be close to zero [83].

We first performed 2^8-point FFT on each 10 second sound and generated the sequence of the results for all sound data (approximately 40 trials). We then ran the ENT program with this sequence and Table VII shows the results for each environment. The sequence of frequency-domain features succeeded in *Entropy, Arithmetic Mean*, and *Monte Carlo Value for pi* tests, but failed in *Chi-square* and *Serial Correlation Coefficient* tests.

5.3.4. Proposed Authentication Scheme

Previously, we discussed our experimental study on acoustic feature extraction in Section 5.3.2. We found that the peaks in frequency domain occur at similar locations. However, they may not be exactly the same. Second, FFT computation is very expensive to be performed on mobile devices. Therefore, we may need to use a small number of FFT points even though they do not support the perfect distinctiveness. Third, sounds vary over time, but they may not be highly random and time-variant in the short term.

On the basis of these observations, we propose a simple acoustic feature extraction and verification methods. Since frequency-domain feature extraction produces an overall result detailing the frequencies contained in the entire audio signal, there is no distinction as to where these frequencies occurred in the signal. However, environmental sound is a time-varying signal caused by random environmental activities. This work uses a hybrid approach to take advantages of spectral and temporal information. The recorded audio signal is split into several frames and frequency-domain feature extraction is then performed on each frame.

For the secure feature exchange, a Bloom filter is used in this work. [86] used the Bloom filter to test membership and similarly this work uses this filter to express multiple acoustic features in one set. As discussed above, the distinctiveness may not be perfect because of the trade-off between computational complexity and accuracy. The multiple acoustic features are suggested in order to improve the possibility to be matched and distinctiveness. The Bloom filter is represented as a bit array. Each acoustic feature is hashed using hash functions such as MD5 and SHA-1. They have good properties: computationally impossible to find the original message from the hash result and computationally impossible to find two distinct messages with the same output [86]. Due to the different lengths of the results of the hash functions, the modulo operation is used to fit the same length of the bit array. The modulo function is applied to two numbers: the hash result and the length of the filter. These results are used as indexes to determine the bit to set in the filter. The bits of these indexes in the array are set as 1 and the other bits are set as zero. The filter is used as the feature set in this work. The main advantage of using the Bloom filter in this dissertation is that the length of the feature set can be fixed with different number of features in the set. Therefore, the network packet size of the feature exchange step does not need to be adapted. In Fig. 46, the number of features equals the number of the window, w.

Table VIII describes the notation used in this section. Fig. 46 presents the acoustic feature extraction mechanism proposed in this dissertation. Suppose the requester, r

sends a request to the verifier, v. The overall steps of the feature extraction are as follows:

1. v generates a random number, n, and sends it to r. The random number is used to prevent this authentication process from reusing valid features of previous communications.
2. Both v and r start recording sounds.
3. Each device performs the feature extraction process. The recorded sounds are divided into w windows and FFT computation is applied to each window. The peak of each window is calculated and the results are denoted as P_i when i is from 1 to w.
4. A peak value, P_i with $1 \leq i \leq w$, is concatenated with n, $P_i|n$. Apply the hash function to each result, $\mathrm{H}(P_i|n)$.
5. The hashed results are applied to the modulo function, $H(P_i|n) \bmod l$ when l is the length of the filter. The Bloom filter is set using these modulo results. Because the peaks of the windows can be duplicated, a bit in the filter can be set multiple times. Thus, the number of bits with the value of 1 is up to w.
6. r sends the filter as the feature set to v.

After receiving the feature set, v performs the verification step. To verify it, v compares every bit of two arrays: one received from r and another computed locally. If the percentage of the matching bits $\geq t\%$, r is authenticated successfully by v. Otherwise, the authentication fails. Fig. 47 describes the verification step proposed in this dissertation.

This proposed acoustic feature extraction and verification scheme can be extended to the key exchange protocol and location-based access control using the existing tech-

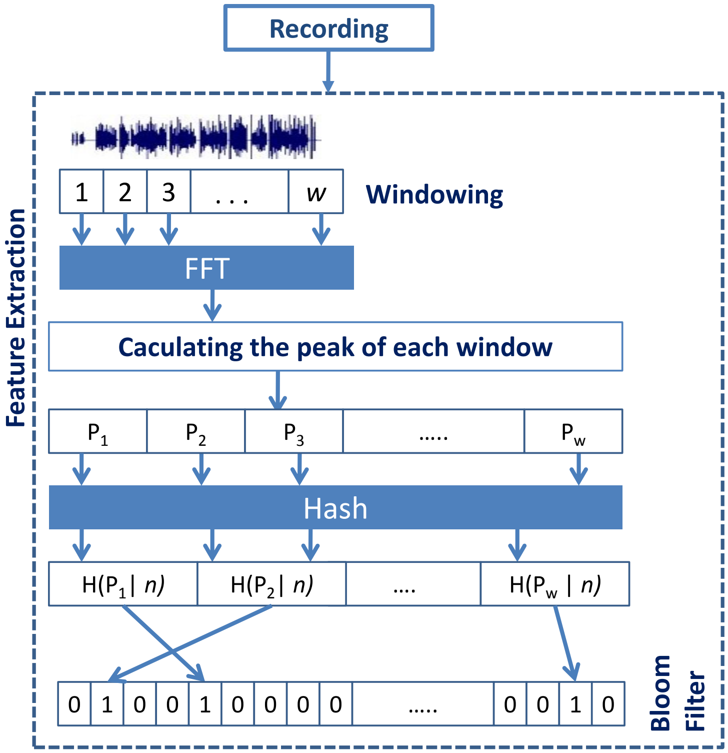


Fig. 46. Proposed Acoustic Feature Extraction Mechanism.

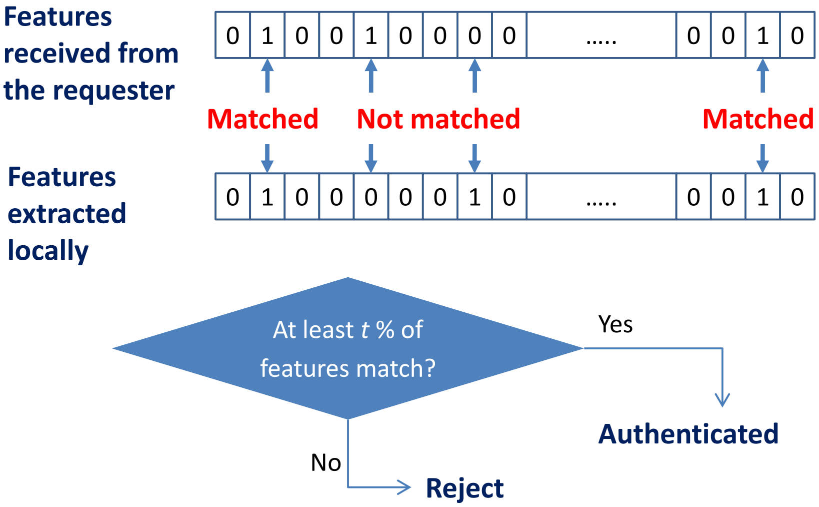


Fig. 47. Proposed Verification Mechanism

Table VIII. Notations.

Notation	Description
ID_i	The unique ID of device i.
P_i	The peak of window i. It is the index of the maximum amplitude on the window i.
FS_i	The feature set calculated by device i.
$a \bmod b$	The modulo operation of two operands: a and b. The modulo operation returns the remainder when dividing a by b.
$H(m)$	The hash of the message m
$m_1 \| m_2$	The concatenation of the message m_1 and m_2
$A \rightarrow B : m$	A sends the message m to B

niques. In this part, we will give an example of the extended Diffie-Hellman key exchange protocol [90] with the proposed scheme. The Diffie-Hellman protocol will be discussed in Section 5.4. The protocol is as follows:

1. $r \rightarrow v$**:** $ID_r|n1$

   r sends the request to v with its ID, ID_r and a random number, $n1$. r and v start recording sound. Then, they divide sounds into windows, perform the FFT function on each window, and find the peak of each result, P_i with $1 \leq i \leq w$ when w is the number of the window.

