ISSN: 1945-3078 (Print), 1945-3086 (Online) Volume 1, Number 5, December 2009

# Wireless Sensor Network

Chief Editor : Kosai Raoof

www.scirp.org/journal/wsn/

Scientific Research

# Journal Editorial Board

**ISSN 1945-3078 (Print) ISSN 1945-3086 (Online)**

**http://www.scirp.org/journal/wsn/**

*Wireless Sensor Network*, **2009, 1, 365-495**
Published Online December 2009 in SciRes (http://www.SciRP.org/journal/wsn/).

# TABLE OF CONTENTS

**Volume 1 Number 5** **December 2009**



# Wireless Sensor Network (WSN)

# Journal Information

## SUBSCRIPTIONS

The *Wireless Sensor Network* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

E-mail: service@scirp.org

**Subscription rates:** Volume 1 2009
Print: $50 per copy.
Electronic: free, available on www.SciRP.org.
To subscribe, please contact Journals Subscriptions Department, E-mail: service@scirp.org

**Sample copies:** If you are interested in subscribing, you may obtain a free sample copy by contacting Scientific Research Publishing, Inc at the above address.

## SERVICES

**Advertisements**
Advertisement Sales Department, E-mail: service@scirp.org

**Reprints (minimum quantity 100 copies)**
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: service@scirp.org

## COPYRIGHT



## PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: wsn@scirp.org

*Wireless Sensor Network*, **2009, 1, 365-369**
doi:10.4236/wsn.2009.15045 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# An Application of Context Middleware Based on Fuzzy Logic for Wireless Sensor Networks

**Ye NING[1,2], Ruchuan WANG[2,3], Shouming MA[2], Zhili WANG[1]**
[1]*Department of Information Science, Nanjing College for Population Program Management, Nanjing, China*
[2]*Institute of Computer Science, Nanjing University of Post and Telecommunications, Nanjing, China*
[3]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China*
*Email*: *cathery*163@163.*com*
*Received May* 25, 2009; *revised July* 22, 2009; *accepted July* 25, 2009

## Abstract

The research of context-aware computing based on wireless sensor network (WSN) aims at intelligently connecting computers, users, and environment. So its application system should be flexibly adaptable to dynamic changes of context and application requirements and proactively provides the information satisfied with current context for users. The middleware can be very effective to provide the support runtime services for context-aware computing. In this paper we propose middleware architecture for context processing. This architecture is based on fuzzy logic control (FLC) system for context reasoning and sensor fusion. We propose a formal context representation model in which a user's context is described by a set of roles and relations correspond to a context space. A middleware prototype has been developed, which detect tourist' physical context and provide reminding. The experiments prove that the model and approach proposed are feasible.

**Keywords:** WSN, Context-Aware, Middleware, FLC, Pervasive Computing

## 1. Introduction

Context-aware has recently become a hot topic in the areas of pervasive computing, which was introduced by Schilit and Theimer [1]. With the development of wireless sensor networks (WSNs), the space and physical context information can be obtained by a large number of sensor nodes. How to intelligently integrating multiple sensors and sensor fusion is a crucial technology for context processing in WSNs.

To avoid increasing complexity and allow the users is to concentrate on his tasks, applications and services must be aware of their contexts and automatically adapt to their changing contexts-known as context-awareness.

In most situations, humans react opportunistically, switching among a set of possible goals, abandoning and adding new goals in response to events and opportunities. One of the most difficult challenges in designing context aware systems is to recognize and allow for such unpredictable behavior [2]. In this respect, middleware can be very effective to provide the support if they can reduce the effort required to develop distributed software and runtime services for applications with the abovementioned characteristics, in addition to providing the normal services, such as interoperability, location transparency, naming service, etc [3].

In this paper, we present the design and implementation of a middleware approach for context-awareness, and adopted fuzzy logic [4] as an intelligent reasoning method for selecting data dissemination protocols in the design of the decision mechanism.

The remainder of the paper continues as follows: in Section 2 discusses related work and main problem. In Section 3 we describe our formal context representation model and the structure of FLC system for Context processing. Section 4 describes an application we have developed for our middleware prototype. Finally, the paper ends with conclusion in Section 5.

## 2. Relate Work and Problem Formulation

We are not aware of any integrated middleware platform that aims to achieve all of the goals described above.

As early as in 2002, Huadong Wu etc. proposed a contextual information model and built a generalize-able sensor fusion software architecture that can support mapping sensors' raw output data into the contextual information hierarchy [5].

The Gaia project [6] developed at the University of Illinois is a distributed middleware infrastructure that provides support for ubiquitous computing.

The EasyLiving project [7] from Microsoft focuses on development of an architecture and technologies for intelligent environments.

Work at Illinois has developed the Universal Interoperable Core (UIC) which is a reflective middleware platform designed for handheld devices [8].

Recent research work has focused on middleware extensions for pervasive computing by standardizing on web services and service discover-oriented technology [9].

To sum up, it is important that middleware for context-aware application should be able to fulfill the following functionalities and objectives.

1) The middleware architecture should be modular and extensible.

2) The middleware should be based on a service-oriented architecture, in which each application and device is represented as a service entity.

Based on the above-mentioned study, we developed a middleware for pervasive computing, which adopted fuzzy logic as context processing method.

## 3. Fuzzy Logic Based Context Processing

### 3.1. A Formal Context Representation Model

We use a context space theory model shown in [10] for model fundamental nature of context and enable context and situation awareness for context processing. Our context model gives a common representation for context that all entities in the environment use of pervasive computing. Instead, it provides a common base on which various reasoning mechanisms can be specified to handle context.

**Definition 1:**

We define a attribute value $a_i$ as any type of data that is associated with Contextual information, include physical contexts, environmental contexts, informational contexts, personal contexts, social contexts, application contexts and system contexts [11].

**Definition 2:**

For a context state $C_i^t = (a_1^t, a_2^t, \cdots a_N^t)$, defined over a collection of N attribute-values, where each value $a_i^t$ corresponds to an attribute $a_i$'s value at time t.

**Definition 3:**

Let $\mathrm{C}_{\mathrm{space}}$ be a context space in an environment of pervasive computing, describes the application's current state in relation to chosen context. In our model, context spaces are represented as first-order calculus. The basic model has the form of *predicate*(*subject*,$C_{att}$), in which

- $subject \in S^*$: set of the expression entity of context information, e.g. visitor, location, etc.

- $predicate \in V^*$: set of predicate name, e.g. is located in, has status, etc.

- $\mathrm{C}_{\mathrm{att}} \in C^*$: set of all values of context state in $S^*$, e.g. warm, cold, open, close, empty, etc.

For example, Location (Marry, laboratory) means Marry is located in the laboratory.

The basic context model can be extended to form a set of contexts by combining the predicate and Boolean algebra (union, intersection and complement).

For example,

$\mathrm{BodyTemperature(Marry,38)} \wedge \mathrm{Pluse(Marry,90)}$ represents physical signs about Marry.

### 3.2. Fuzzy Logic-Based Context Processing

Fuzzy logic was proposed by Lotfi A. Zaheh in 1965 [12] to emulate the way that the human brain processes uncertainty, uncertainty, imprecision, and vagueness. Fuzzy logic is suitable for context management because it is capable of processing imprecise and unreliable information coming from pervasive computing, and it can describe a problem in a common sense format in which expert knowledge, instead of differential equations, can be applied [13].

For $\forall a_i, a_i \exists \mathrm{C}_{\mathrm{space}}$, be an input is applied to a FLC system, the inference engine computes the output set corresponding to each rule. On analyzing the context processing of various potential services, we use singleton fuzzification and "IF-THEN" rules of form [14].

Rl: IF $a_1$ is $F_1^l$ and $a_2$ is $F_2^l$ and $\cdots$and $a_N$ is $F_N^l$, THEN y is $G^l$.

Assuming singleton fuzzification, when an input $\mathrm{A}^{'} = \{a_1^{'}, \cdots, a_N^{'}\}$ is applied, the degree of firing corresponding to lth rule is computed as

$\mu_{F_1^l}(a_1^{'}) * \mu_{F_2^l}(a_2^{'}) * \cdots * \mu_{F_N^l}(a_N^{'}) = \mathrm{T}_{i-1}^{N} \mu_{F_i^l}(a_i^{'})$ Where * and T both indicate the chosen t-norm. In this paper, we focus on the height defuzzifier and used trapezoidal membership ship functions to represent low, high, very strong, very weak to represent moderate, medium, strong, and weak.

In this paper, we are primarily interested in developing middleware based on the structure of FLC mentioned above. We design a FLC system for travels services, which is one of component in our pervasive computing prototype Tourist Reminder.

The FLC system in this paper receives context information from sensor equipments as the inputs of the FLC and the fuzzification module converts inputs into fuzzy linguistic variable inputs.

On analyzing the data requirements of travels services, four linguistic variables were defined, representing the physical sign of tourist. The membership functions of these input parameters of the fuzzy logic are illustrated in Figure 1.These member functions have been determined based on the simulation result. The labels in the fuzzy variables are presented as follows.

- Age= {infant, youth, midage, old};
- BodyTemp= {Normal};
- Pulse= {t1, t2, t3, t4}; //different intervals of normal pulse
- R= {N (Normal), L (Lower), H (Higher)};// degree of reminding for physical signs
- Based on above fuzzy variables, we can define fuzzy IF-THEN rules such as follows.

1) If (Age is infant) and (Pulse is t1) then (R is N)
2) If (Age is infant) and (Pulse is t2) then (R is H)
3) If (Age is infant) and (Pulse is t3) then (R is H)
4) If (Age is infant) and (Pulse is t4) then (R is H)……

A sample fuzzy calculation at a value of context information point is described in Figure 2.

## 4. Fuzzy Middleware Protoype Implementation

We have evaluated the context processing mechanism that based on fuzzy logic system by developing a simple prototype application called Tourist Reminder. As Figure 3 shows, we use medical sensors (body temperature sensor, pulse sensor and blood oxygen sensor, etc) and GPS/RFID in detecting tourist's physical signs/location anywhere and anytime. All sensor data are transmitted by ZigBee wireless sensor nodes to the middleware running on PC or PDA for analyzing and providing reminding message to tourist.

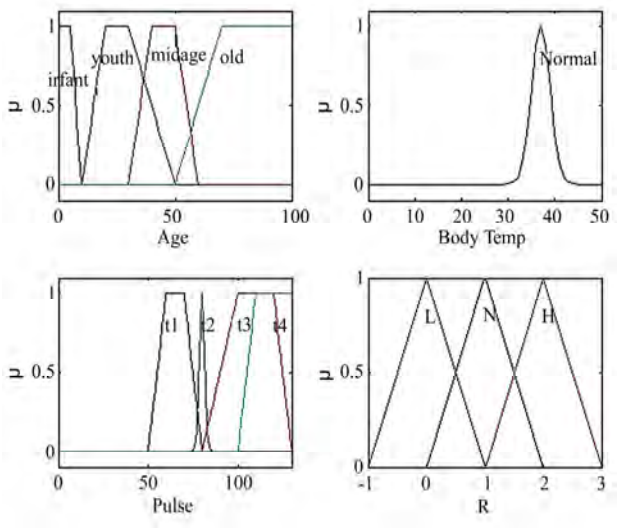



**Figure 1. Membership functions for input context (physical signs).**

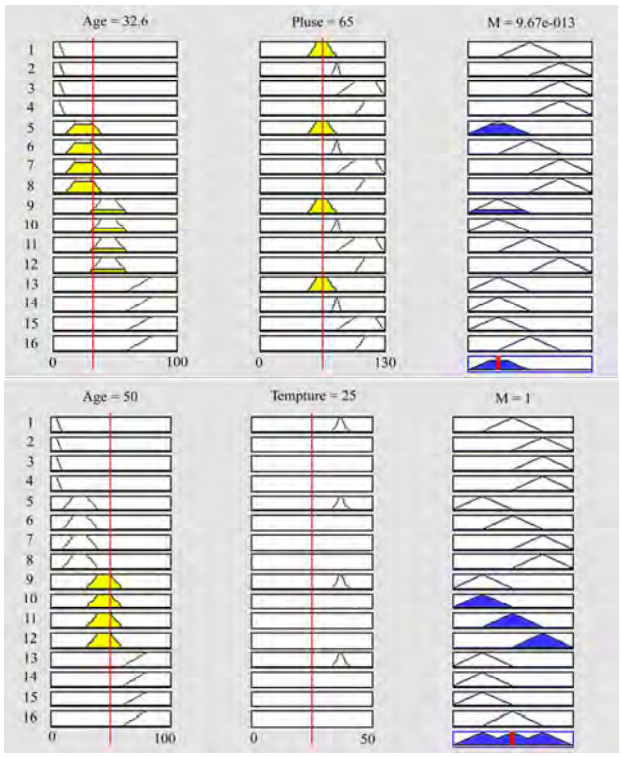



**Figure 2. A sample fuzzy calculations.**

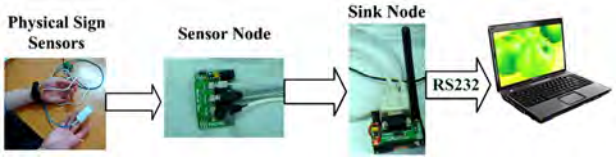



**Figure 3. The process of prototype application.**

### 4.1. Middleware Architecture

To create Tourist Reminder, we developed generic reference architecture applicable to pervasive computing space. As Figure 4 shows, the middleware contains separate physical, sensor platform, service, knowledge, context management, and application layers. Physical Layer: contains a variety of sensors and actuators which monitor and gather context information about the pervasive environment.

- Context Acquire Layer: integrates the sensors and actuators from the layer beneath and export their service representations to the layers above, which includes query processing component that can process filters and queries for sensor readings sent to it from the query processor in the service layer.
- Context service layer: contains the Open Services Gateway Initiative (OSGi) framework, which maintains leases of activated services. The layer provides the service discovery, composition, and invocation mechanisms for applications to locate and make use of particular

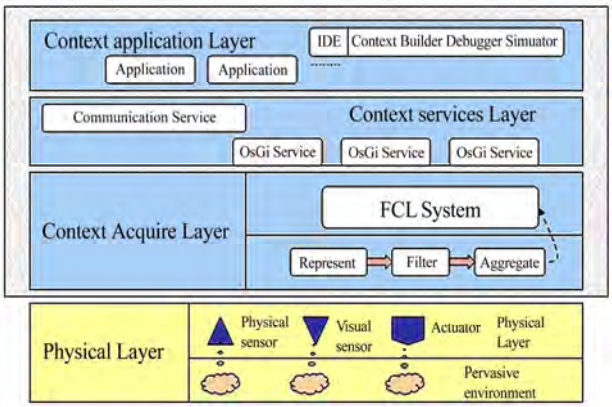
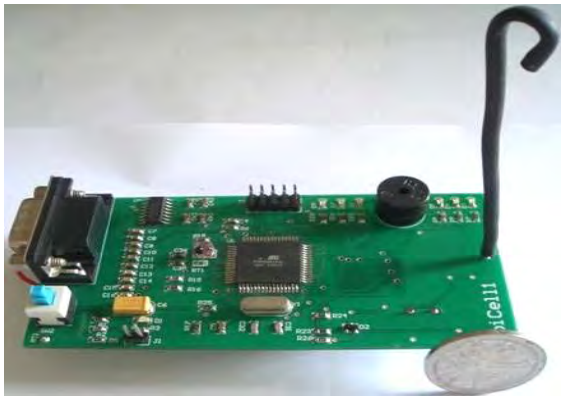



**Figure 4.The architecture of middleware.**



**(a) UbiCell node**

**(b) physical sign sensors and PC**

**Figure 5. The hardware of Tourist Reminder.**

sensors or actuators. It holds the registry of the software service representation of all sensors and actuators connected to the hardware nodes and FLC system as inference engine.