2. $v \rightarrow r : ID_v|n2|(g^a \textbf{ mod } p)|FS_v$

   v generates a random number a and calculates $ID_v|n1|(g^a \bmod p) = n$. g and p are a prime number and a generator explained in Section 5.4. Using n, it calculates the feature set, FS_v using the steps discussed above. Then, v generates a random number, $n2$ and sends the message, $ID_v|n2|(g^a \bmod p)|FS_v$ to r.

3. $r \rightarrow v : ID_r|(g^b \textbf{ mod } p)|FS_r$

   r calculates the feature set using P_i and $n1$ that has been generated at step 1, and ID_v and $(g^a \bmod p)$ that were received from v at step 3. Then, it performs the verification step with the calculated feature set and received feature set, FS_r. If they match, r proceeds to the next step. It generates a random number b and the shared key k, g^{ab} is computed. Then, r calculates $ID_r|n2|(g^{ab} \bmod p) = n$. Using n, it calculates the feature set, FS_r and sends the message, $ID_r|n2|(g^b \bmod p)|FS_r$ to v.

4. v performs a verification process similar to the previous step (step 3). It calculates $ID_r|n2|(g^{ab} \bmod p) = n$ with ID_r and $g^b \bmod p$ extracted from the received message. Using n and peaks calculated at step 1, it computes the feature set. If the calculated feature set and FS_r match, r gets authenticated and the shared key $k = g^{ab}$ will be used for communication between v and r.

To analyze the security of this protocol, we will consider the possible attacks against *man-in-the-middle attack, replay attack* and *guessing attack* defined in Section 5.2.2.

- *Replay attack*: If the feature set does not vary over time, an attacker can capture the valid features from the previous communication and simply reuse them to get authenticated. In the proposed protocol, two devices select random numbers $n1$ and $n2$ and the peaks (acoustic features) are changed continuously. Therefore, an attacker can not copy and reuse the valid feature set from the previous communications.

- *Man-in-the-middle attack*: A man-in-the-middle attacker tries to intercept and change messages with the valid feature in order to impersonate a legitimate user. Therefore, he or she needs a valid feature set that has at least $t\%$ (Note t is the threshold) of bits matched. Experimental results in Section 5.3.5 will show that a device located outside of the restricted area can not sense enough acoustic features to generate a valid feature set.

- *Guessing attack*: means guessing a valid feature set in order to impersonate a co-located device. This depends on the length of the feature set. To represent the 128-bit output of the MD-5 hash function in a filter, the length of the bit array can be 128-bit. With the longer bits of the feature set, it is hard to guess a valid set.

5.3.5. Experimental Results

The acoustic feature extraction technique proposed in Section 5.3.4 was implemented on a Google Android Dev 1 phone that is designed for advanced developers. It is a SIM (Subscriber Identity Module)-unlocked and hardware-unlocked device. Therefore, it is possible to use any SIM card in the device [87]. The specification of Android Dev Phone 1 is in Table IX. We tested it in a room for a smart home scenario. To find the effect of distances on the distinctiveness, two Android phones were placed in a room within 10m, 20m, and 30m of each other. Then, one Android phone was placed inside and another was placed outside of the room.

We used 2^8-point FFT function because of the reasonable delay and computational complexity. The length of each window was 1 second. The sampling rate was 8kHz that can be supported by android.media.AudioRecord library [88]. The 1 second window of sound is *downsampled* by 32 (from 8k to 256) to reduce the sampling rate of a signal. When the signal is downsampled, the signal may have aliasing that causes unwanted signals in the desired frequency band [89]. To prevent aliasing, the Shannon-Nyquist sampling theorem must be satisfied. The sampling theorem says that a digital signal processing system with a sampling rate of f_s can sample a signal with its highest frequency up to half of the sampling rate, $f_s/2$ (called folding frequency) without aliasing

Table IX. Android Dev Phone 1 Hardware Specifications.

Android Dev Phone 1 [87]
Touch screen
Trackball
3.2 megapixel camera with autofocus
Wi-Fi
GPS-enabled
Bluetooth v2.0 with
- Handsfree profile v1.5
- Headset profile v1
3G WCDMA (1700/2100 MHz)
Quad-band GSM (850/900/1800/1900 MHz)
256MB Flash Memory
192MB RAM
QWERTY slider keyboard
Includes 1 GB MicroSD card

noise [89]. This folding frequency will be decreased after downsampling. Therefore, if the signal to be downsampled has frequency components larger than the new folding frequency, aliasing effects will occur. To overcome this problem, in signal processing, a low-pass filter is used as an anti-aliasing filter to reduce the bandwidth of the signal before the signal is downsampled.

This work used a Finite Impulse Response (FIR) filter that is one of the primary types of filters used in Digital Signal Processing [89]. The output of an FIR filter is defined as:

$$y(n) = \sum_{i=0}^{K} b_i x(n-i) = b_0 x(n) + b_1 x(n-1) + \cdots + b_k x(n-k), \tag{5.6}$$

where $x(n)$ is the input signal and $y(n)$ is the output signal. b_i are the filter coefficients and $K+1$ denotes the FIR filter length. We call it the K-th order filter (also called K-th tap filter). This length, K, is related to the amount of memory needed, the number of calculations required, and the amount of filtering that it can do. Reducing K used in the filter will reduce the number of calculations to process, but the quality of the filtering will be degraded. To calculate b_i for FIR low pass-filter in our case, we used the MATLAB function $fir1(K, W_n)$ that returns the vector of b the FIR lowpass filter of order K and the normalized cutoff frequency W_n. W_n is a number between 0 and 1, where 1 corresponds to the sampling frequency in the Shannon-Nyquist sampling theorem. In this work, we used $k = 32$ and $W_n = 0.0625$.

After passing the low-pass FIR filter, we implement the downsampling part. To implement it with a downsampling factor = M, we can simply keep every Mth sample. In our case, to downsample the signal by 32, we keep every 32nd sample, and throw 31 samples out of every 32 samples away.

After the downsampling step, the acoustic feature extraction and verification processes discussed in Section 5.3.4 are performed on each phone.

To evaluate the satisfaction of distinctiveness requirement, we use *false positive* and *false negative rates.*

- *False negative*: the error of failing to reject authentication when it is, in fact, false.

- *False positive*: the error of rejecting authentication when it is actually true.

In a perfect authentication system, the false positive rate and negative rate should be zero. Non-zero false negative rates mean that the system is vulnerable, so the false negative is more critical.