- Context application layer: sits at the top and consists of the execution environment that provides an API to access and control sensors, actuators, and other services. It contains a service authoring tool to enable rapid and efficient development and deployment of services and applications.

### 4.2. Middleware Prototype Implement

We develop sensor node called UbiCell[15], which ingrate models of GPS and physical sign sensors. The hardware of middleware prototype is composed of PC, UbiCell and sink nodes (Figure 5).

In this project, the develop platform of software was based on J2sdk1.4+Eclipse3.2.

As shows in Figure 6, the main function of Tourist Reminder include subscribed service, query service, message reminding, GIS location, etc.

- Client as the user of tourist service subscribes the services to Server (middleware) according to the ID about tourist.
- After sampling and aggregating the physical context about subscribed tourist, Server realizes reasoning based FLC system and returns the relevant messages of services (reminding/query).

Figure 7 is one of the capture images about sampling data of body temperature. As shows in Figure 7, the trend of data is stability.

## 5. Conclusions

Context-aware computing has been a key issue for pervasive computing based on WSN. In a pervasive computing environment, Services should be intelligent

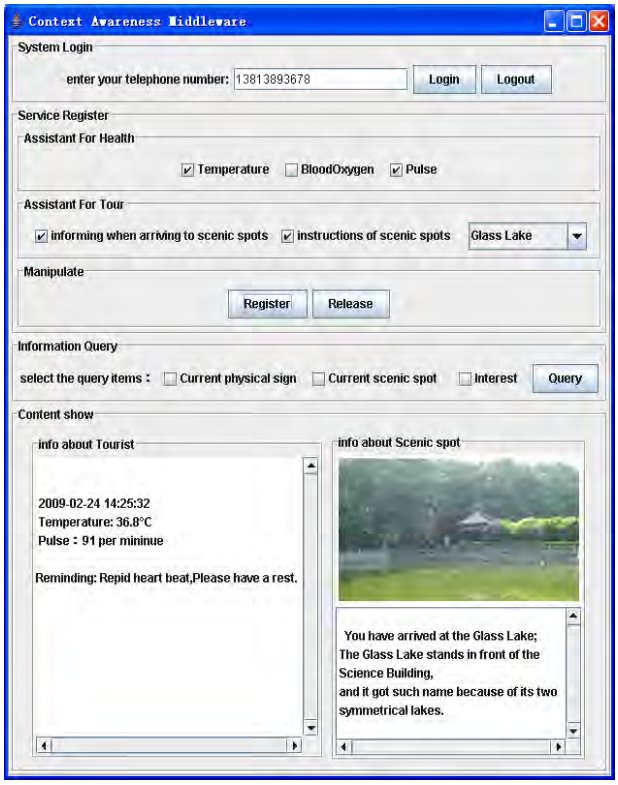



**Figure 6. The interface of Tourist Reminder.**

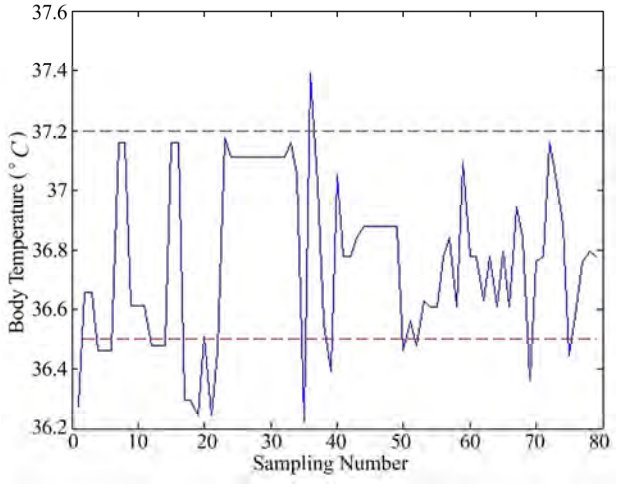



**Figure 7. The capture image of sampling data.**

enough to understand the real world. Our study in this paper demonstrates that fuzzy logic based middleware is feasible for facilitate context processing. A key feature of our model is the presence of FLC based context architecture.

The work of this paper is a part of our ongoing middleware prototype for pervasive computing which provides the reminding service to tourist. Now we furthering work are to apply in practice trade.

## 6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 60773041), Science Foundation of Jiangsu High School (No. 09KJB510020) and sponsored by Qing Lan Project.

## 7. References

[1] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," IEEE Network, Vol. 8, No. 5, pp. 22–32, 1994.

[2] P. Dourish, "What we talk about when we talk about context," Personal and Ubiquitous Computing, Vol. 8, No. 1, pp. 19–30, 2004.

[3] A. Helal, "Programming pervasive spaces," The Standards and Emerging Technologies Department, IEEE Pervasive Computing magazine, Sumi Helal, Dept. Editor, Vol. 4, No. 1, January–March 2005.

[4] H. J. Zimmermann, "Fuzzy sets theory and its applications," Second, revised edition, Kluwer Academic Publishers, 1991.

[5] H. Wu, M. Siegel, and S. Ablay, "Sensor fusion for context understanding," Instrumentation and Measurement Technology Conference, IMTC/2002, Proceedings of the 19th IEEE, Vol. 1, pp. 13–17, May 2002.

[6] M. Roman, C. K. H., R. Cerqueira, *et al*, "Gaia: A middleware infrastructure to enable active spaces," IEEE Pervasive Computing, pp. 74–83, 2002.

[7] Microsoft Research, "Easy living," 2009, http://research.microsoft.com/easyliving/.

[8] M. Roman, F. Kon, and R. H. Campbell, "Reflective middleware: From your desk to your hand," IEEE DS Online (Special Issue on Reflective Middleware), 2001.

[9] S. Ou and K. Yang, "An effective offloading middleware for pervasive services on mobile devices," Pervasive and Mobile Computing, Vol. 3, No. 4, pp. 362–385, 2007.

[10] A. Padovitz, *et al*, "An approach to data fusion for context awareness," Fifth International Conference on Modelling and Using Context, CONTEXT'05, Paris, France, July 2005.

[11] M. Korkea-aho, "Context-aware applications survey," 2009, http://www.hut.fi/~mkorkeaa/doc/context-aware.html.

[12] L. A. Zadeh, Fuzzy sets, Information and Control, pp. 338–353, 1965.

[13] M. Marin-Perianu, C. Lombriser, *et al*, "Distributed activity recognition with fuzzy enabled wireless sensor networks," Technical Report TRCTIT-07-68, Enschede, September 2007.

[14] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," International Journal Man-Machine Studies, Vol. 35, pp. 385–455, 1981.

[15] Wireless Sensor Network Research Center of Nanjing University of Post and Telecommunications, "UbiCell product manuals," 2009, http://www.wsns.net.cn.

*Wireless Sensor Network*, **2009, 1, 370-382**
doi:10.4236/wsn.2009.15046 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Target Tracking with QoS Support in Large Wireless Sensor Networks

**Ghasem Naddafzadeh SHIRAZI, Peijie WANG, Xiangxu DONG, Chen Khong THAM**
*Department of Electrical and Computer Engineering*
*National University of Singapore, Republic of Singapore*
*Email*: {*naddafzadeh, g*0700293, *dong*07, *eletck*}*@nus.edu.sg*
*Received July* 15, 2009; *revised August* 10, 2009; *accepted August* 17, 2009

## Abstract

Quality of Service (QoS) is important in the application of target tracking in wireless sensor networks (WSNs). When a target appears, it will trigger an event from one or more sensors. A target can only be accurately detected if a certain number of event packets are received by the sink in a predetermined detection time interval. In this paper, we propose a buffer management scheme based on event ordering to achieve QoS. We also propose a directional QoS-aware routing protocol (DQRP) for the dissemination of the event ordering list. After the dissemination, a priority queue buffer management scheme is used to ensure QoS. Our buffer management scheme works in conjunction with DQRP to ensure accurate as well as energy-efficient target detection in the presence of multiple targets. The novelty of our network architecture is that a distributed admission control scheme is implemented on each node based on a geographic routing algorithm. In our scenario, a target can only be accurately detected if a certain number of event packets are received by the sink in a predetermined detection time interval. Our main performance metric is the number of targets/events being detected. Our protocol maximizes the number of targets being detected.

**Keywords:** Target Tracking, QoS, Multi-Sink Wireless Sensor Networks.

## 1. Introduction

With the advancements in wireless communications and the development of small electronic sensing devices, wireless sensor network (WSN) technology was greatly developed in the past few years. A WSN comprises a large number of densely deployed sensing devices to sense the phenomenon or the occurrence of an event. Sensor nodes may be required to do data aggregation and fusion locally. More importantly, the sensor nodes are required to report their measurements to the sink. Similar to the traditional end-to-end networks, communications in WSN suffer from delay and loss. Quality of Service (QoS) support is required to ensure the performance of a network. The QoS in traditional computer networks is generally defined as the performance level of a network service offered to the user, therefore QoS centers on network quantities such as delay, loss and reliability.

However, the communication in WSNs is data-centric and non end-to-end. In WSNs, the main consideration is the number of packets the sink receives about an event and not how many packets the sink receives from any particular sensor node. Therefore, QoS in WSNs is different from the traditional end-to-end networks. Furthermore, depending on different applications, the QoS in WSNs may also include coverage, accuracy, etc. Moreover, minimizing energy consumption is another important consideration in WSNs. Once sensor nodes are deployed in the physical area, it is hard to recharge them. Since replacement is costly, energy efficiency is highly demanded in WSNs.

## 2. Related Work

QoS in WSNs contains a wide range of issues. In the literature, there are many papers focusing on the explicit definition of QoS in WSNs as well as the specific implementation of QoS. Reference [2] serves as a good survey of the QoS support in WSNs. It generally describes the QoS in WSNs in two different perspectives; one is application-specific QoS and the other is network QoS.

Depending on different applications, the application-specific QoS can be defined using parameters such as coverage, measurement accuracy, and optimum number of active sensors.

From the perspective of network QoS, the main purpose is to efficiently utilize the network resources to deliver the QoS-constrained data. The QoS metrics can be defined as transmission delay or packet loss. Reference

[1] proposes using energy efficiency, system lifetime, latency, accuracy, fault-tolerance and scalability to evaluate sensor network protocols. It also defines the architecture of a sensor network and many basic models in WSNs, such as the communication models, the data delivery models and the network dynamic models. These models can help us to define the QoS metrics more precisely. In [3], the authors present a QoS control for WSN. They assume a broadcast channel for the base station to dynamically control the number of active sensors in virtue of the Gur Game mathematical paradigm. The QoS here is defined as the optimum number of active sensors in the network. Reference [4] devotes to quantify the tradeoff between power conservation and quality of surveillance in target tracking WSNs. It also provides guidelines for efficient deployment of sensor nodes for target tracking applications.

Our motivation is to perform an accurate detection and tracking of moving objects in a WSN where multiple events occur with variable inter-arrival times. We consider a target tracking application in which the events are *appearance* or *movement* of the targets. Some previous works on target tracking focus on the coverage problem [4,5]. A general assumption has been made that once the sensing range of senor nodes covers the trajectory of the moving target, the target can be tracked. Reference [6] proposes to use quality of monitoring (QoM) to ensure a high reporting accuracy in the presence of noises and signal attenuation. The authors of [6] state that both QoM and coverage need to be taken into account when solving target tracking problem.

Instead of restricting the tracking problem in coverage or QoM, we establish the problem from the perspective of network consideration. Accurate tracking and location estimation of a mobile target may require a minimum number of packets to be received by the sinks. It is not enough just to ensure coverage. Loss of data packets will result in loss of accuracy when estimating the actual locations of the targets. Therefore, QoS support is required in order to ensure accurate target tracking.

## 3. Challenges in Providing QoS in Target-Tracking WSNs

### 3.1. Definition of QoS

We explicitly define the QoS requirement in the application of target tracking in WSNs as:

*A target can only be accurately detected if a certain number of event packets are received by the sink in a predetermined detection time interval.*

Our main performance metric is the *number of targets/events being detected*. Our protocol aims to *maximize* the number of targets detected with the events satisfying the user's QoS requirements.

### 3.2. Problem Description

Traditional QoS schemes used in computer networks, like Intserv or Diffserv [7], do not work well in our problem. This is because the target moves, and the sensor node which is responsible for transmitting the event packets changes. Therefore, it cannot be known in advance which sensors or the resources required to maintain QoS. Besides, reservations of resources require the knowledge of capacity of the network which is difficult to achieve in dynamic WSNs. To solve the above problems, we propose a QoS-aware network architecture designed to be used in multi-hop and multi-sink WSNs. Our network architecture consists of two components, a buffer management scheme and a novel QoS-aware routing algorithm named **D**irectional **Q**oS **R**outing **P**rotocol (DQRP). The main objective of DQRP is to support the use of a distributed admission control scheme for WSNs.

We further illustrate the problem of target tracking if no QoS support is used by carrying an experiment using MICAz motes. The scenario is illustrated in Figure 1. There are two source sensor nodes which have detected one target each and they are sending packets at a rate of 20 packets/s to the sink through an intermediate sensor node. The size of each data packet is 20 bytes excluding the various headers. The sensor which detects event 1 sends data from t=0 to t=180 while the sensor which detects event 2 sends data from t=60 to t=240. The sampling interval is 5 seconds and an event is only detected when 90% of the data packets are received between 2 sampling periods.

Figure 2 shows the results obtained. From the graph, we note that event 1 can always be detected when event 2 has not occurred. When there are two events, there are times when none of the events can be successfully detected (e.g. from t=60 to 80 and from t=160 to 180) although the total throughput remains fairly high. This illustrates the need for a buffer management scheme to give priority to certain events to ensure that we can maximize the number of events detected.

Our buffer management scheme on each sensor node can do priority discarding of the data packets whenever it encounters congestion by giving higher priority to the event which happens earlier. The main remaining challenge

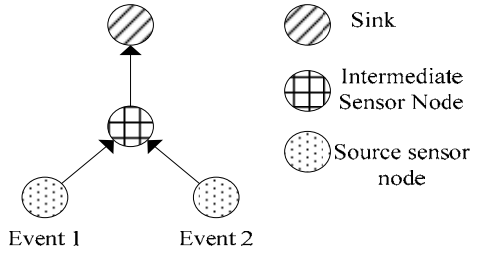



**Figure 1. Scenario setup.**

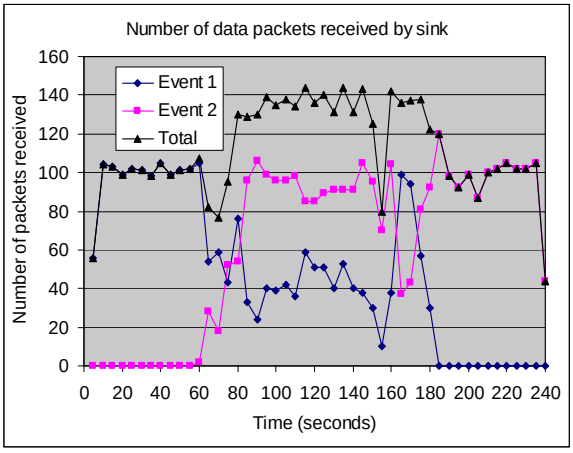



**Figure 2. Experimental results.**

is how to construct an event ordering list at each sensor node in a distributed way.The solution would be simple if global information is available. Each event packet can be associated with a global timestamp which is the time in which the event first occurs. When congestion happens, the packet with largest timestamp is dropped because we give priority to earlier events. However, this requires global clock synchronization which is hard to achieve in practical WSNs. Since WSNs are deployed over long periods of time, time drifts are a very serious problem because cheap and inaccurate clocks are usually used in sensor nodes.