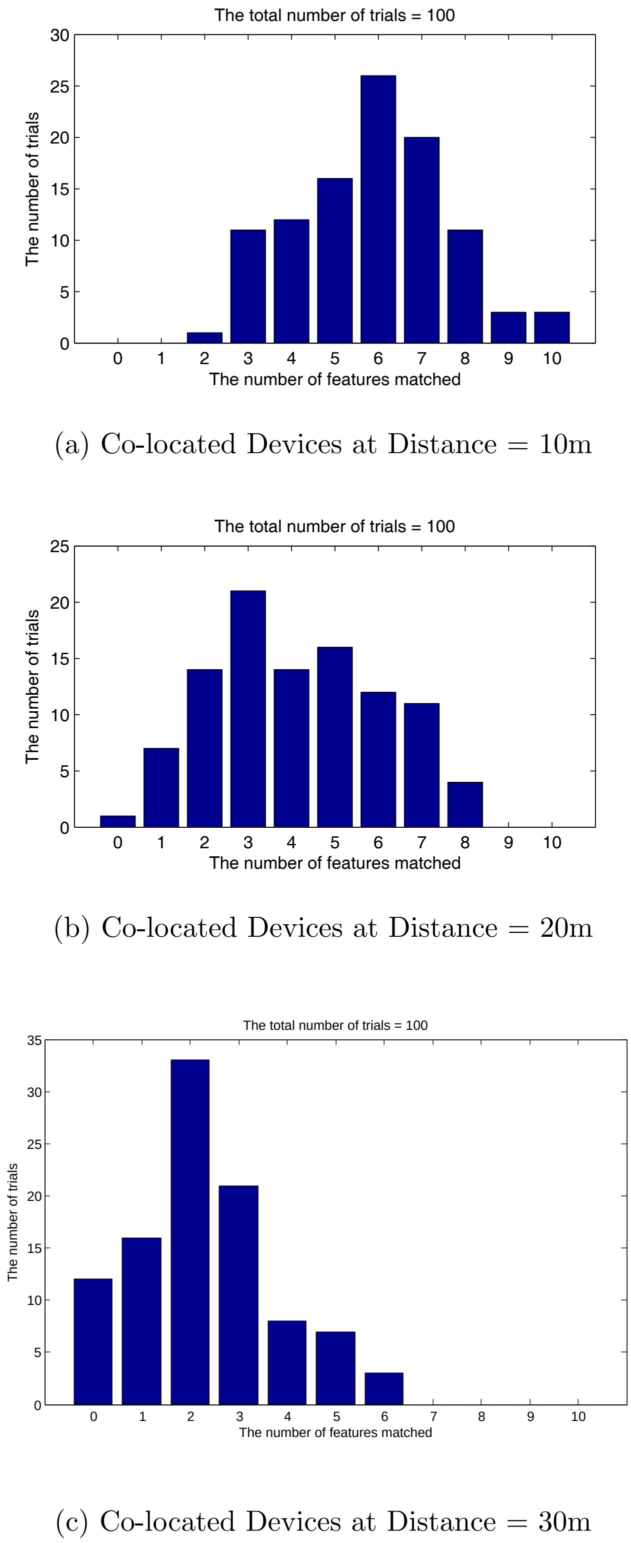
Fig. 48 and Fig. 48 show the experimental results of co-located devices at different distances and non co-located devices with 10 acoustic features (the number of windows, $w = 10$), respectively. As shown in Fig. 48, we found that the number of features matched decreased as the distance increased. As the results in Fig. 49, we set the threshold rate, $t = 40\%$ to prevent non-zero false negative rate.

Fig. 50 and 51 present the FNR and FPR with different test cases: co-located devices with distances = 10m, 20m, and 30m and non co-located devices when $w = 10$ and $w = 6$, respectively. For both cases, the threshold, $t = 40\%$ can be used for FNR = 0. However, the threshold value may be different in different environments (e.g. classroom or conference room). With the larger threshold, t, obviously, the possibility of FNR decreases, but FPR increases. This means that the system prevents attackers getting authenticated, but a device may need to retry several times even though it is co-located with another and supposed to be successfully authenticated.

Another issue is that this scheme provides reasonable results when two devices are within proximity and therefore it may not cover an entire room area. This is mostly because we use a small number of the FFT function and the microphone of the Android phone has a noise like a hissing sound. We expect many mobile devices like smart phones and tablet PCs have better hardware resources. In addition, we can improve this scheme by using multiple contextual informations as well as the trust propagation technique discussed in Section 6.1.

5.4. Related Work

The Diffie-Hellman key exchange protocol [90] is a well-known key agreement protocol. It allows two entities to agree on a secret key over an insecure medium. Fig. 52 describes the basic Diffie-Hellman protocol. Suppose Alice and Bob want to set up a shared key. The protocol uses two parameters p and g. They are both public. p is a prime number and g (usually called a generator) is a primitive root less than p. A



(a) Co-located Devices at Distance = 10m

(b) Co-located Devices at Distance = 20m

(c) Co-located Devices at Distance = 30m

Fig. 48. Number of Matched Features between Co-located Devices with 10 Features.

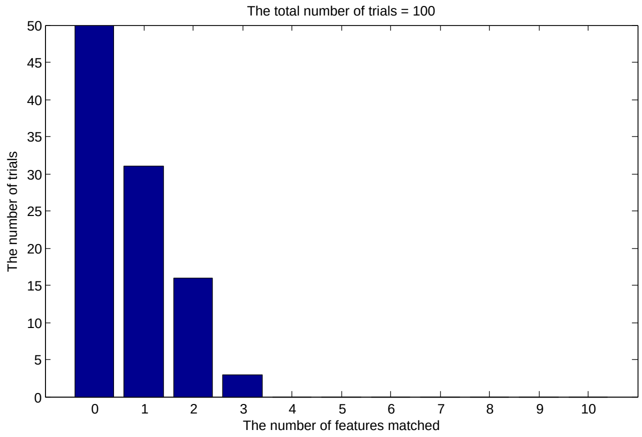


Fig. 49. Number of Matched Features between Non Co-located Devices with 10 Features

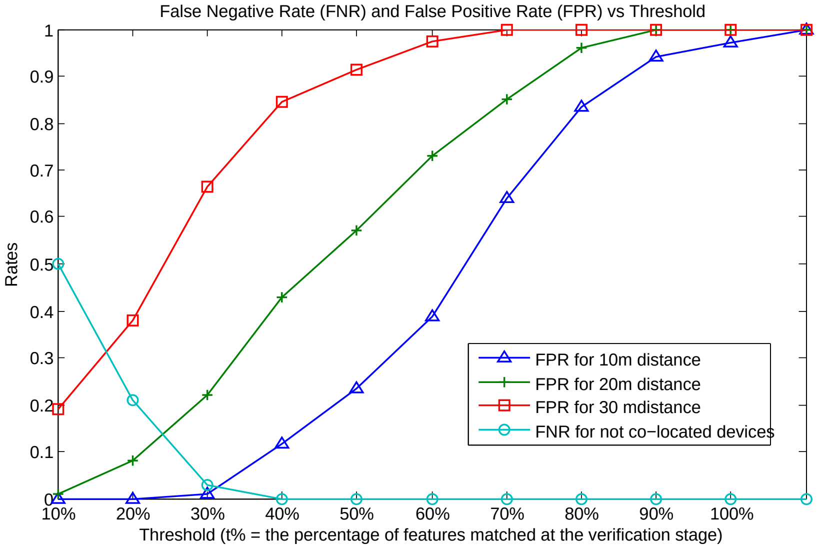


Fig. 50. False Negative Rate and False Positive Rate at Different Distances with 10 Features.

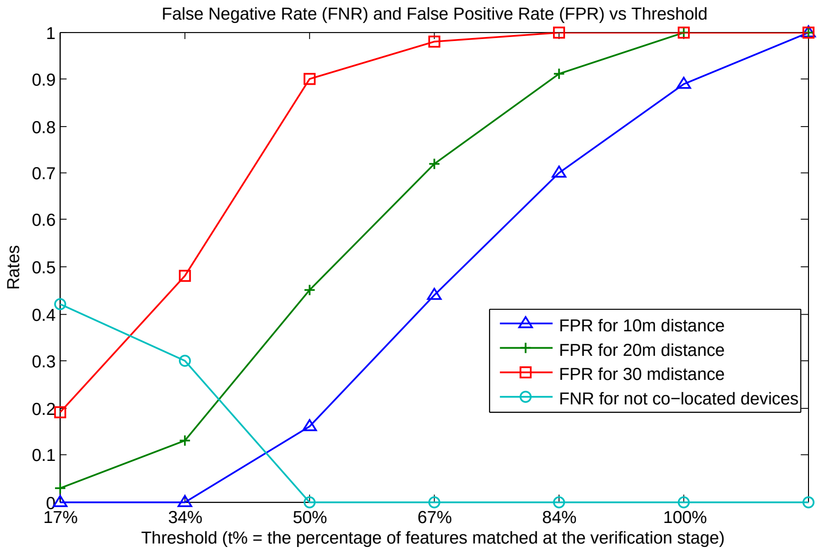


Fig. 51. False Negative Rate and False Positive Rate at Different Distances with 6 Features.

primitive root is any number with the following property: for every number n between 1 and p-1 inclusive, there is a power k of g such that $n = \text{mod}\ (g^k, p)$.