Another possible solution for this problem is to use broadcasting. We can divide the communication into two phases. In the first phase, whenever a new event happens, the sensor node that detects it first broadcasts a new control message indicating that a new target has appeared and therefore there is a new event. All the nodes which receive the broadcast control message would insert the new event into its event ordering list. All the events that arrive later would have lower priorities. In the second phase, data packets are transmitted. Whenever a node encounters congestion, it would discard the data packet with lower priority, i.e. discard the data packet from the events which happen later.

The authors of [8] propose an efficient broadcast scheme for WSN called Broadcast Protocol for Sensor networks (BSP). The BPS uses an adaptive-geometric approach to reduce the number of retransmission by maximizing each hop length. In ideal BPS, the whole network area is covered by numbers of identical hexagons, where the length of the side of the hexagon is the node transmission range. Reference [8] tries to avoid these retransmissions by defining a transmission threshold *Th*. If a node overhears another node within distance *Th* has transmitted one packet, it would not retransmit the same packet. *Th* is a very important parameter in BPS as it represents the tradeoff between numbers of retransmission (redundancy) and delivery ratio (reliability).

In WSNs, the limited energy is usually expected to be used to transmit useful information (data packets). Therefore, the redundancy in control messages is not a good solution especially in our case where multiple events can happen intermittently. If the new events' arrival rate is high, it would lead to a large amount of redundant control messages occupying the nodes' buffers; therefore the number of events that can be correctly detected would reduce significantly in this case due to congestion.

Another reason we use unicast transmission instead of broadcasting in our proposal is that unicast may be more reliable than broadcasting. Assume the nodes density is D, the channel loss probability is $p_l$, and let the transmission range of sensor node be *t*. Suppose the unicast mechanism retransmit a packet for at most k times in case of packet loss. Thus the probability of a packet is lost at a node is given by

$$p_{bl} = p_l^{a_n} \tag{1}$$

in broadcasting, where $a_n = \pi t^2 D$ is the expected number of neighboring nodes of the destination node, and

$$p_{ul} = p_l^{k} \tag{2}$$

in unicast. Therefore, it is clear that unicast will be more reliable than broadcasting if $k > a_n$ is satisfied.

Our solution avoids the use of broadcast packets by making use of geographic routing algorithm to disseminate event ordering information using data packets. Each data packet consists of the event ID and therefore no additional control messages are required. Geographic routing uses nodes' locations as their address, and forwards packets in a greedy manner towards the destination. The greedy manner means that a packet is only forwarded to a node when it is closer to the destination than the current one. References [10] and [11] are two well known proposals on geographic routing algorithms. Our proposed geographic routing algorithm makes use of an angle to implement the greedy routing. We ensure that the density of our network is sufficient so greedy geographic routing works most of the time. The choice of angle would ensure that most of the nodes in the WSN covered area are able to receive packets from every event. These data packets containing the event ID are an important medium for the nodes to know the right ordering of events.

## 4. System Model

Our main goal is to maximize the number of targets being detected in a WSN where multiple targets appear with different inter-arrival times. If the rate of packets received is less than the desired rate, the event is considered *lost*, and the system is not able to detect it. In the

following subsections, we describe different aspects of our model, namely application, medium access control (MAC), and physical (PHY) levels, as well as transport, routing, and scheduling protocols. A distributed admission control scheme is then proposed in Section 5.

Each target is uniquely identified by a target ID, *i*. The $i^{th}$ target, $T_i$, causes packet generation at a rate of $r_i$ in the sensor nodes which detect it. Each event requires a packet delivery requirement of $d_i$. Therefore a minimum packet rate of $d_i r_i$ is required at the sink for detection of the event.

The choice of MAC protocols and physical layer characteristics (PHY) affect the capacity of the network, *C*. However, it does not affect our goal of maximizing the number of detected events given that the network capacity is *C*. A higher *C* would lead to more events detected and vice versa. Therefore, our QoS network architecture does not assume the use of any particular MAC or PHY.

Our novel routing algorithm, DQRP, ensures that each packet takes a different route to the sink such that after $\beta$ packets have been sent by the source, all the sensors in the network will learn of the new event. Angle geographic routing is used and each data packet is given an angle of routing $x$ degrees. DQRP is further explained in Section 5.

Each node maintains *b* buffers. Congestion occurs when the number of packet arrivals exceeds the number of buffers available, therefore buffer overflow occurs. Each node maintains an event ordering list. Earlier events are given higher priority when deciding which packet to drop in the event of buffer overflow.

We consider a multiple sink architecture in which sinks are able to share their received data. Therefore, sensor nodes can choose any of the sinks as the destination. Moreover, we assume that the location of sinks and neighboring sensors (i.e. sensor nodes within the transmission range) are known. The arrival of events models a Poisson Arrival Process. The targets stay in the WSN for a period which follows an exponential distribution. Only one sensor sends data packets to a sink for an event at any given time.

## 5. Proposed QoS-Aware Network Architecture

The main objectives and desired properties of our proposed protocol are given as follows:

1) Path diversity: failure of a sink or failure of any sensor node degrades performance gracefully. Any malicious or selfish sensor node can only degrade performance gracefully.

2) Maintaining QoS: in event of congestion, earlier events will get higher priority. This means that if event *i* occurs before event *j*, if event *i* does not achieve the required QoS, event *j* also does not achieve the required QoS.

3) Event Priority: In the event of congestion, event *i* will take priority over event *j* if event *i* occurs before event *j*.

A formal definition of an event being detected is given as: *An event is detected at time t if $d_i r_i$ packets are received in the interval [t-$t_{min}$, t] where $t_{min}$ is the sampling interval, $d_i$ is the delivery ratio required by the user and $r_i$ is the sending rate of the event.*

This means that an event *i* is only detected at time *t* if the sinks receive a minimum number of packets defined by the QoS requirements of the user. The main performance metric is the number of events detected at a sampling time and our goal is to maximize the number of events detected. In addition, delay, throughput (total amount of traffic received by the sinks), and energy consumption should be considered as auxiliary performance metrics. Note that high throughput does not imply that the number of events detected is high, as illustrated in Figure 2.

### 5.1. Priority Buffer Management

We need to implement a priority buffer management scheme in every node to ensure that if congestion occurs and if we have the global event ordering list, we can decide to give priority to certain events. Our priority buffer management scheme is very efficient as insertion and removal takes O(1) time.

### 5.2. DQRP Protocol

In this paper, we propose using a novel directional QoS routing protocol (DQRP) to disseminate event ordering information. In DQRP, the detection of an event is reported to the entire network in order to provide QoS management in case of congestion. More specifically, when a target is first detected, the corresponding sensor sends different data packets via different routes to the multiple sinks. These routes are determined in such a way that all nodes are informed about the existence of the event after a certain time. The rationale behind informing the entire network is to provide the network with correct global event ordering, and hence enabling distributed admission control in the network.

To ensure that the maximum number of nodes can be informed in an efficient way, it is required to find routes to the sinks with the following two properties:

P1. Each node should be at least in one route.

P2. Number of shared nodes between every two routes should be minimized.

In other words, a new data packet should inform a new node about the event. Figure 3 shows two different set of

routes. The left diagram on Figure 3 shows that some of the intermediate nodes are not included in any route. In contrast, the right diagram in Figure 3 shows all nodes are included in at least one route.

In general, for a network topology where there are 4 sinks located at the edges of the network as shown in Figure 3, we can divide the whole area into four quadrants. The key challenge is to determine the number of packets to send to each sink and the paths of those data packets should cover most of the nodes in the rectangle region. Figure 4 shows a general situation of the four quadrants in a 2 dimensional space. Assume the source node locates at the origin (0,0), the destination sink locates at (x,y). Let the transmission range be *t*, and $x=m't$, $y=mt$. First, we have the following lemma,

**Lemma 1** The largest area of $A_i$, depending on *m'*, occurs at

$$r_{m'} = m't \text{ or } r_{m'+1} = (m'+1)t \tag{3}$$

where $A_i=I(C(Rj))-I(C(Rj-1))$. $C(Ri)$ is the circle with radius $R_i = it$ and $I(C)$ is defined as the intersected region of $C$ with the rectangle. The value corresponding to $Max(A_i)$ is (whichever is larger):

$$\begin{aligned} & A^* = \pi t^2(2m-1)/4 \\ or \quad & A^* = [r_{m'+1}{}^2 \sin^{-1}(x/r_{m'+1}) - r_{m'}{}^2 \sin^{-1}(x/r_{m'}) \\ & + x\sqrt{r_{m'+1}{}^2 - x^2} - x\sqrt{r_{m'}{}^2 - x^2}\,]/2 \end{aligned} \tag{4}$$

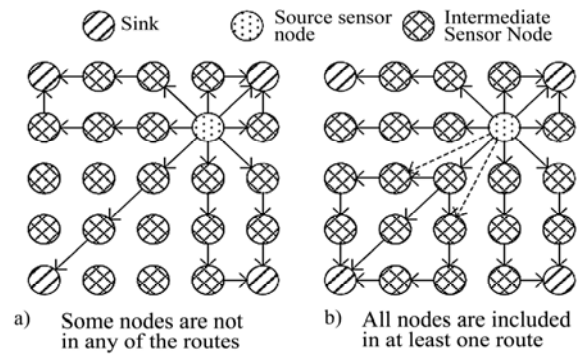



**Figure 3. Two different set of routes from the source node to the sinks.**

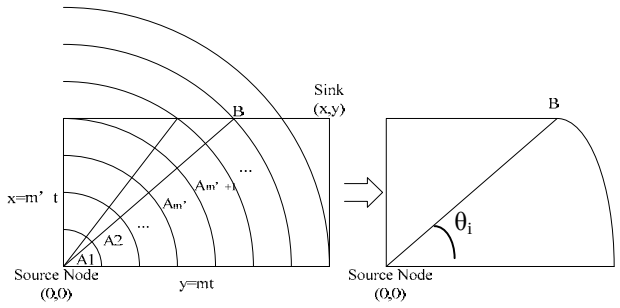



**Figure 4. Intersection between transmission range spheres and the rectangle between the source (u) and the sink (s).**

***Proof:*** Denote area by | |. If $Ri \le mt$, the area of $A_i$ is given by

$$\begin{aligned} A_i &= (C|R_i| - C|R_{i-1}|)/4 \\ &= \pi t^2(2i-1)/4 \end{aligned} \tag{5}$$

If $Ri > mt$, then the area is computed by dividing it to an arc and a right triangle, and is given by

$$A_i = (|Arc(R_i)| + |T(R_i)|) - (|Arc(R_{i-1})| + |T(R_{i-1})|) \tag{6}$$

where T(Ri) is the right triangle with hypotenuse equal to *Ri* and Arc(*Ri*) is I(C(*Ri*))- T(*Ri*). It is easy to find $|Arc(R_i)|$ and $|T(R_i)|$, and they are given by

$$\begin{aligned} |Arc(R_i)| &= r_{m'}{}^2 \sin^{-1}(x/r_i) \\ |T(R_i)| &= x\sqrt{r_i^2 - x^2}\,]/2 \end{aligned} \tag{7}$$

Substituting (5) in (4) gives

$$\begin{aligned} A_i &= [r_{i+1}{}^2 \sin^{-1}(x/r_{i+1}) - r_i^2 \sin^{-1}(x/r_i) \\ &+ x\sqrt{r_{i+1}{}^2 - x^2} - x\sqrt{r_i^2 - x^2}\,]/2 \end{aligned} \tag{8}$$

By taking derivatives of (5) and (8), it turns out that (5) is a increasing function of *Ri*, while (8) is a decreasing function of *Ri*. Thus, the maximum occurs either when $r_i = r_{m'} = m't$ (max. of (5)) or when $r_i = r_{m'} = m't$ (max. of (8)).

Next, we present our general angle dividing algorithm. The proposed angle dividing algorithm guarantees that in each routing area that divided by angle $\alpha_1, \alpha_2, ..., \alpha_l$, the largest number of nodes that need to receive distinguishing data packets are close to and bounded by

$$a_{upp} = S_{upp} D \tag{9}$$

where $a_{upp}$ denotes upper bound of the number of nodes, $S_{upp}$ denotes the upper bound of area of the region, and *D* is the nodes density of the WSN. We use *A* to denote region, *S* to denote area and assume the transmission range is *t*. The algorithm is presented in Algorithm 1. We explain Algorithm 1 by a simple example in Figures 5,6. The main idea is that the angles should be chosen such that the shadowed regions share the same area $S_{upp}$. It should be noted that the value of $S_{upp}$ will affect the delivery ratio as well as the convergence speed of our angle routing algorithm. A larger value of $S_{upp}$ allows larger angles $\alpha_1, \alpha_2, ..., \alpha_l$ to exist. A larger angle may increase the delivery ratio in the sense that if a data packet for a node in the shaded area is lost, other packets that share the same path before reaching the shaded area can serve as the informers. For example, in Figure 6, node 4 and node 5 will share the same intermediate node, node 0 and node 1. If a packet being sent to node 5 is lost at node 0, node 0 and node 1 can still learn

1. Suppose y is always the longer side of the rectangle. Let $m$=int($y/r$), $m'$=int($x/r$) and $r_i$ =ir, $i=1,2,\ldots,m$.

2. Compute $S_{A_m} = (S_{ARC_m}+S_{TRI_m})- (S_{ARC_{m-1}}+S_{TRI_{m-1}})$. In general, we have $S_{A_i} = (S_{ARC_i}+ S_{TRI_i}) -(S_{ARC_{i-1}}+ S_{TRI_{i-1}})$.

Where $S_{ARC_i} = \theta_i \ r_i^2/2$. $S_{TRI_i} = x\ \sqrt{r_i^2 - x^2}/2$, $\theta_i = \sin^{-1}(\frac{x}{r_i})$

3. $k$ = max $i$ such that $S_{A_k} > S_{upp}$.

4. $\alpha_0 = 2\, S_{upp}/(r_k^2 - r_{k-1}^2)$.

5. While $S_{A_k} > S_{upp}$ do {

$S_{A_k} = S_{A_k} - S_{upp} = S_{A_k} - \alpha_j (r_k^2 - r_{k-1}^2)/2$

$S_{A_{k-1}} = S_{A_{k-1}} - \alpha_j (r_{k-1}^2 - r_{k-2}^2)/2$

.....

$S_{A_1} = S_{A_1} - \alpha_j r_i^2/2$

$\alpha_j = \alpha_0$, $j$ ++.

}

7. While $i > m'$ do {

$i$--;

If $S_{A_i} < S_{upp}$ .continue;

Else {

$\alpha_j = \theta_i - (\alpha_0 + \alpha_1 + \ldots \alpha_{j-1})$

$S_{A_i} = S_{A_i} - S_{upp} = S_{A_i} - \alpha_j (r_i^2 - r_{i-1}^2)/2$

$S_{A_{i-1}} = S_{A_{i-1}} - \alpha_j (r_{i-1}^2 - r_{i-2}^2)/2$

.....

$S_{A_1} = S_{A_1} - \alpha_j r_1^2/2$

$j$ ++;

}

}

**Algorithm 1. DQRP path-finding protocol.**

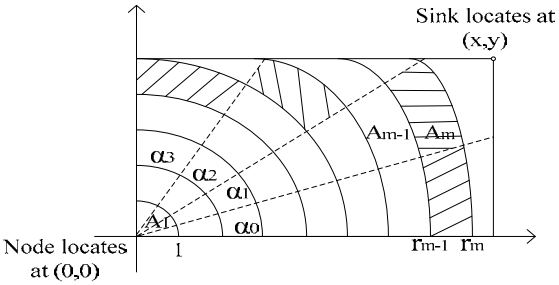



**Figure 5. An example of the angle dividing algorithm.**

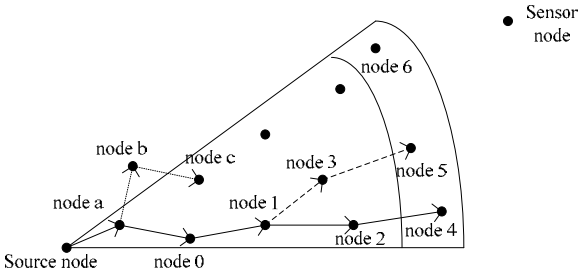



**Figure 6. An angle routing example**

the occurrence of the event by delivering the packet sent to node 4. In addition, as greedy routing is used with a larger angle, it is more likely that a node can find its subsequent node in that angle's region. This is the case of node c in Figure 6, which is out of node a transmission range, thus node b will help deliver the packet.

However, if the angle is too large, it will degrade the performance of the algorithm. As the routing information is given by angle, the node who has a packet to transmit does not know exactly which next node to pass the packet. It will choose any node that is valid to pass the packet.

In Figure 6, node 1 may pass the packet to either node 2 or node 3. Thus, node 4 may receive the packet twice while node 5 may miss the packet. In this case, we say that node 5 is node covered.

**Finding the appropriate value of *Supp***

Note that $a_{upp}$ determines the maximum number of nodes in a single hop region of each route. For example, if we set Supp to A* = **max**(SAm` , SAm`+1), then according to our old proposed algorithm the maximum possible value is **max(**$a_{upp}$**) = D·A*** . To analyze in more detail, we need to find the approximate area which is covered by each route, $\boldsymbol{A_R}$.

Firstly, it can be easily seen that the number of hops in each route to the sink is $H = \sqrt{m'^2 + m^2}$. Secondly, the area in each hop is bounded by Supp. This gives the following upper bound

$$A_R < H S_{upp} \tag{13}$$

The number of nodes in each route is then given by $\mathbf{N_R} = \mathbf{A_R \cdot D}$. If we apply the above upper bound, then there are approximately

$$\begin{aligned} N_R &= \sqrt{m'^2 + m^2}\, S_{upp} D \\ &= H S_{upp} D \end{aligned} \tag{14}$$

nodes in each route. Recall the properties of a "*good*" set of routes, i.e.

P1. Each node should be in one route from node u to sink s.

P2. Number of mutual nodes between routes should be minimum.

We find out that the minimum number for $\max(a_{upp})$ should be 1. By setting Supp=1/D, we get the minimum of $\max(a_{upp}) = (1/D)D = 1$, which satisfies P1. Hence, we have

$$A^* \geq S_{upp} \geq 1/D \tag{15}$$

Apparently, *1/D* is the best value for Supp if 100% of nodes are to be updated. In other words, increasing $S_{upp}$ may cause dissatisfying P1. However, this minimum may possibly cause some of the hops with smaller areas to be empty which is against P2. As a solution, we approximate each route's area as a triangle as can be seen in Figure 7. Note that corner triangles should also be con-

tinued to the sink, so we assume that each of these triangles has a height equal to $S_{upp}/t$, and a base equal to $Ht$. Thus we have

$$A_R \cong HtS_{upp}/2t = HS_{upp}/2 \quad (16)$$

To satisfy P1 and P2, it is required to set $N_R = H$. Therefore we have $N_R = A_R D = HS_{upp}D/2$, which implies

$$S_{upp} = 2/D \quad (17)$$

It should be also noted that for larger values of $S_{upp}$, P1 might not be satisfied due to the fact that the number of nodes may be increased to more than one and hence, be omitted from the routes. However, for larger values of $S_{upp}$, it is more unlikely to have same node receives the packet twice (i.e. satisfying P2).

Another important point is that if we assume we have only local information, then density D may not known at each node. We can either assume that we have D (non local info), or estimate it using

$$D^u_{est} = N^u_{neib} / (\pi t^2/4) \quad (18)$$

where $N^u_{neib}$ denotes the number of neighbors of node *u*.

## 6. Delay Analysis

### 6.1. Propagation Delay

The propagation delay, the approximate delay is proportional to $H = \sqrt{m'^2 + m^2}$. Therefore, transmission delay is proportional to distance between source and the sink, dt = c H, where c is the average propagation delay for one hop transmission.

### 6.2. Priority Queue delay

When there is buffer management in nodes, the packet sees an average delay, **ds**, at each hop. **ds** depends on

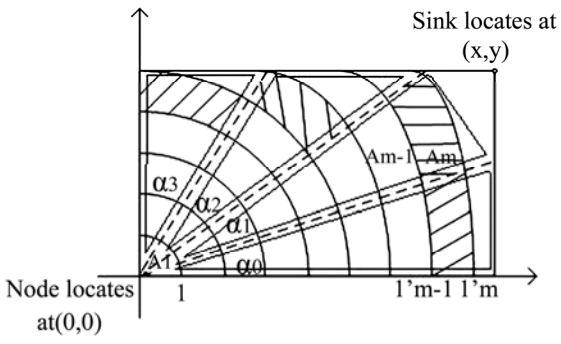



**Figure 7. Approximating the routes with triangles.**

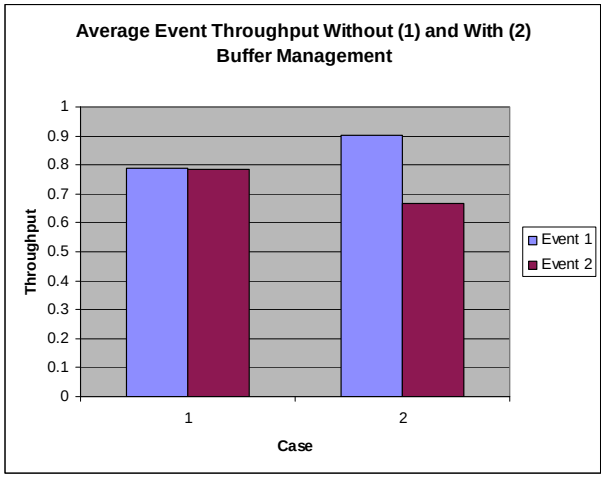



**Figure 8. Normalized throughput of 2 events with simple queue management (case 1) and priority queue (case 2).**

congestion level and also the priority of the packet. In [12], an analysis for the average delay in a non- preemptive Head of Line (HOL) priority queue is given. A non-preemptive HOL queue has the property that the packet in the head of line, which is being processed by the server, should be completely served before other packets can be processed. Using the results (Equation (9) in [12]) and simplifying it for a HOL M/M/1 priority queue, we get the following formula. [1]

$$ds_i = \mu_i + \frac{\sum_{m=1}^{i}\rho_m}{1-\sum_{m=1}^{M}\rho_m} + \frac{\sum_{m=1}^{i-1}\sum_{n=1}^{i-1}\rho_m\rho_n}{2(1-\sum_{m=1}^{i-1}\rho_m)(1-\sum_{m=1}^{i}\rho_m)} \quad (19)$$

where $\mu_i$ is the average service time for ***i***'th priority class; **M** is the total number of priority classes; and $\rho_i$ is the offered load of packets in priority class ***i***. Denoting the packet arrival rate of priority class ***i*** by $\lambda_i$, $\rho_i$ is defined as $\rho_i = \lambda_i\mu_i$. In general, a smaller index for a priority class implies a higher priority, and hence less delay for that class. In fact, from the Equation (10), it can be seen that the first class packets observe the least delay ( i.e. $ds_1 = \mu_1$ ). More importantly, this delay is less than the delay in a queue without prioritizing ( $ds' = \frac{1}{1-\rho}\mu$ ). The fact that $ds_1 < ds'$, shows that the more important packets leave the queue quickly. On the other extreme hand, the delay for last priority class $ds_M$ is obviously larger than $ds'$, and hence, the packets from less important class observe more delay compared to non- prioritized mode.

[1]The second and third terms in the right hand side of the Equation (1) should also have the same unit as $\mu_i$ by multiplying them into 1 time unit.

$ds_i$ is the queuing delay for one hop only. Similar to what we had for the propagation delay, the total queuing delay will be $dq_i = H\, ds_i$.

Figure 8 shows the effect of implementing a simple priority queue scheme on the throughput from two different priority classes. The setting is similar to Figure 2 when both events are active. As can be seen, after applying the buffer management, the class 1 packet will have more throughput compared to normal queue management. For example, if $\alpha_i = 0.9$, then the first event can be detected with the priority queue management in contrast to normal queue management.

### 6.3. Total Delay

The total delay is given by $d_i = d_t + dq_i = H\,(c + ds_i)$. Therefore, for the packets in a certain priority class, the nearest sink to the node observes the least delay. Hence, after disseminating of the information about the existence of one event by the proposed angle routing, the nearest sink can be used in order to achieve the least possible delay. The data dissemination phase guarantees that the maximum number of events is detected, because of the global event ordering knowledge in the network.

## 7. Analysis of Our Protocol

At first glance, our DQRP protocol seems to reduce the throughput or the number of events detected because the average number of hops per event packet increases. We should prove using linear programming to show that DQRP can actually increase our performance metric which is the number of targets detected. Shortest-path greedy geographic routing is not optimal in multi-sink sensor networks.

We consider a $n \times n$ grid topology of sensor nodes with 4 sinks located at the corners of the grid. The sensor network can be modeled using the representation of a graph $G=(V,E)$ where $V$ is the set of vertices and $E$ is the set of edges. The sink and all sensor nodes are in the set of vertices. Each vertex is associated with a location information given by $(x_i, y_i)$. Each sensor node has a maximum transmission range of $t$ There is an edge $(u, v)$ in $E$ if the nodes are within transmission range of each other. Formally, this is stated as

Edge $u,v \in E$ *iff* $\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \le t$

Data is sent from a sensor node through intermediate sensor nodes to the sink if the sink is not within direct transmission range of the sensor. We let $f(u,v)$ be the total amount of data transmitted from sensor node $u$ to sensor node $v$. These data includes data from other sensor nodes and data originating from the node itself. We let $G$ be a weighted graph with a weight function $w$. The weight $f(u,v)D$, $w(u,v)$ of the edge $u,v \in ED$ is the cost of transmission from node $u$ to node $v$. The transmission cost is dependant on the distance between the nodes as well as the propagation model used. In general, $w(u,v) \infty d^k$ where $d$ is the distance between 2 nodes and $k$ is the path loss exponent. We let $k$ be 2 in our scenario. As we consider only a static grid topology, $w(u,v)$ is a fixed value. We consider 3 types of routing:

1) Nearest-sink Greedy Geographic Routing: This means that the node will choose the nearest sink to send the data packets to and the next hop it chooses will maximize the distance gained towards the sink.

2) Nearest-sink Non-Greedy Geographic Routing: This means that the node will choose the nearest sink to send the data packets to and it can choose any neighbor which is nearer to the sink than it is to forward the data packets.

3) Multi-sink Non-Greedy Geographic Routing: This means that the node can choose any sink to send the data packets to and it can choose any neighbor which is nearer to the target sink than it is to forward the data packets. We minimize the maximum energy consumed by any sensor node by linear programming:

### 7.1. Nearest-Sink Greedy Geographic Routing

Minimize $p$ subject to the following constraints:

$$f(u,v) = 0 \quad \text{for each} \quad (u,v) \notin E \tag{20}$$

$$f(u,v) \ge 0\text{D} \quad \text{for each} \quad (u,v) \in E \quad \text{and } v \text{ maximizes distance gained towards the nearest sink of } u \tag{21}$$

$$\sum_{v \in V} f(u,v) \le c \quad \text{for each} \quad u \in V - \{\text{sink}\} \tag{22}$$

$$\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = L_u \quad \text{for each} \quad u \in V - \{\text{sink}\} \tag{23}$$

$$\sum_{v \in V} f(u,v) = 0 \quad \text{for u} \in \{\text{sink}\} \tag{24}$$

$$\sum_{u \in \text{sink}} \sum_{v \in V} f(v,u) = \sum_{u \in V - \{\text{sink}\}} L_u \; \text{u} \in \{\text{sink}\} \tag{25}$$

$$\sum_{v \in V} f(u,v) w(u,v) \le p \;\; u \in V - \{\text{sink}\} \tag{26}$$

The variable $c$ is the maximum amount of data transmitted by any sensor node among all the sensor nodes in the network. Constraint (20) means that if two sensor nodes are not within the transmission range, the flow between each other is 0. If two sensor nodes are within transmission range, the flow between each other must be non negative as stated in Constraint (21). Constraint (21) will vary based on the type of routing algorithm used. Constraint (22) states that the total amount of transmission data by any sensor node cannot exceed $c$. We set $c$ to be 10 packets of data. Constraint (23) states that all received data are to be forwarded and every sensor node has $L_u$ units of data to be sent to the sink. This value

depends on the number of event packets assigned to that node. Constraints (24) and (25) states that the sink is not sending any data and should receive all the data from the sensor nodes. Constraint (26) states that if the linear programming problem can be solved, the solver should produce a solution that minimizes the energy consumption of the sensor nodes.

### 7.2. Nearest-Sink Non-Greedy Geographic Routing

The linear program is similar to the previous case but (28) is modified.

Minimize $p$ subject to the following constraints:

$$f(u,v) = 0 \quad \text{for each} \quad (u,v) \notin E \tag{27}$$

$f(u,v) \geq 0$ for each $(u,v) \in E$ and $v$ is nearer to the nearest sink than $u$ is (28)

$$\sum_{v \in V} f(u,v) \leq c \quad \text{for each} \quad u \in V \text{ - } \{\text{sink}\} \tag{29}$$

$$\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = L_u \quad \text{for each} \quad u \in V \text{ - } \{\text{sink}\} \tag{30}$$

$$\sum_{v \in V} f(u,v) = 0 \quad \text{for} \quad u \in \{\text{sink}\} \tag{31}$$

$$\sum_{u \in \text{sink}} \sum_{v \in V} f(v,u) = \sum_{u \in V - \{\text{sink}\}} L_u \quad \text{for} \quad u \in \{\text{sink}\} \tag{32}$$

$$\sum_{v \in V} f(u,v) w(u,v) \leq pd \text{ for each } u \in V \text{ - } \{\text{sink}\} \tag{33}$$

### 7.3. Multi-sink Non-Greedy Geographic Routing

The linear program is similar to the previous case but (35) is modified.

Minimize $p$ subject to the following constraints:

$$f(u,v) = 0 \quad \text{for each} \quad (u,v) \notin E \tag{34}$$

$$f(u,v) \geq 0 \quad \text{for each} \quad (u,v) \in E \tag{35}$$

$$\sum_{v \in V} f(u,v) \leq c \quad \text{for each} \quad u \in V \text{ - } \{\text{sink}\} \tag{36}$$

$$\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = L_u \quad u \in V \text{ -} \{\text{sink}\} \tag{37}$$

$$\sum_{v \in V} f(u,v) = 0 \quad \text{for u} \in \{\text{sink}\} \tag{38}$$

$$\sum_{u \in \text{sink}} \sum_{v \in V} f(v,u) = \sum_{u \in V - \{\text{sink}\}} L_u \text{ , u } \in \{\text{sink}\} \tag{39}$$

$$\sum_{v \in V} f(u,v) w(u,v) \leq p \text{ for each } u \in V \text{ - } \{\text{sink}\} \tag{40}$$

We vary the number of packets generated by each event and determine the maximum number of events which can satisfy the QoS. Each event is randomly assigned to sensor nodes. The transmission range is set to $\sqrt{2}$, therefore each node has 8 neighbors. The maximum number of events is reached when the solver cannot find a solution to the linear program. We use the solver in MATLAB for this analysis. Figure 9 shows the maximum number of events satisfying the QoS requirements for a 5 by 5 grid topology and Figure 10 shows the results for a 6 by 6 grid topology. These results show that multipath non-greedy geographic routing can indeed improve network performance.