The protocol is as follows:

1. Alice generates a random number a and Bob generates a random number b.
2. They derive their public values using p and g and their private values. Alice's public value is g^a mod p. Bob's public value is g^b mod p.
3. Alice and Bob exchange their public values.
4. Alice computes $g^{ba} = (g^b)^a$ mod p, and Bob computes $g^{ab} = (g^a)^b$ mod p. Since $g^{ba} = g^{ab} = k$, Alice and Bob now have a shared secret key k.

In this protocol, it is computationally infeasible to calculate the shared secret key $k = g^{ab}$ mod p given the two public values g^a mod p and g^b mod p when the prime p is sufficiently large [90]. In other words, even though g^a mod p and g^b mod p are known to everyone, they can not compute g^{ab} mod p without a or b.

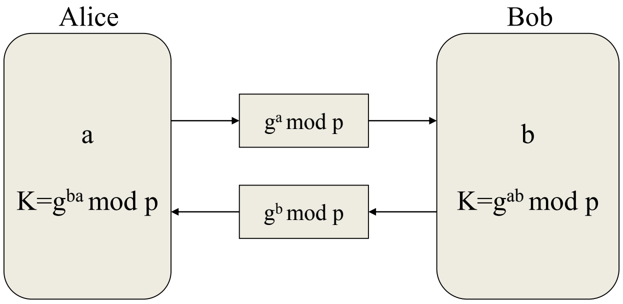


Fig. 52. Diffie-Hellman Key Agreement Protocol.

However, the Diffie-Hellman key exchange protocol does not have an authentication process. Thus, it is vulnerable to a *man-in-the-middle attack* which monitors a communication between two parties and falsifies the exchanges to impersonate one of the parties. A number of enhancements [91], [92], [93] have been developed, but they use a public-key certificate or pre-shared password for authentication.

Recently several researchers have developed systems to authenticate and establish a key without a pre-existing trust relationship such as a public-key certificate and password. Stajano et al. have introduced the Resurrecting Duckling technique in [94]. When a new device wakes up, it is imprinted by the active master (like a duckling recognizes the mother). This imprinting must be done by the physical contact. However, the requirement of physical contact among all communicating devices will be too restrictive for our targeted applications.

McCune et al. [95] have developed an authentication mechanism using a visual channel called Seeing-is-Believing (SiB). SiB assumes a human user is capable of visually identifying the target device. In this approach, the sender first sends its key to the other via wireless channel. At the same time, it displays the hash of its key in the form of a barcode. On the receiver side, the device should have a photo camera. Using this photo camera, a user scans the bar code on the sender's display. Then, the receiver translates the scanned barcode into a binary key, and compares it with the key received over the wireless channel. One problem in SiB is that devices must have additional hardware (a display or a photo camera). Also, this lays burden upon a human user.

Some papers [96] [97] have proposed using an audio channel for authentication. In Loud-and-Clear [96], two devices exchange their keys over the wireless channel. The hash of the key is translated in syntactically correct (but usually nonsensical) English-like sentence. One device plays it and the other displays it. The human user compares them and verifies their keys. HAPADEP in [97] is similar to Loud-and-Clear, but it does not need a display. After key exchange, both devices play the key as sound. A user verifies the key by listening to a sound recording. Clearly, these schemes are not suitable to our targeted applications because of the reliance on human assistant. Another drawback is that translating and encoding (to sound) is too expensive for the resource-limited mobile devices.

The Amigo system by Varshavsky in [98] has used dynamic characteristics of the radio environment. Two devices first listen to their radio environment and derive a shared secret from it. If these devices are in close proximity, they will share the same secret. The advantage of this technique is that human involvement and additional hardware are not required. However, the performance of this technique highly depends on the types of WiFi cards and network conditions, such as WiFi usage. Some environments may have no radio signals or not vary over time, and so attackers can easily learn about the radio environments.

6. CONCLUSIONS AND FUTURE WORK

Collaboration is one of the important features of ENS because sharing available resources and information can enhance the system performance as well as context awareness. However, it could produce additional communication and data processing to the system, especially gateways. To address this problem, this dissertation proposed a hierarchical architecture with mobile gateways called Mobile Edge Computing Devices (MECD). An MECD is designed to manage the local network of end nodes that are associated with a moving object and process data generated by them locally. Local data processing can reduce the amount of data to be sent to the server. In addition, it allows local (neighboring) devices to share information directly without participation of servers. Therefore, the hierarchical system architecture with MECDs improved scalability by managing networks in a distributed manner. The case study in Chapter 3 proved the feasibility of the proposed architecture and provided valuable lessons learned for the further research.

This dissertation also addressed a reachability problem between MECDs and the remote server because of the physical environmental constraints in Chapter 4. The MECD is an interface between the end nodes and servers and thus the unreachability of the MECD is a critical problem in ENS. In the proposed hierarchical system architecture, the MECD-level network is designed for collaboration among MECDs. This helps in not only sharing information, but also sending data to the remote server for devices which have no direct connection to the server. This dissertation proposed using mesh networking for the MECD-level network because a mesh network can deliver reliability by redundant paths. The simulation results showed that any MECD can communicate with the remote server through the mesh network of neighboring MECDs for an intelligent container scenario.

Finally, Chapter 5 provided a new solution to authenticate strangers in ENS. The self-organizing region-based authentication problem was defined and the acoustic feature-based technique was proposed to solve this problem. Experimental studies on acoustic feature extraction were conducted to compare the proposed technique and other time-domain feature extracion techniques. Then, the proposed technique was tested for a smart home scenario and the results showed that this scheme worked in a small region.

6.1. Future Work

In this section, we discuss the future work and research directions.

6.1.1. Energy Efficiency Problem of MECDs

MECDs are mobile gateways that are designed to be carried by a moving object such as a person, vehicle, or animal. To be portable, the MECD needs to be powered by a small battery. Therefore, the power management module of the MECD described in Section 3.2 will be a key component for the future research. For example, by the documents of the Department of Homeland Security, cargo containers make several roundtrips per year and the intelligent container systems are required to support at least one year lifetime with 3,000 hours operations as discussed in Section A.1. During our experimental tests in Section A.2, we confirmed that energy-efficiency will be a key gating factor to develop these battery-powered mobile gateways.

Generally, a power saving mode (also called a sleeping mode) can be used to save energy consumption. However, MECDs are responsible to support communication between end nodes and the remote server and also communication with other MECDs. Therefore, supporting a sleeping mode for MECDs is challenging because the MECD needs to be synchronized with its local end nodes as well as neighboring MECDs. Moreover, the reachability of the MECD-level network may be degraded when some of MECDs are sleeping and can not participate forwarding the packets to the remote server. Therefore, the future work is needed to minimize the energy consumption without degrading the system performance.

6.1.2. Network Analysis and Management for Mesh-networked MECDs

The MECD-level network was evaluated for the intelligent container application in Section 4.2. The simulation in this work assumed that all cargo containers in the network were the same size. In the real world, however, a cargo container can be any size of ISO (International Organization for Standardization) containers. The additional simulation with heterogeneous cargo containers is left for the future work in order to obtain a better validation of the MECD-level networks.