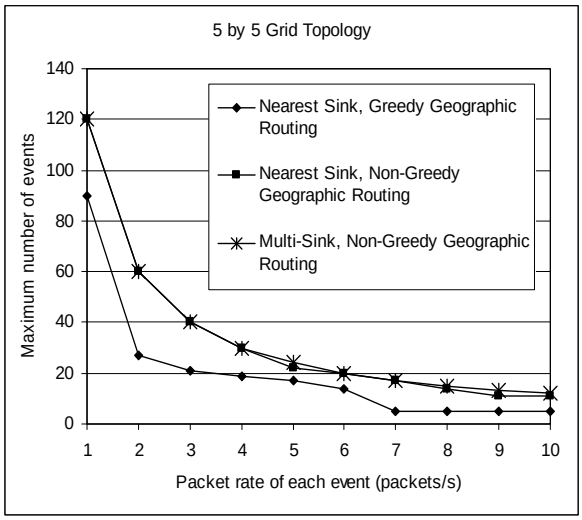



**Figure 9. Maximum number of events satisfying QoS requirements in a 5 by 5 grid topology.**

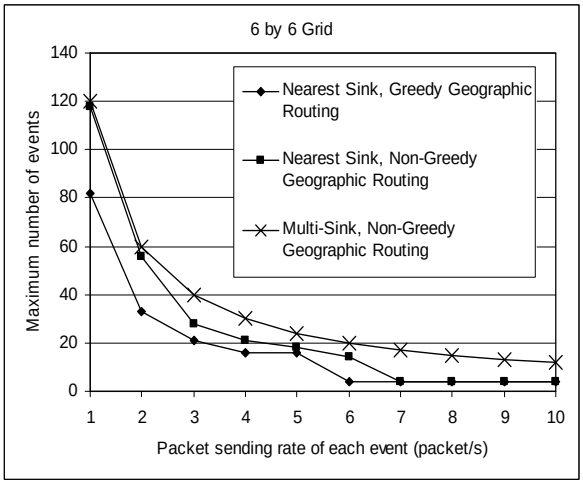



**Figure 10. Maximum number of events satisfying QoS requirements in a 6 by 6 grid topology.**

## 8. Implementation Results

We test the effectiveness of our priority queue buffer by implementing the scheme on the MICAz motes. In the first scenario, we use 3 motes in which there is 1 sink and 2 traffic sources as shown in Figure 11. In the second

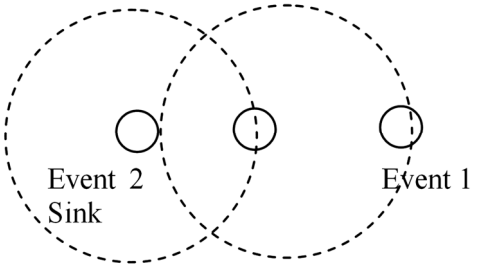



**Figure 11. Network topology in scenario 1.**

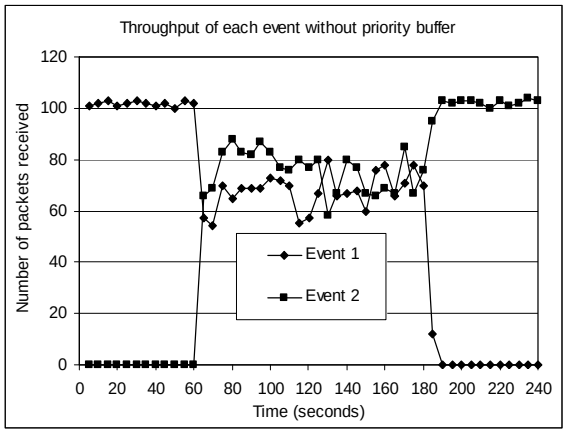



**Figure 12. Throughput of events without priority buffer.**

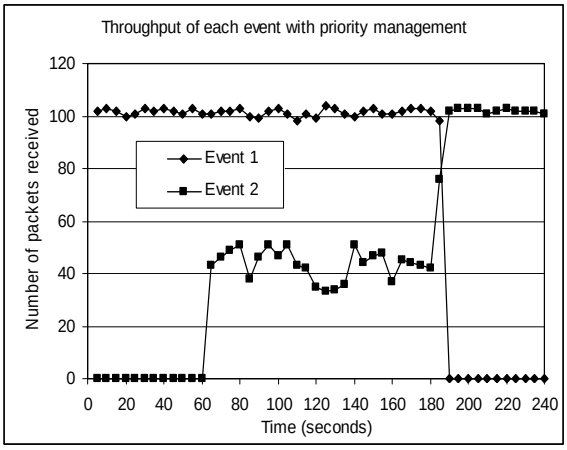



**Figure 13. Throughput of events with priority buffer.**

scenario, we use a total of 18 motes in which there is 1 sink and 5 traffic sources. The nodes are randomly deployed in a lab. Each mote is between 1 to 5 hops away from the sink. Each event sends packets at a rate of 20 packets per second for the first scenario and 10 packets per second. The sampling interval is set to 5 seconds.

Figure 12 shows the number of packets received by each event when there is no priority buffer management scheme and Figure 13 shows the number of packets received by each event when there is priority buffer management scheme. Figure 14 shows the number of events detected based on our QoS requirements. It can be clearly seen that the priority management scheme improves QoS performance. In the random deployment scenario, each event is gradually introduced into the network until there are 5 events.

After that, events gradually leave the network. Figure 15 shows the number of events detected. The results show that our priority buffer management scheme performs better most of the time.

One of the main problems that we face when carrying out the experiments is that the experiment was done in a lab where shadowing and fading effects are more serious compared to an open field. This causes the link quality to vary rapidly. As a result, there are a lot of fluctuations in the number of packets received and routes between nodes change quite often. Our scheme could potentially improve the performance much higher if the experiment is carried out in an open environment (e.g. open field) where there are less obstacles and the link quality is more stable.

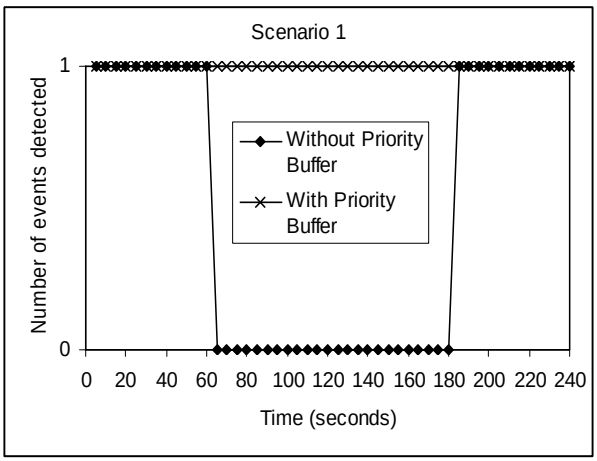



**Figure 14. Number of events satisfying our QoS requirements. The required data delivery ratio is 90%.**

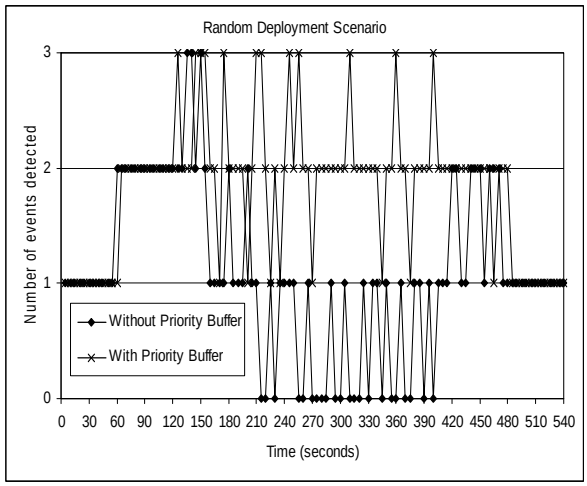



**Figure 15. Number of detected events in a random deployment scenario satisfying our QoS requirements. The required data delivery ratio is 70%.**

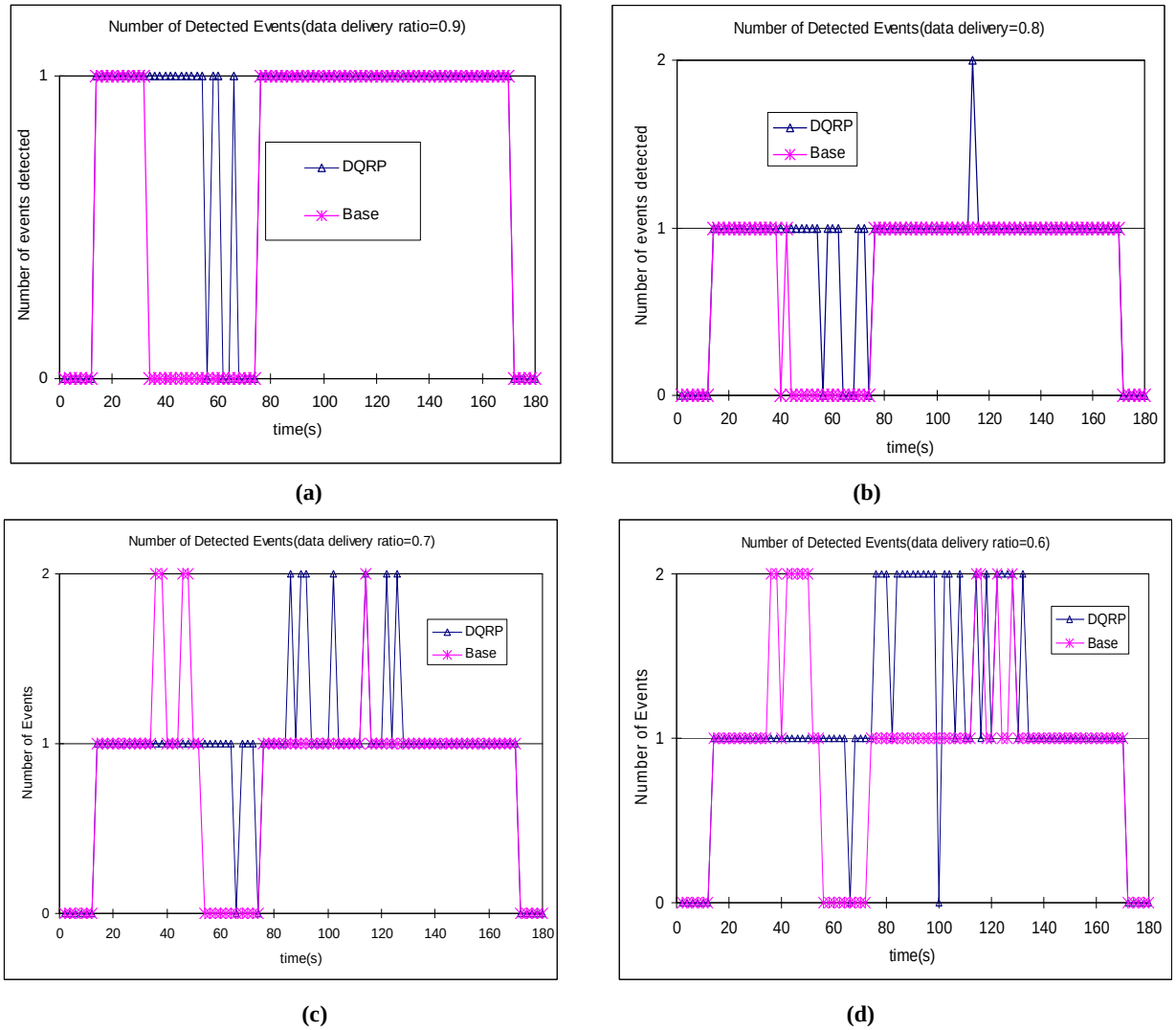



**Figure 16. Number of detected events, out of max. 4 events, in the basic setup (stars) and DQRP (triangles). Vertical axis shows the number of events detected, and horizontal axis shows the time. The required data delivery ratio for an event detection at sinks is a) 0.9, b) 0.8, c) 0.7, and d) 0.6.**

## 9. Simulation Results

We simulate a network consisting of 25 nodes with DQRP and using priority buffer management scheme based on event ordering. The simulation tool used is TOSSIM. There are 4 sinks located at each corner of the simulation area. We simulate two scenarios, one with 4 events and another with 8 events. Events are gradually inserted into the network. Events can also leave the network after some time. The sampling interval is 2 seconds. We compare the number of events detected with varying QoS requirements from 60% delivery ratio to 90% delivery ratio. The results for the scenario with 4 events are shown in Figure 16 and the results for the scenario with 8 events are shown in Figure 17. The results show that DQRP with priority buffer management scheme based on event ordering achieves better performance.

## 10. Conclusions and Future Work

In this paper, we have presented a QoS network architecture for target tracking WSNs consisting of a priority buffer management scheme based on event ordering and a routing algorithm to disseminate the event ordering.

We believe that better performance can be obtained from our system by using cross-layer design. For example, once we have the global event-ordering list, at the MAC layer, we can assign more time slots using a TDMA MAC protocol to events that have a higher priority. This can provide more assured QoS than the CSMA MAC protocol that is used in our simulations and experiments.

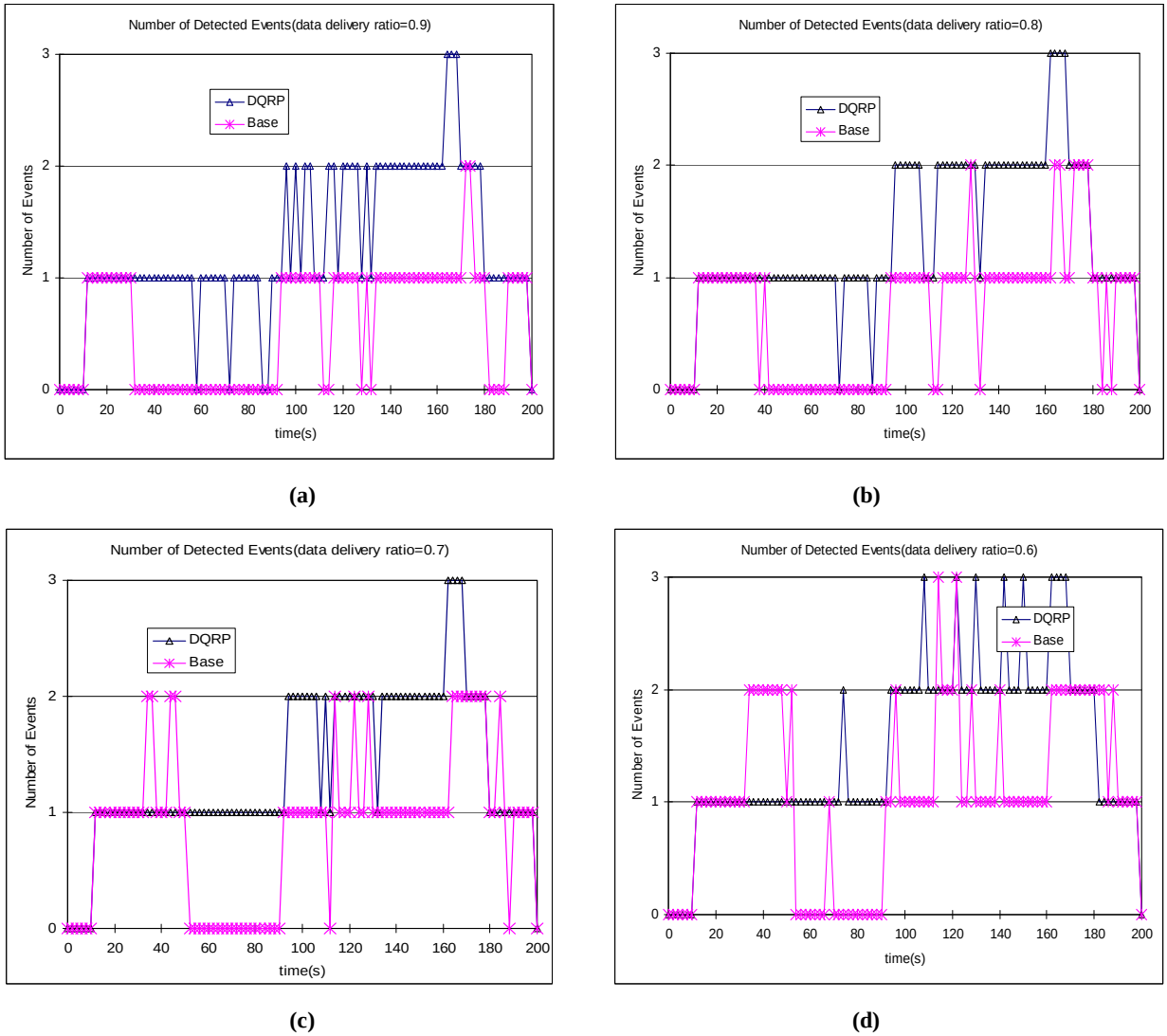



**Figure 17. Number of detected events, out of max. 8 events, in the same format as Figure 16.**

## 11. References


[1] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of sensor network communication models," ACM Mobile Computing & Communication Review, Vol. 6, No. 2, pp. 1−8, 2002.