To simplify the evaluation for the intelligent container scenario, it was assumed that the forwarder was the MECD at one of top corners. Another research question for the future work is how to select a proper forwarder that can provide the perfect server reachability for all other neighboring MECDs for a general ESN. The forwarder will consume additional energy to communicate with the remote server and thus forwarder selection needs to be dynamic based on the current energy conditions of the group members in the mesh-networked MECDs. Furthermore, each MECD has different workload because the number of end nodes in its local network and the amount of data generated by these end nodes are different. Therefore, the condition of other types of resources such as available computational capacity and bandwidth can be considered as well.

As discussed in Section 4.2.5, the optimal transmission power of the MECD to minimize the total energy consumption of the MECD-level network depends on the temperature, network size and network density. A dynamic scheduling of the MECD's transmission power is left for the future work.

6.1.3. Self-organizing Region-based Authentication

As we discussed in Section 5.3.5, one of challenging problems in the acoustic feature-based scheme is that sound is sensitive to a distance between devices. The audio-based self-organizing authentication technique could successfully distinguish co-located devices from non co-located devices with a small region. However, in some cases, our authentication technique fails to authenticate devices even if they are in the same region. Therfore, we need a way to extend the coverage for general ENS applications.

A future work could be using multiple types of contextual information to supplement this weakness of the acoustic features. For example, a room can share similar physical conditions such as temperature, light, motion, and wifi signal strength. [99] uses various ambient information including sound, light, color, motion, and wifi to generate a fingerprint. This finger print is used to find logical location information such as "Starbucks", "Walmart", and "Pub". This is similar to our goal except we need to take account of temporal changes. In other words, fingerprinting is to classify contextual information considering spatial relation while the region-based authentication requires classification of contextual information considering both temporal and spatial relation.

Finally, the future work can focus on the heterogeneity problem in ENS. Consider a smart home application (called home automation) that consists of a wide range of home devices from sensors to PCs. Home devices have different capability, for example, PCs have more resources and computational power than sensors. In our acoustic feature-based scheme, there is the trade-off between accuracy and computational complexity. For example, a large size of FFT provides better distinctiveness, but it needs more complex computation and resources. A PC may be able to compute a large size of FFT, but a sensor may not be able to perform it. In addition, using different microphones may affect the results. Therefore, an acoustic feature-based scheme with the heterogeneous devices can be a research problem for the future work.

Heterogeneity is not only related to hardware, but also software. With respect to security, home devices have different security requirements. Fire alarm systems have high security requirement, but toast ovens may need no security support. In [100], Krisnamurthy et al. suggested a classification of security services for a wireless home network as follows: 1) No security (e.g, toaster ovens and refrigerators), 2) Moderate security (e.g., switches on air-conditioning), 3) Wireline equivalent security (e.g., routine telephones), 4) High security (e.g., desktop, laptop computers, and cellular phones), 5) Ultra-high security (e.g., long distance call and video on demand), and 6) Critically high security (e.g., surveillance cameras and security alarm system). Considering this heterogeneity of security requirements, the different security mechanisms need for different devices. A future work can be designing a new system to apply different security levels based on the requirements and capabilities of devices.

APPENDIX A

STUDIES FOR INTELLIGENT CONTAINER SYSTEMS

This chapter includes our survey and experimental studies for intelligent container systems for homeland security as well as global supply chain management.

A.1. Requirements for Intelligent Container Systems

Intelligent container systems should be able to support security requirements driven by government regulation. Therefore, we first explored the intersection of the requirements of the related Department of Homeland Security (DHS) programs used to inform our system design.

An Advanced Container Security Device (ACSD) and a Marine Asset Tag Tracking System (MATTS), two major programs initiated by DHS Homeland Security Advanced Research Projects Agency (HSARPA), have different objectives and requirements with some key overlaps and synergies.

The objective of the ACSD program is to provide the next generation of maritime shipping container security devices with multiple sensing modalities, smart condition monitoring, automated alerting, and advanced communications [101]. The proposed devices must provide a quantum increase in the level of protection and assurance. Primary emphasis is on assuring the physical security of the container including container breach and status of any seals or locks. Secondary emphasis is on detection of certain prohibited cargoes, internal ambient conditions, manipulation or change of state of the contents, and recording container interaction history. The Container Security Device request (CSD) is for the identification of currently available inter-modal shipping container devices [102]. CSD has very similar goals with ACSD program, but requires less functionality.

MATTS [103] focuses on developing and prototyping Tag System using RF (Radio frequency), IR (Infrared) or other modality for shipping containers in the marine environment in order to facilitate universal tracking. The MATTS device is a multi-modal communication gateway for ACSD to provide global remote communications and tracking information for a container. A recently published RFI (Request for Information), *Single Pricing Structure for Global Connectivity to Cellular Data Services To Support Marine Asset Tag Tracking System (MATTS) Communication*, is intended to address the communication requirements of these devices [104]. In order to send alarm and status information, HSARPA proposed to use the existing cellular data services with highly reliable global connectivity for the MATTS devices.

Table X summarizes the functional requirements of CSD, ACSD and MATTS devices. We applied the superset of these functional requirements of CSD, ACSD, and MATTS for our system design.

A.2. Experimental Results and Discussions

A.2.1. RFID Read Ranges

For our test from Singapore to Kaohsiung, Taiwan, extensive analysis was required to determine the correct combination of RFID reader, antenna and RFID tag to support the read ranges required for container door coverage and compliant with the 0.5 Watts ERP requirement. First, we measured the average read ranges of a SkyeTek M9 UHF RFID reader [29] with the maximum output power (27 dBm). According to the datasheet from SkyeTek [29], the M9 reader can reach approximately 3.5m (about 138 inches) with 27 dBm output power and 6 dBi antenna. However, we observed the actual read range from our experiment shown in Table XI was much smaller than the values from SkyeTek. We used three different types of UHF RFID tags (Alien EPC Class 1 Gen 2 [105], Avery

Table X. The Functional Requirements of CSD, ACSD, and MATTS.

Function		**CSD**	**ACSD**	**MATTS**
Sensing	Detection of door opening/closing/removal	Yes	Yes	Yes
	Detection of breaching of the container walls, floor, or ceiling (6 side)	No	Yes	No
	Monitoring container seal or lock status	Option	Yes	Yes
	Sensing loading/unloading of RFID tags	No	Yes	No
	Sensing environmental conditions (temperature, humidity, shock, etc)	Yes	Yes	Yes
	Detection of a person or animal	No	Yes	No
	Tracking and monitoring the location of the container	Option	No	Yes
Alerting	Monitoring the sensors for reportable events	Yes	Yes	Yes
	Notification	Yes	Yes	Yes
Data	Recording and maintaining alert events	Yes	Yes	Yes
	Input and retrieval of data	Option	No	Yes
Commuuication	Local and remote communications	Yes	Yes	Yes
Life time	4.5 trips per year (21 days duration & 7 days loading/unloading)	No	30,000 hours	1 year
Cost	per transit	$50.00		

EPC Class 1 Gen 2 [106], and AWiD ISO 180006B [107]) and three different types of external antennas (Cushcraft S9028PC 8 dBiC [108], Symbol Z1747 6.4 dBdc [109], and Sensormatic 6.75 dBd [110]). Second, we adjusted the output power of these three antennas to 0.5 Watts ERP required for the test from Singapore to Taiwan. Using 0.5 Watts ERP, the average read ranges of all combination of antennas and tags are 10 - 18 inches.

From our experiments, we found that additional investigation and experimentation is required here to ensure the viability of on-board container RFID readers.

Table XI. RFID Read Ranges.