[2] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," Proceedings of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA, June 2004.

[3] R. Iyer and L. Kleinrock, "QoS control for sensor networks," Proceedings of IEEE International Conference on Communications (ICC'03), Vol. 1, pp. 517−521, May 2003.

[4] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," ACM International Conference on Mobile Computing and Networking (MOBICOM), 2004.

[5] Q. X. Wang, W. P. Chen, R. Zheng, K. Lee, and L. Sha, "Acoustic target tracking using tiny wireless sensor devices," In International Workshop on Information Processing in Sensor Networks (IPSN), 2003.

[6] G. H. He and J. C. Hou, "Tracking targets with quality in wireless sensor networks," proceedings of the 13th IEEE International Conference on Network Protocols, 0-7695-2437−8/05, 2005.

[7] K. I. Park, "QoS in packet networks," Springer, pp. 159−179, 2005.

[8] A. Durresi, V. K. Paruchuri, S. S. Iyengar, and R. Kannan, "Broadcast protocol for sensor networks," Technical Report, LSU-CSC, October 2003.

[9] A. Durresi, V. K. Paruchuri, L. Barolli, and R. Jain, "Qos-energy aware broadcast for sensor networks", Proceeding of the 8th IEEE Int. Symp. on Parallel Architectures, Algorithm and Networks (ISPAN'05), 1087−4089/05, 2005.

[10] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in Ad-Hoc wireless networks," ACM Wireless Networks, November 2001.

[11] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," ACM/IEEE Int. Conf. on Mobile Computing and Networking (Mobicom), 2000.

[12] J. Walraevens, B. Steyaert, M. Moeneclaey, and H. Bruneel, "Delay analysis of a HOL priority queue," Telecommunication Systems, Vol. 30, No. 1–3, pp. 81–98, 2005.

[13] G. N. Shirazi, P. Wang, D. Xiangxu, Z. A. Eu, and C. K. Tham, "A QoS network architecture for multi-hop multi-sink target Ttracking WSNs," IEEE Inte Conference on Communication Systems (ICCS), pp. 17−21, 2008.

*Wireless Sensor Network*, **2009, 1, 383-396**
doi:10.4236/wsn.2009.15047 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Antenna and Base-Station Diversity for WSN Livestock Monitoring

**Konstantinos SASLOGLOU, Ian A. GLOVER, Hock Guan GOH, Kae Hsiang KWONG**
**Michael P. GILROY, Christos TACHTATZIS, Craig MICHIE, Ivan ANDONOVIC**
*Department of Electronic and Electrical Engineering*
*University of Strathclyde, Glasgow, United Kingdom*
*Email*: {*ksasloglou, ian.glover, i.andonovic*}*@eee.strath.ac.uk*
*Received June* 6, 2009; *revised August* 3, 2009; *accepted August* 6, 2009

## Abstract

Antenna and base-station diversity have been applied to a wireless sensor network for the monitoring of livestock. A field trial has been described and the advantage to be gained in a practical environment has been assessed.

**Keywords:** Antenna Diversity, Base Station Diversity, Animal Monitoring, Wireless Sensor Networks, Distribution, Rayleigh Distribution, Fading

## 1. Introduction

A wireless sensor network (WSN) is a collection of spatially dispersed sensors that communicate via a set of wireless transceivers [1]. Each transceiver forms one node in the resulting transceiver network. Information collected by the sensors may be transmitted to a central base-station, either directly or by relaying it via one or more intermediate nodes. The network topology and protocols may be fixed and predetermined, or adaptive and self-organizing. Recent advances in micro-electro-mechanical systems, transceiver miniaturisation and transducer technology have made WSNs flexible, scalable and commercially viable. They have found wide and diverse application in many areas including military [2], industrial [3], commercial [4] and domestic [5]. Of particular relevance to the work presented here are applications to environment and habitat monitoring, agriculture and animal husbandry [6–15].

Here we apply antenna and base-station diversity to star-configured WSNs for animal husbandry in the dairy and beef industries [16].

The monitoring and/or tracking of mammals usually involves a single antenna attached to a collar worn around the animal's neck. As the animals move, line-of-sight (LOS) paths between the sensor node and base-station might become obscured by other animals. This fading mechanism is likely to be especially important in the context of animals with a herding instinct on open grassland. Monitoring livestock in the dairy and beef industries corresponds precisely to this case.

The incorporation of antenna diversity at the sensor node combined with the use of two, widely separated, base-stations (yielding base-station diversity) dramatically increases the probability of LOS conditions. The principal objective of the work reported here is the experimental assessment of the practical diversity improvement that can be expected in this little considered, but commercially important, application.

## 2. System and Methodology

The transceiver used for the experiment was the MICAz [17] shown in Figure 1. It operates in the ISM band between 2.40 GHz and 2.48 GHz.

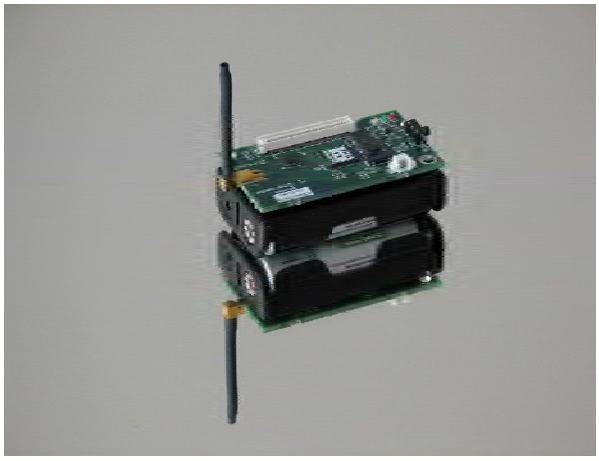


**Figure 1. MICAz module.**

The transceiver, with transmit power set to -10 dBm, was mounted on a PCB, Figure 2. An RF switch was used to connect the transceiver (both receiver and transmitter) to two antennas.

Each antenna is an inset-fed microstrip patch with a ceramic element attached to the top of the radiating surface. The radiation pattern of the antenna is shown in Figure 3 for three frequencies, in two orthogonal linear polarisations and in three orthogonal planes.

The antennas are alternately connected for 1 s to the transceiver using the RF switch. The switching cycle therefore has a period of 2 s and the sampling frequency for a particular antenna is 0.5 Hz. The mobile node assemblies were attached to animals using collars such that one antenna was located on the left-hand side of the animals' necks, and one on the right-hand side, Figure 4.

The trial area was rectangular in shape, approximately 20 m x 12 m, and enclosed by brick walls and a pitched metallic roof, Figure 5.

The base-stations are located at the mid point of the left and right hand side of the trial area (Figure 4) at a height of 4 m. They comprise of an identical transceiver to those used at the sensor nodes interfaced to an MIB600 programming board, Figure 6. The base-station

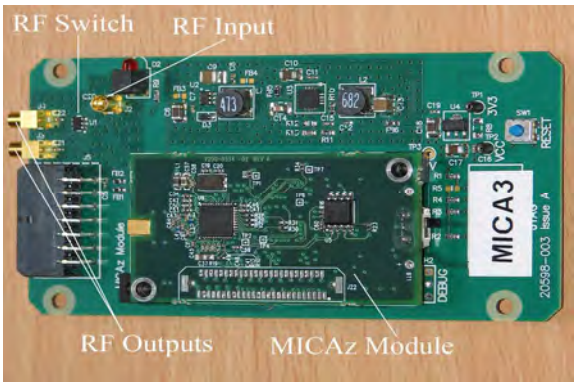



**Figure 2. Assembled PCB.**

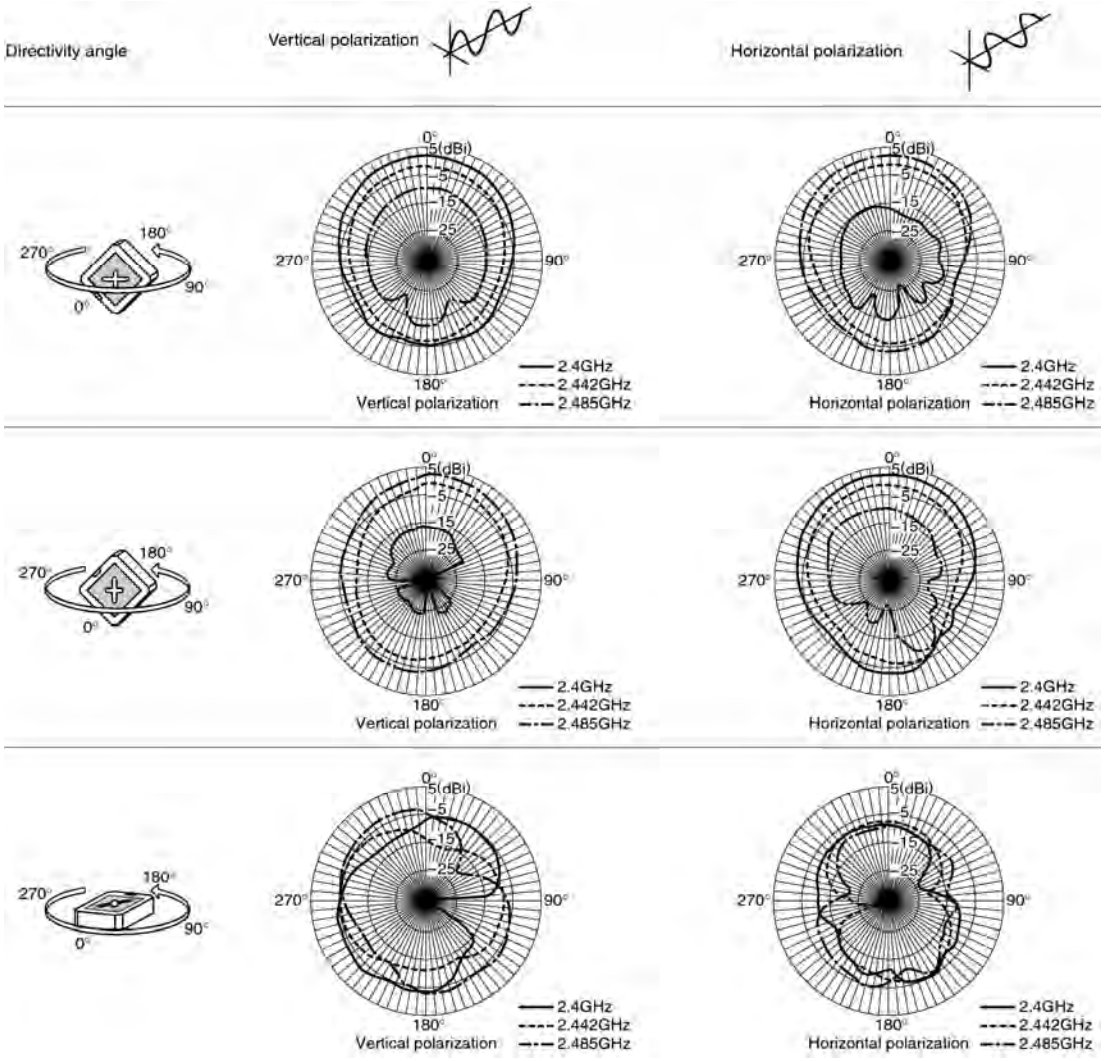



**Figure 3. Antenna radiation patterns.**

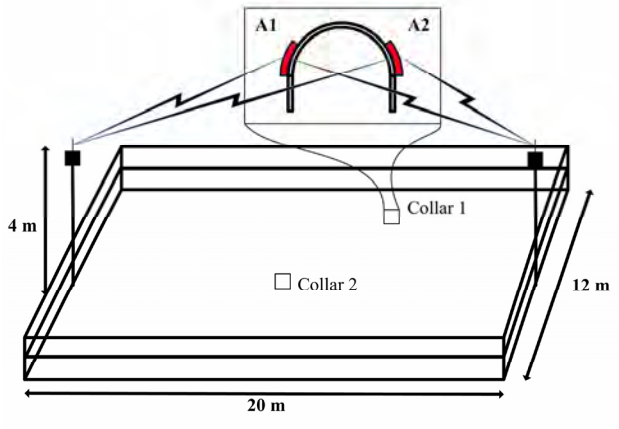



**Figure 4. Antenna configuration.**

**Figure 5. Trial area.**

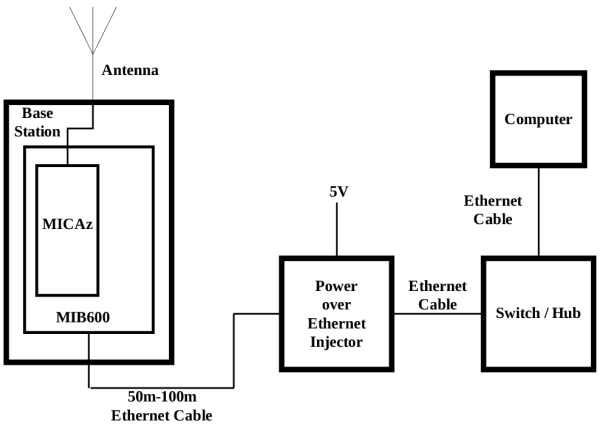



**Figure 6. Base-station.**

antennas were vertically polarised and approximately omnidirectional in the horizontal plane with a gain of 6 dBi [18].

Nine animals were released into the trial area, two carrying collar mounted sensor nodes. The received power was recorded for 75 minutes at both base-stations. The movement of animals was sufficiently slow such that each 1 s block of contiguous data received from a given antenna can be assumed to originate from a single location. The resulting data was smoothed by calculating the moving average of 15 samples representing a measurement integration time of 30s.

## 3. Results

The time-series of signal power received from base-station 1 (BS1) and base-station 2 (BS2) are shown in Figures 7(a), (b), (c) and (d) and Figures 8(a), (b), (c) and (d), respectively. (a) and (b) represent data originating from antenna 1 (A1) and antenna 2 (A2) mounted on collar 1 (C1). (c) and (d) represent data originating from A1 and A2 mounted on C2. The upper subplots in each subfigure show the raw 0.5 Hz data samples and the lower subplots show the 15-sample moving average. The horizontal line in the figures represents the mean power for each measurement set.

Power fluctuations of up to 20 dB occur in the raw time-series at both base-stations. The peak-to-peak variation of received power recorded for each antenna on each collar at each base-station over the total observation time is shown in Figure 9.

For diversity advantage to be realised the fluctuation of received power in the two channels must be decorrelated. The correlation coefficient $\rho_{X,Y}$ between two random variables $X$ and $Y$ with expected values $\mu_X$ and $\mu_Y$ and standard deviations $\sigma_X$ and $\sigma_Y$ is:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y} \qquad (1)$$

where $E$ denotes the statistical expectation and *cov* denotes the covariance [19]. The correlation coefficients between the signals received by antennas mounted on the same collar are presented in Table 1. The definition of correlation (Equation 1) excludes any constant (LOS) component which explains the low values.

The probability distribution of the received signal from a particular mobile antenna to a particular base-station might be expected to be close to Ricean as a result of multipath propagation with a strong LOS component. The Ricean distribution is given by:

$$p(r) = \begin{cases} \frac{r}{\sigma^2} e - \frac{(r^2 + A^2)}{2\sigma^2} I_o\left(\frac{A_r}{\sigma^2}\right) & for(A \geq 0, r \geq 0) \\ 0 & for(r < 0) \end{cases} \qquad (2)$$

**Table 1. Correlation coefficient for signals received by different antennas on a single collar.**

| *Base Station* | *Collar* | *Correlation Coefficient between A1 and A2* |
|---|---|---|
| BS 1 | Collar 1 | 0.2407 |
| | Collar 2 | 0.0414 |
| BS 2 | Collar 1 | –0.0733 |
| | Collar 2 | 0.0158 |

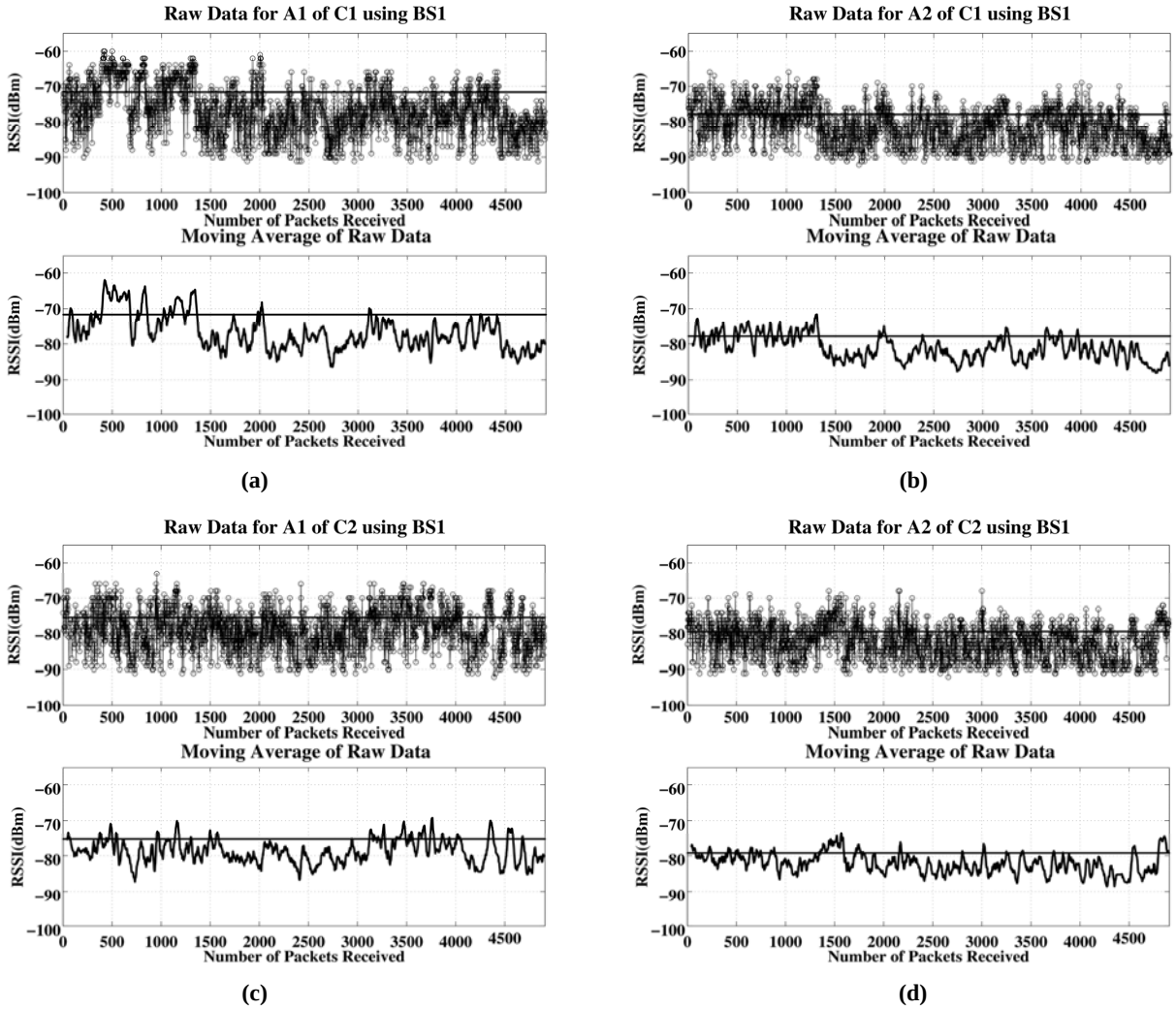



**Figure 7. Time-series recorded at base-station 1.**

where $A$ denotes the peak amplitude of the dominant component and $I_0$ is the modified zero order Bessel function of the first kind [20]. The $K$-factor of a Ricean distribution is the ratio between the (constant) component of signal power due to the LOS path and the (fluctuating) component of signal power due to all other paths, i.e.:

$$K = \frac{A^2}{2\sigma^2} \tag{3}$$

As the LOS component becomes smaller $K$-factor decreases and the Rician probability density function (pdf) becomes more skewed. As $K$-factor falls significantly below 1.0 the pdf approaches a Rayleigh distribution. As the LOS component becomes larger, $K$-factor increases and the distribution becomes less skewed. As $K$-factor tends to infinity the Ricean distribution tends to a normal distribution. Figure 10 shows the pdfs of the power received at BS1. (a) and (b) are the pdfs of data obtained from A1 and A2 on C1. (c) and (d) are the pdfs of data obtained from A1 and A2 on C2.

A normal distribution of power in dBm (i.e. a log-normal distribution of power in watts) appears to be the best fit to the data. If fading is due predominantly to multipath propagation this suggests the presence of a strong LOS component. An alternative interpretation would be that the log-normal fading reflects cascaded independent shadowing processes.

Figure 11 shows the pdf of the power received at BS2. (a) and (b) represent the data transmitted by C1 (for A1 and A2 respectively) and (c) and (d) represent the data transmitted from C2 (for A1 and A2 respectively). Superficially, this distribution appears to be closer to Rayleigh (in dBm) than normal. The mean signal level is significantly lower than that for BS1 (due to the larger distance), however, and is approaching the receiver sensitivity which is –94 dBm. Since no signal is recorded when the received power falls below -94 dBm the pdf is

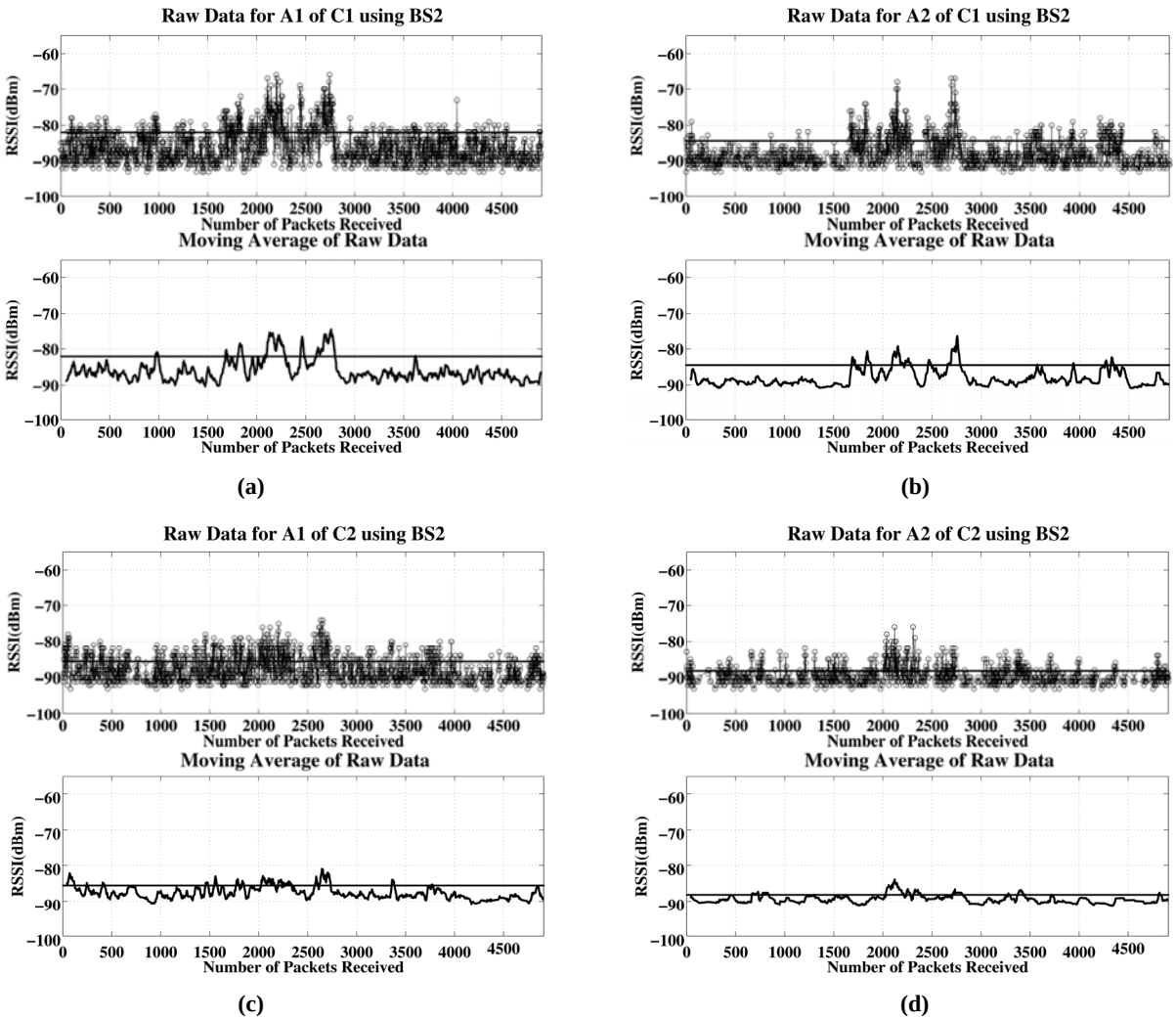



**Figure 8. Time-series recorded at base-station 2.**

effectively truncated at this level. It seems likely, therefore, that the pdf of the underlying signal is normal (in dBm) even though the pdf of the recorded (truncated) signal is skewed.

The corresponding cumulative distribution functions (cdfs) are presented in Figures 12 and 13. The best-fit normal curves along with 95% confidence intervals are superimposed.

The close fit of the normal distribution for the data logged at BS1 is apparent. The fit is less good for the data obtained from BS2. Figures 14 and 15 show similar plots for received voltage.

Figure 16 represents similar data to that presented in Figures 7 and 8 but of shorter time duration (approximately 40 minutes). The advantage of the use of base-station diversity is especially apparent in this data. Figures 16(a) and (b) represent the power received at BS1 and BS2, respectively, from the signal transmitted from A1 on C1. Figures 16(c) and (d) represent the power received at BS1 and BS2, respectively, from the signal transmitted from A2 on C1.

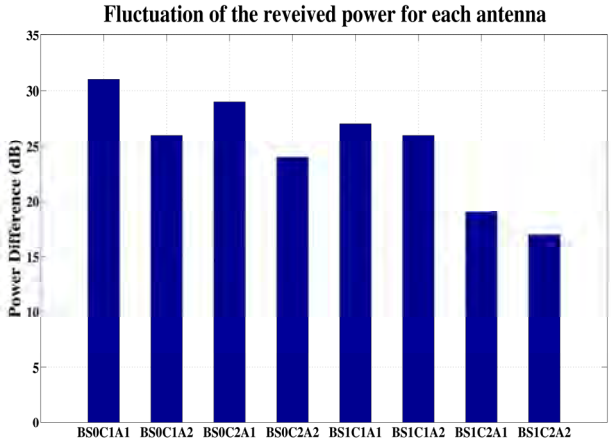



**Figure 9. Peak-to-peak variation of received power at each individual antenna. (From left to right, BS1C1A1, BS1C1A2, BS1C2A1, BS1C2A2, BS2C1A1, BS2C1A2, BS2C2A1, BS2C2A2.)**

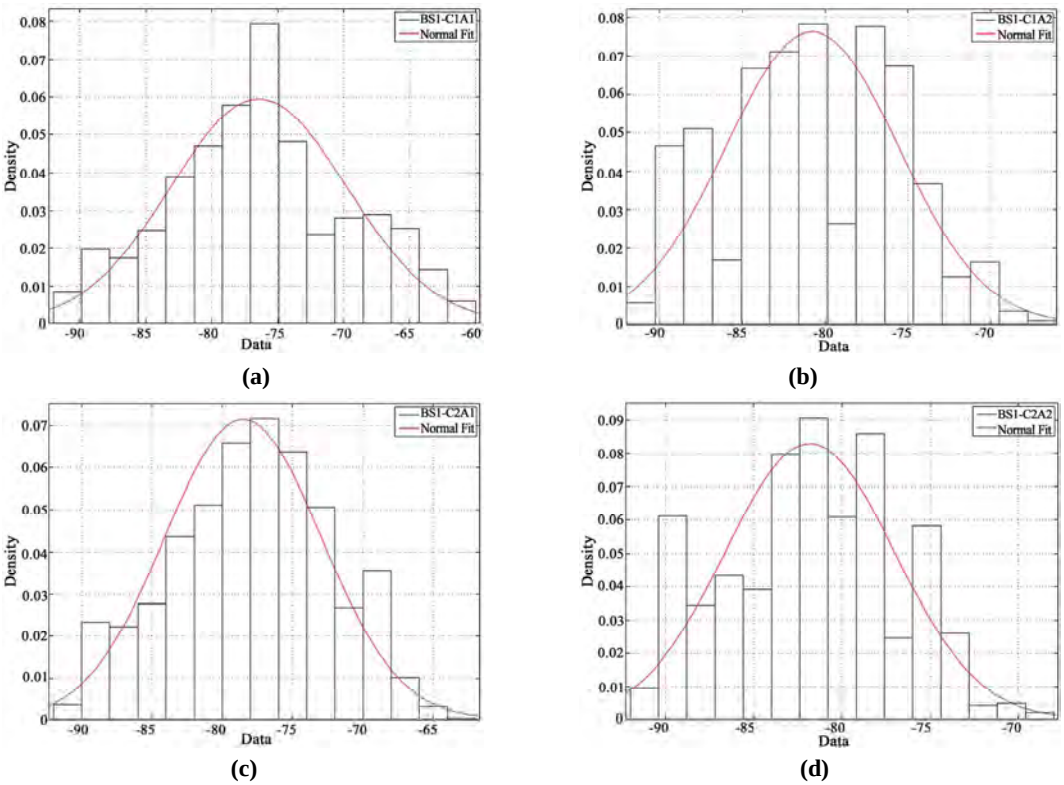



**Figure 10. Pdfs of power (dBm) at BS1 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best-fit normal distribution.)**