Antennas	UHF RFID Tags		
	EPC Class 1 Gen 2		ISO 180006B
	Alien	Avery	AwiD
Cushcraft S9028PC (8 dBiC)	20 inches	37 inches	70 inches
Symbol Z1747 (6.4 dBdc)	16 inches	37 inches	75 inches
Sensormatic (6.75 dBd)	43 inches	88 inches	86 inches

A.2.2. Energy Consumption

During our test, we confirmed that energy-efficiency will be a key gating factor to scale implementation of these battery-powered devices. Containers make several roundtrips per year, have an extended multi-year lifespan, and do not have an entity with umbrella ownership for maintenance and support. Therefore, these solutions must be standalone and relatively maintenance-free, preferably taking advantage of ambient vibration to harvest power.

Since the MATTS requirements recommend at least one year lifetime with 1 year (3,000 hours) operations and ACSD requires 30,000 hours of annual operation, the components of a wireless sensor network solution, such as the gateway and motes need to be highly energy efficient. Unlike the gateway and motes, the RFID reader needs to support only the 760 hours needed for loading/unloading operations. As we discussed above, we can reduce energy consumption of the RFID reader using sleep mode.

To determine the actual lifetime of our system, we first measured the energy consumption of the MicaZ [25] and TelosB [26] motes used in our prototype implementation. The MicaZ mote with a MTS310 sensor board [28] attached drew 11.25 mA of current consuming a power of 33.75 mW, while the TelosB mote just drew 0.09 mA of current consuming 0.42 mW of power. As expected, this clearly shows that MicaZ is the more energy inefficient of the two. We decided to measure the actual lifetime of the MicaZ mote because the MicaZ motes were used for Reader-mote modules and also deployed inside the container. We programmed a MicaZ mote with a MTS310 sensor board to broadcast a packet once every 10 seconds with sensor readings. After the readings have been broadcast the mote shifts to power saving mode (switching off radio and the sensor board). Each time the mote broadcasts sensor data (in a packet), it also forwards the current voltage level of the battery. At the start of the experiment, the mote had 2 new AA batteries which measured 3.0V. We conducted the experiment till the mote started indicating a low battery around 2.0V mark. Below this mark (2.0V) we were not able to receive the transmitted signals at the distance of 4 meters. We experienced negligible packet loss at the base station which was placed about 4 meters away. Fig. 53 presents the variation in the battery voltage with time. Even though the voltage value was being collected once every 10 seconds, we just show two values per day (one taking in the morning and another in the evening) to illustrate the trend. We can see that the mote lasts about 46 days before the battery on the mote has to be replaced. Obviously, reducing frequency of sensing and broadcasting data can increase the lifetime of MicaZ motes.

At the next step of our experiment, we examined the high energy consumption of the Stargate and RFID reader. The Stargate with MicaZ and AmbiCom CF WiFi card attached drew 406 mA of current consuming a power of 1827 mW. The M9 RFID reader on single reading drew 270 mA of current consuming a power of 1215 mW while a continuous reading mode drew 615 mA of current consuming a power of 2767.50 mW. During a multi-day shipment from Singapore to Taiwan, we used a large (car size) bat-

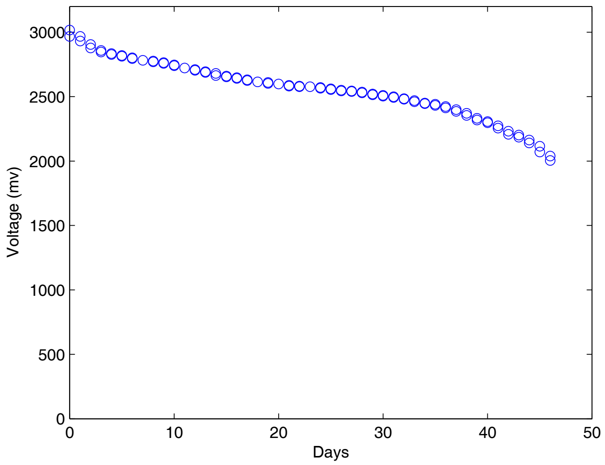


Fig. 53. Battery Drain on Functioning MicaZ Mote over Time.

tery. However, ACSD restricts the size of a container security device so as to not reduce the volume of a container or impact handling. To meet 1 year operations requirements, the energy efficiency should be improved.

A.2.3. Lessons Learned

From our experimental results, we have learned many of the technology constraints as well as broader issues that must be resolved in order to support all the DHS container security requirements.

Current cost requirements of $50.00 US per transit is difficult to achieve with today's technology offerings. However, these RF technologies have been rapidly decreasing in cost while increasing in performance and reliability due to numerous breakthroughs in silicon, component, and solutions design.

From the experimental results, we found the RFID reading ranges were not large enough to cover a container. To address this problem, multiple readers as well as targeted antenna and tag design can be used. In addition, the highly energy efficient devices as well as management (sleep mode, adjusted frequency of reading/sensing/broadcasting) are required.

REFERENCES

[1] F. Adelstein, S.K.S. Gupta, G.G. Richard III, and L. Schwiebert. Fundamentals of Mobile and Pervasive Computing. McGraw-Hill, 2004.

[2] G. J. Pottie and W. J. Kaiser. Principles of Embedded Networked Systems Design. Cambridge University Press, 2005.

[3] S. Fukunaga, T. Tagawa, K. Fukui, K. Tanimoto, and H. Kanno. Development of Ubiquitous Sensor Network. Oki Technical Review, Vol. 71, No. 4, Oct. 2004.

[4] ITU-T Technology Watch Briefing Report Series, No. 4. Ubiquitous Sensor Networks. http://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000040001PDFE.pdf

[5] M. Kim, Y. Lee1, and J. Ryou. What are Possible Security Threats in Ubiquitous Sensor Network Environment? In *Proc. of Asia-Pacific Network Operations and Management Symposium (APNOMS 2007)*, LNCS4773, pp. 437-446, 2007.

[6] C. S. Raghavendra, K. M. Sivalingam, and T. F. Znati. Wireless Sensor Networks, 2nd ed. Springer, 2004.

[7] L. Zhang and Z. Wang. Integration of RFID into Wireless Sensor Networks: Architectures, Opportunities and Challenging Problems. In *Proc. of the 5th Int'l Conf. on Grid and Cooperative Computing Workshops*, pp. 463-469, Oct. 2006.

[8] B. Glover and H. Bhatt. RFID essentials. O'Reilly, 2006.

[9] S.F. Barrett and D.J. Pack. Embedded Systems Design and Applications with the 68HC12 and HCS12. Pearson/Prentice Hall, 2005.

[10] S. J. Kim, G. Deng, S. K.S. Gupta and M. Murphy-Hoye. Intelligent Networked Containers for Enhancing Global Supply Chain Security and Enabling New Commercial Value. In *Proc. the 3rd International Conference on Communication System Software and Middleware (COMSWARE'08)*, 2008.

[11] S. J. Kim, G. Deng, S. K.S. Gupta and M. Murphy-Hoye. Enhancing Cargo Container Security during Transportation: A Mesh Networking Based Approach. In *Proc. the 2008 IEEE International Conference on Technologies for Homeland Security (HST'08)*, 2008.

[12] UN Model Regulations, 14th Rev. Ed.
http://www.unece.org/trans/danger/publi/unrec/rev14/14files_e.html

[13] Dangerous goods.
http://en.wikipedia.org/w/index.php?title=Dangerous_goods&oldid=191950058

[14] K. K. Venkatasubramanian and S. K. S. Gupta. Physiological Value Based Efficient Usable Security Solutions for Body Sensor Networks In *ACM Transactions on Sensor Network*, Volume 6 Issue 4, July 2010.

[15] Kay Romer, Oliver Kasten, and Friedemann Mattern. Middleware Challenges for Wireless Sensor Networks. In *ACM SIGMOBILE Mobile Computing and Communications Review*, Volume 6, Number 2, pages:59-61, October 2002.

[16] I. F. Akyildiz and X. Wang. A Survey on Wireless Mesh Networks. In *IEEE Communication Magazine*, Volume 43, Number 9, 2005.

[17] M. S. Siddiqui and C. S. Hong. Security Issues in Wireless Mesh Networks. In *Proc. of International Conference on Multimedia and Ubiquitous Engineering (MUE)*, April 2007.

[18] S. Ravi, A. Raghunathan, P. Kocher and S. Hattangady. Security in Embedded Systems: Design Challenges. In *ACM Transactions on Embedded Computing Systems (TECS)*, Volume 3, Issue 3, Pages: 461 - 491, August 2004.

[19] S. Aissi, N. Dabbous, and A. R. Prasad. Security for Mobile Networks and Platforms. Artech House, 2006.

[20] M. Bishop. Introduction to Computer Security. Addion-Wesley, 2005

[21] M. Satyanarayanan. Pervasive Computing: Vision and Challenges. In *IEEE Personal Communications*, Volume 8, Issue 4, Pages:10 - 17, 2001.

[22] F. Stajano. Security for Ubiquitous Computing. John Wiley & Sons, Ltd., 2002.

[23] A. K. Jain, R. Bolle, and S. Pankanti. Biometrics: Personal Identification in Networked Society. Kluwer Academic Publisher, 2002.

[24] CrossBow. CrossBow SPB400 Stargate gateway datasheet. http://www.xbow.com/Products/productdetails.aspx?sid=229

[25] CrossBow. CrossBow MicaZ 2.4GHz datasheet. http://www.xbow.com/Products/productdetails.aspx?sid=164

[26] CrossBow. CrossBow TelosB 2.4GHz datasheet. http://www.willow.co.uk/TelosB_Datasheet.pdf

[27] Wikipedia. Zigbee. http://en.wikipedia.org/wiki/ZigBee

[28] CrossBow. CrossBow MTS300/310 Sensor Board. http://www.xbow.com/Products/Product pdf files/Wireless pdf/MTS MDA Datasheet.pdf

[29] SkyeTeck. SkyeModule M9 Developer Kit. http://www.skyetek.com/ProductsServices/DeveloperKits/SkyeModuleDKM9/tabid/289/Default.aspx

[30] Regulatory status for using RFID in the UHF spectrum. http://www.epcglobalinc.org/tech/freq_reg/RFID_at_UHF_Regulations_20070904.pdf

[31] UHF Applications. http://tii.developerconference.ext.ti.com/post-conf/downloads/rfid-tutorial3.pdf

[32] C. Chen and J. Ma. MEMOSEN: Multi-radio Enabled Mobile Wireless Sensor Network. In *Proc. of the 20th Int'l Conf. on Advanced Information Networking and Applications*, 2006.

[33] K. Akkaya and M. Younis. Energy-aware Routing to a Mobile Gateway in Wireless Sensor Networks. In *Proc. of the 2nd Int'l Workshop on Wireless and Ad-Hoc Networks*, Nov. 2004.

[34] M. Shakya, J. Zhang, P. Zhang, and M. Lampe. Design and Optimization of Wireless Sensor Network with Mobile Gateway. In *Proc. of the 21st Int'l Conf. on Advanced Information Networking and Applications Workshops*, May 2007.

[35] K. Bannister, G. Giorgetti, and S. K. S. Gupta. Wireless Sensor Networking for Hot Applications: Effects of Temperature on Signal Strength, Data Collection and Localization. In *Proc. of the 5th Workshop on Embedded Networked Sensors (HotEmNets)*, Jun. 2008.

[36] Wikipedia. Containerization. http://en.wikipedia.org/wiki/Containerization

[37] E. H. Robl. The intermodal container FAQ.
http://www.robl.w1.com/Transport/intermod.htm

[38] Wikipedia. Intermodal container.
http://en.wikipedia.org/wiki/Intermodal_freight_shipping_container

[39] Desktop Engineering. Tracking marine containers for homeland security. http://www.deskeng.com/articles/aaackc.htm

[40] R. H. Katz. Radio propagation. http://www.sss-mag.com/pdf/1propagation.pdf

[41] Wikipedia. Log-distance path loss.
http://en.wikipedia.org/wiki/Log-distance_path_loss_model

[42] W. Xiangli, L. Layuan, and W. Wenbo. An energy-efficiency multicast routing algorithm in wireless sensor networks. In *Proc. of 2008 ISECS Int'l Colloquium on Computing, Communication, Control, and Management*, Vol. 2, pp. 572-576, 2008.

[43] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Eenergy-effieicnt communication protocol for wireless microsensor networks. In *Proc. of the 33rd Hawaii Int'l Conf, on System Science*, Vol. 8, pp. 8020-8029, 2000.

[44] F. Linnarsson, P. Cheng, and B. Oelmann. SENTIO: Hardware platform for rapid prototyping of wireless sensor networks. In *Proc. of the 32nd Annunal Conf. on IEEE Industrial Electronics (IECON 2006)*, pp. 3002-3006, 2006.

[45] R. H. Katz. Radio Propagation.
http://www.sss-mag.com/pdf/1propagation.pdf

[46] R. Shokri. A Low-Cost Method to Thwart Relay Attacks in Wireless Sensor Networks. Project Report, IC-71 Security and Cooperation in Wireless Networks, Doctoral School of the I&C School of EPFL, 2007.

[47] RFIDNews: Savi's RFID Licensing for Cargo Containers. http://www.rfidnews.org/weblog/2007/05/10/savis-rfid-licensing-for-cargo-containers

[48] Port Security: The 5% Myth. http://www.americanchronicle.com/articles/viewArticle.asp?articleID=6780

[49] Privacy Impact Assessment for the Automated Targeting System. http://www.dhs.gov/xlibrary/assets/privacy/privacy_pia_cbp_ats.pdf

[50] C-TPAT: Customs-Trade Partnership Against Terrorism. http://www.cbp.gov/xp/cgov/import/commercial_enforcement/ctpat/

[51] CSI: Container Security Initiative. http://www.cbp.gov/xp/cgov/border_security/international_activities/csi/

[52] RFID and Homeland Security. http://www.aimglobal.org/technologies/rfid/resources/articles/dec03/homeland.htm

[53] L-3 communication. Homeland security and emergency management. http://www.l-3com.com/products-services/productservice.aspx?type=ps&id=787

[54] ODIN technologies. Self-inventorying SMART container. http://www.odintechnologies.com/government-gsa

[55] Savi technology. Smart chain asset and shipment management. http://www.savi.com/capabilities/solutions/smartchain-asm.php

[56] AVANTE technology. Intermodal container and cargo transport. http://www.avantetech.com/products/shipping/

[57] R. Jedermann, C. Behrens, R. Laur and W. Lang. Intelligent containers and sensor networks approaches to apply autonomous cooperation on systems with limited resources Understanding Autonomous Cooperation and Control in Logistics, Springer, Berlin, 2007, pp. 365-392.

[58] D. Hutter. Security in Pervasive Computing. LNCS, Berlin, New York, Springer, 2003.

[59] R. Campbell, J. Al-muhtadi, P. Naldurg, G. Sampemane, and M.D. Mickunas. Towards security and privacy for pervasive computing. In *Proc. of International Symposium on Software Security*, Tokyo, Nov. 2002.

[60] A. A. Pirzada, and C. McDonald. Secure pervasive computing without a trusted third party. In *Proc. of IEEE/ACS International Conf. on Pervasive Services*, 2004.

[61] B. Wang, J. Bodily, and S. K.S. Gupta. Supporting persistent social groups in ubiquitous computing environments using context-aware ephemeral group service. In *Proc. of 2nd IEEE Annual Conf. on Pervasive Computing and Communications*, 2004.

[62] D. E. Denning and P. F. MacDoran. Location-based authentication: grounding cyberspace for better security. Computer Fraud and Security, Elsevier Science Ltd., February 1996.

[63] A.I.G.-T. Ferreres, B. R. Alvarez, and A. R. Garnacho. Guaranteeing the Authenticity of Location Information. In *Proc. of IEEE Pervasive Computing*, vol. 7, no. 3, 2008, pp. 72-80.

[64] C. A. Patterson, R. R. Muntz, and C. M. Pancake. Challenges in Location-Aware Computing. In *Proc. of IEEE Pervasive Computing*, vol. 2, no. 2, 2003, pp. 80-89.

[65] F. Anjum, S. Pandey, and P. Agrawal. Secure Localization in Sensor Networks Using Transmission Range Variation. In *Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005.

[66] T. Kindberg, K. Zhang, and N. Shankar. Context Authentication using Constrained Channels. In *Proc. of the 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[67] L. Lazos and R. Poovendran. SeRLoc: Robust Localization for Wireless Sensor Networks. In *Proc. of the ACM Workshop Wireless Security*, 2004, pp. 21.30.

[68] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *Proc. of the ACM Workshop on Wireless Security*, 2003.

[69] A. Vora and M. Nesterenko. Secure Location Verification Using Radio Broadcast. In *IEEE Transaction on Dependable and Secure Computing*, vol. 3, no. 4, 2006, pp. 377.385.

[70] F. Anjum and P. Mouchtaris. Security for wireless ad hoc networks. Wiley-Interscience, 2007.

[71] J. Elson, L. Girod, and D. Estrin. Time synchronization for Wireless Sensor Networks. In *Proc. of the 2001 International Parallel and Distributed Processing Symposium (IPDPS)*, 2001.

[72] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, 2002.

[73] A. Banerjee, K. Venkatasubramanian, and S. K. S. Gupta. Challenges of Implementing Cyber-Physical Security Solutions in Body Area Networks. In *Pro. of the International Conference on Body Area Networks BodyNets*, Los Angeles, CA, April 2008.

[74] C. Huang, Y. Yang, D. Yang, Y. Chen, and H. Wei. Realization of an Intelligent Frog Call Identification Agent. In *Proc. of the 2nd KES International Symposium (KES-AMSTA 2008)*, Incheon, Korea, March 26-28, 2008.

[75] L. R. Rabiner, and Ronald W. Schafer. Introduction to Digital Speech Processing. Now Publishers Inc, 2007.

[76] S. Chu, S. Narayanan, and C.-C. J. Kuo. Content Analysis for Acoustic Environment Classification in Mobile Robots. Proceedings of AAAI Fall Symposium, Aurally Informed Performance: Integrating Machine Listening and Auditory Presentation in Robotic Systems, 2006.

[77] M. Cowling and R. Sitte. Advanced Signal Processing for Communication Systems: Chapter 3. Recognition of Environmental Sounds using Speech Recognition Techniques. Springer US, 2006.

[78] Cross Correlation Written by Paul Bourke.
http://local.wasp.uwa.edu.au/ pbourke/miscellaneous/correlate/.

[79] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. In *ACM/Kluwer Mobile Networks & Applications (MONET) Journal, Special Issue on Wireless Sensor Networks*, August 2005.

[80] K. Venkatasubramanian, A. Banerjee, S. K. S. Gupta. Green and Sustainable Cyber Physical Security Solutions for Body Area Networks. In *Proceedings of 6th Workshop on Body Sensor Networks (BSN'09)*, Berkeley, CA, June 2009.

[81] CrossBow. Mica2 Datasheet.
https://www.eol.ucar.edu/rtf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

[82] R. M. Newman and Elena Gaura. Size does matter - the case for big motes. In *Proc. of the 2006 NSTI Nanotechnology Conference and Trade Show (Nanotech 2006)*, May 2006.

[83] ENT: A Pseudorandom Number Sequence Test Program.
http://www.fourmilab.ch/random/

[84] R. W. Hamming. Coding and Information Theory. Englewood Cliffs NJ: Prentice-Hall, 1980

[85] The Basics of Monte Carlo Simulations.
http://www.chem.unl.edu/zeng/joy/mclab/mcintro.html

[86] F. Zhu, M. W. Mutka, and L. M. Ni. A Private, Secure, and User-Centric Information Exposure Model for Service Discovery Protocols. In *IEEE Transactions on Mobile Computing*, Vol. 5, No. 4, April 2006.

[87] Wikipedia, Android Dev Phone 1.
http://en.wikipedia.org/wiki/Android_Dev_Phone

[88] Android Developers.
http://developer.android.com/reference/android/media/AudioRecord.html

[89] L. Tan. Digital Signal Processing: Fundamentals and Applications. Elsevier, 2008.

[90] W. Diffie and M. E. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory*, pages 644-654, 1976.

[91] W. Diffie, P.C. van Oorschot, and M.J. Wiener. Authentication and authenticated key exchanges. In *Designs, Codes and Cryptography 2*, pp. 107-125, 1997.

[92] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *Proc. IEEE Symposium on Research in Security and Privacy*, Oakland, May 1992.

[93] V. Boyko, P. MacKenzie, and S. Patel. Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In *Advances in Cryptology: Eurocrypt 2000*, LNCS vol. 1807, Springer-Verlag, pp. 156-171, 2000.

[94] F. Stajano, and R. Anderson. The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks. In *Proc. 7th Security Protocols Workshop*, pp. 172-182, 2000.

[95] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication. in *Proc. IEEE Symposium on Security and Privacy*, pp. 110-124, 2005.

[96] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and Clear: Human-verifiable authentication based on audio. in *Proc. 26th IEEE Intl. Conf. on Distributed Computing Systems*, pp. 10-17, 2006.

[97] C. Soriente, G. Tsudik and E. Uzun. HAPADEP: human-assisted pure audio device paring. In *Proc. 11th Information Security Conference*, 2008.

[98] A. Varshavsky, A. Scannell, A. LaMarca, and E. Lara. Proximity-Based Authentication of Mobile Devices. In *Proc. 9th Intl. Conf. on Ubiquitous Computing*, pp. 253-270, 2007.

[99] M. Azizyan, I. Constandache, and R. R. Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proc. of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, September 20.25, 2009, Beijing, China.

[100] P. Krishnamurthy, J Kabara, and T. Anusas-amornkul. Security in Wireless Residential Network. In *IEEE Transactions on Consumer Electronics*, Feburary 2002.

[101] HSARPA BAA 04-06 Advanced Container Security Device Program.
http://www.hsarpabaa.com/Solicitations/AdvContSecDev_BAA_FINAL_508.pdf

[102] RFI: The Container Security Device. http://www.hsarpabaa.com/Solicitations/CSD-RFI-ver-8.pdf

[103] HSARPA SBIR H-SB04.1-005 Marine Asset Tag Tracking System. http://www.hsarpasbir.com/PastSolicitationDownload.asp#21

[104] RFI: The Single Pricing Structure for Global Connectivity to Cellular Data Services to Support Marine Asset Tag Tracking System (MATTS) Communications. http://www.hsarpabaa.com/main/RFI-GCCDS.htm

[105] Alien EPC Class 1 Gen 2 RFID Tags. http://www.barco.cz/data/products/download/ALL-9440Gen2Datasheet.pdf

[106] Avery EPC Class 1 Gen 2 RFID Tags. http://www.rfid.averydennison.com/_media/us/pdf/datasheets/Portfolio.pdf

[107] AWiD. http://www.awid.com

[108] Cushcraft S9028PC Circularly Polarized Panel Antenna. http://www.cushcraft.com/support/pdf/S9028PC12NF.pdf

[109] Symbol Fixed RFID Reader Antennas. http://www.symbol.com/products/rfid-readers/rfid-antenna5

[110] Sensormatic Omniwave Antenna (EPC Class 1 Circular). http://www.sensormatic.com/SensormaticGetDoc.aspx?FileID=9519