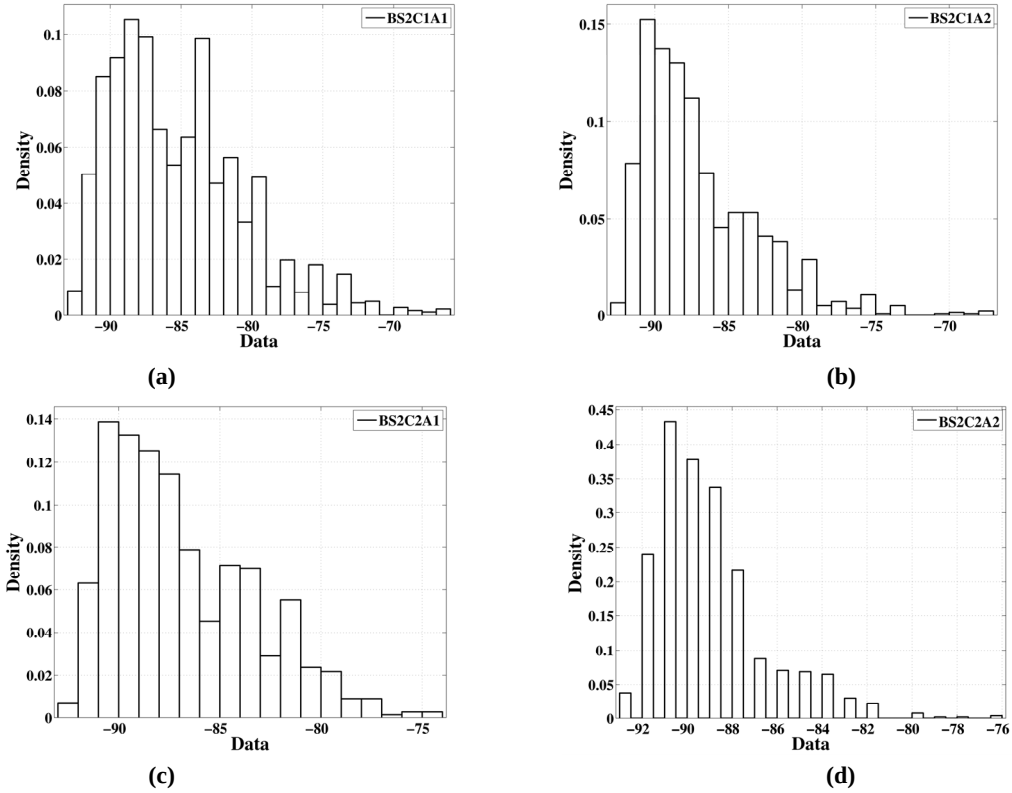



**Figure 11. Pdfs of power (dBm) at BS2 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best-fit normal distribution.)**

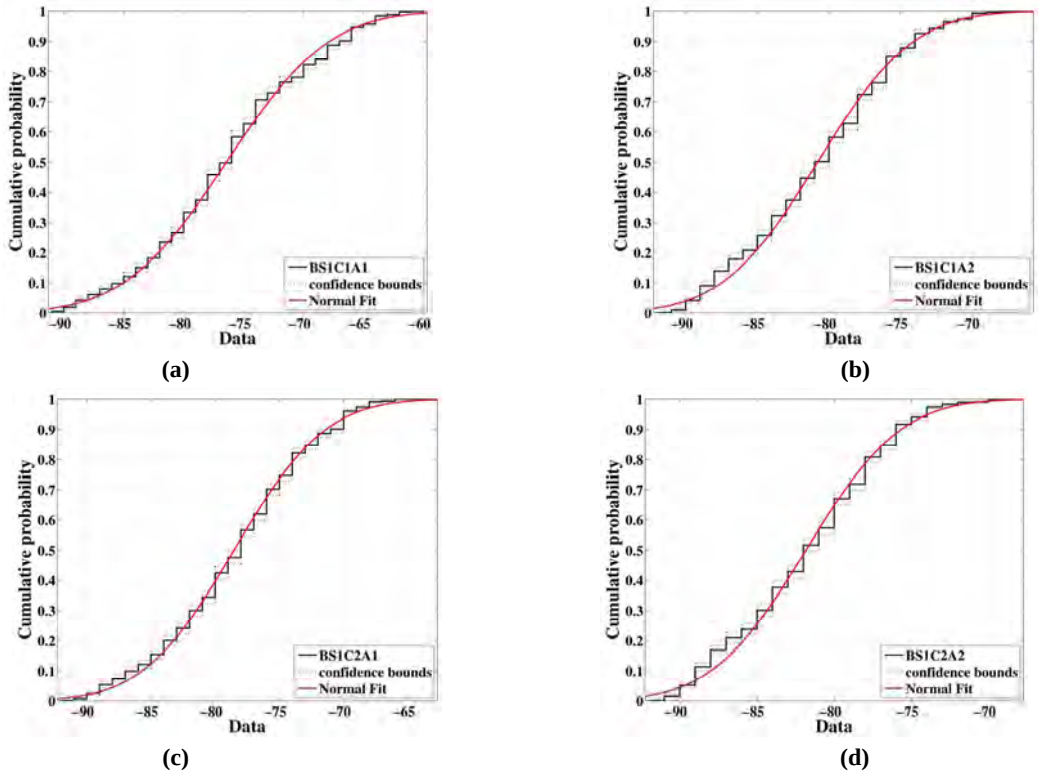



**Figure 12. Cdfs of power (dBm) at BS1 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best-fit normal distribution.).**

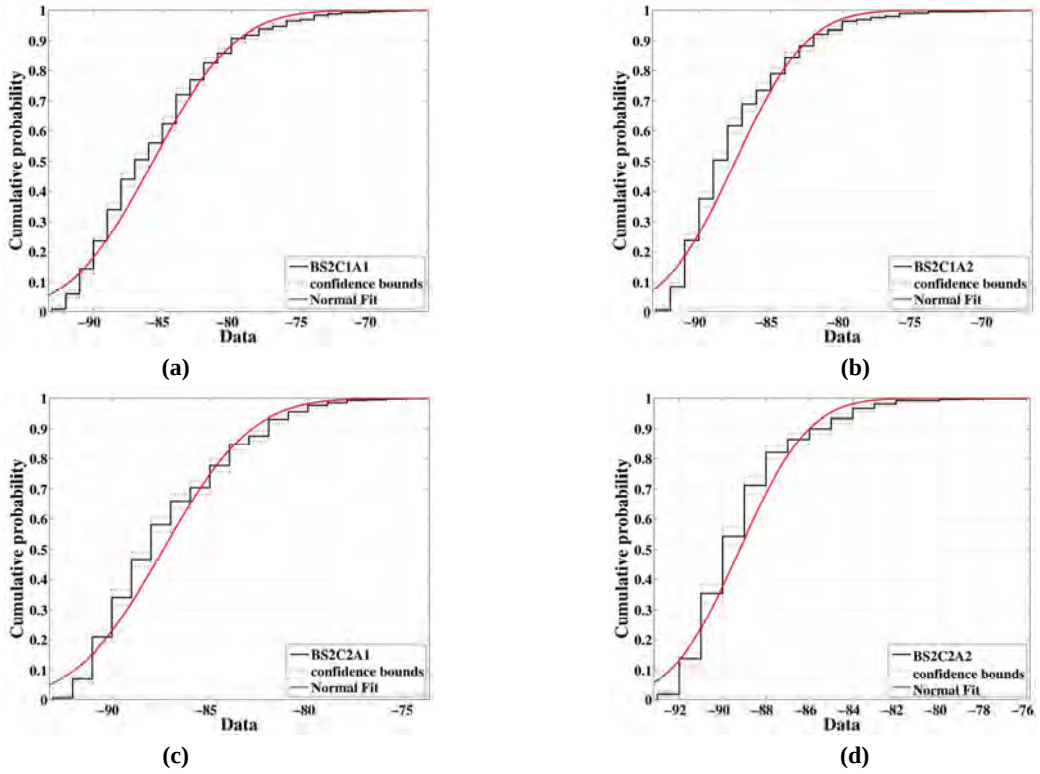



**Figure 13. Cdfs of power (dBm) at BS2 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best-fit normal distribution.)**

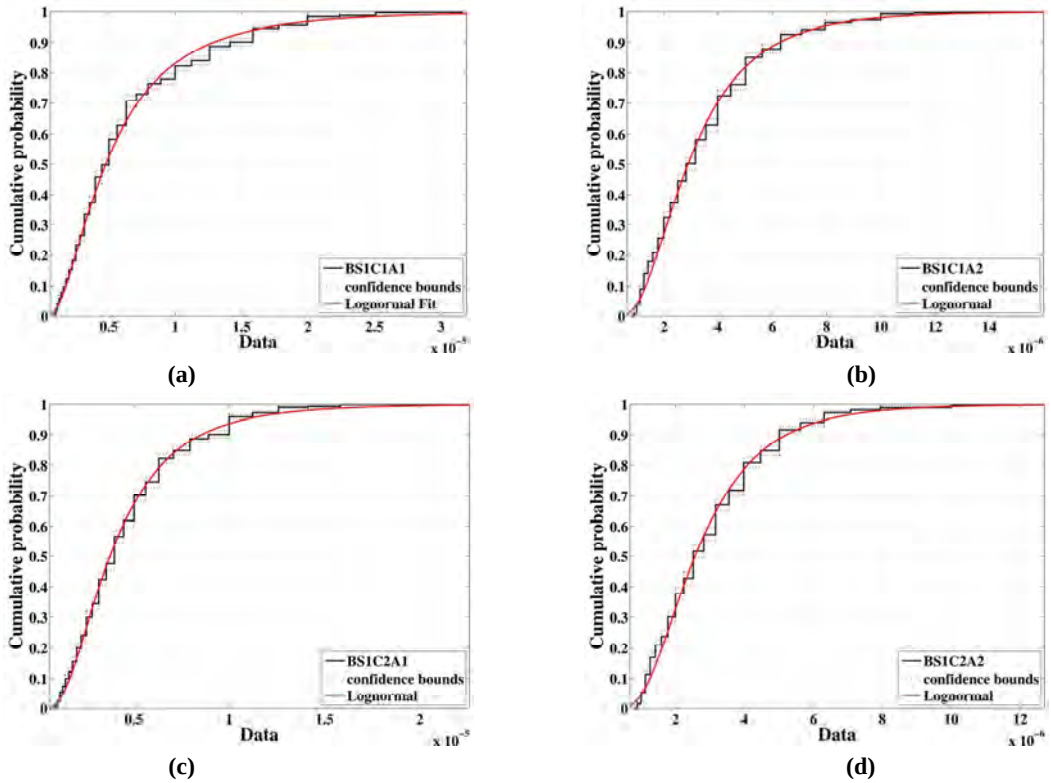



**Figure 14. CDFs of detected voltage at BS1 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best fit log-normal distribution.).**

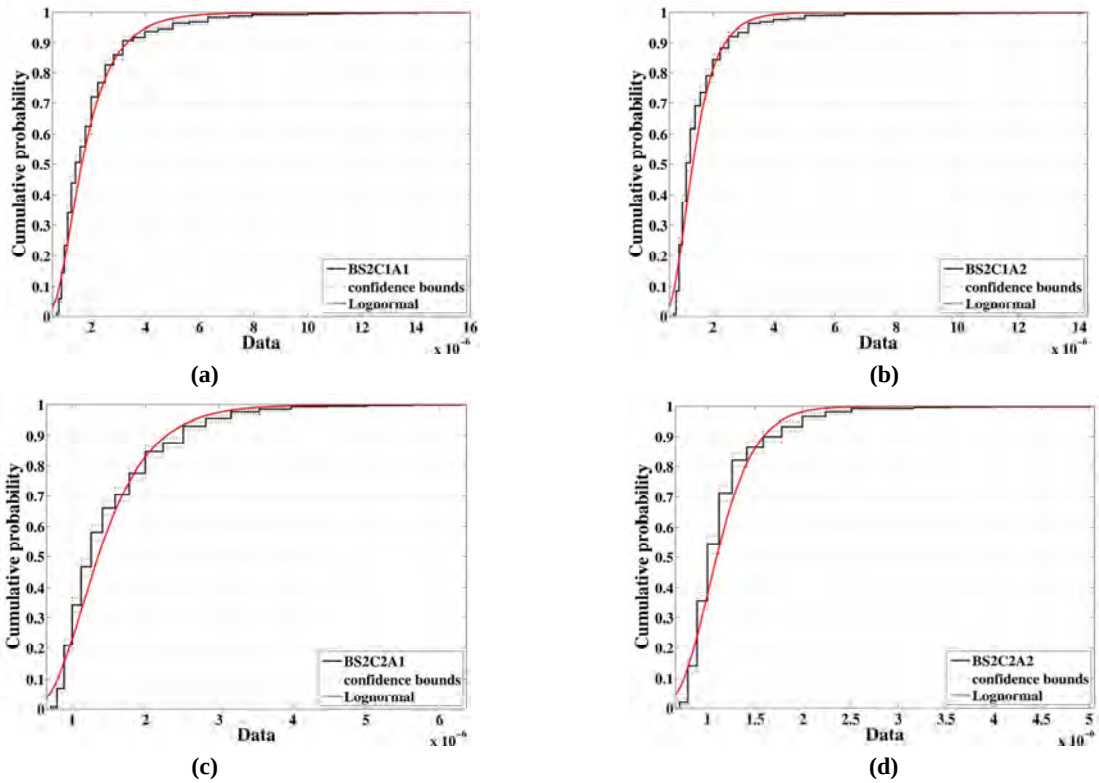



**Figure 15. CDFs of detected voltage at BS2 for (a) A1 on C1, (b) A2 on C1, (c) A1 on C2 and (d) A2 on C2. (Smooth curve represents the best fit log-normal distribution.).**

**Table 2. Correlation coefficient of signals received by different base-stations.**

| *Collar* | *Antenna* | *Correlation Coefficient between BS1 and BS2* |
|---|---|---|
| C1 | A1 | –0.2646 |
| | A2 | –0.3062 |
| C2 | A1 | –0.0212 |
| | A2 | –0.2217 |

The mean received power for each measurement is indicated by a horizontal line in Figure 16. The correlation coefficients between BS1 and BS2 signals are presented in Table 2.

Base-station diversity clearly offers advantage. The consistently small negative correlation is interpreted as being due to essentially zero short-term correlation due to the physically independent multipath propagation structure experienced by the base-stations, and a longer-term negative correlation due to the changes in distance between collar and base-stations as the animals move.

## 4. Diversity Gain

Two types of diversity gain have been evaluated. Antenna diversity relates to the advantage obtained by having two antennas on one collar. Base-station diversity relates to the advantage obtained by having two base-stations.

### 4.1. Antenna Diversity

Since there are two collars each with two antennas and two base-stations, the trial contains four independent instances of antenna diversity. These are: (i) diversity collar 1 to BS1, (ii) diversity collar 2 to BS1, (iii) diversity collar 1 to BS2 and (iv) diversity collar 2 to BS2.

Figure 17 shows the cdfs corresponding to each of these antenna diversity instances. In each sub-figure, there are four curves: the base-station signal received from A1, the base-station signal received from A2, the mean base-station signal received (calculated using A1 and A2), and the maximum base-station signal received (selected from A1 and A2). The mean base-station signal is adopted as the reference with which to calculate diversity gain. This is because either of the diversity antennas could be adopted as the reference. Taking the mean therefore reduces statistical noise to give a better estimate of expected value. The mean value is calculated from the received powers in dBm. This results in a final diversity gain (in dB) corresponding to the geometric mean of the diversity gains expressed as ratios obtained using A1 and A2 references.

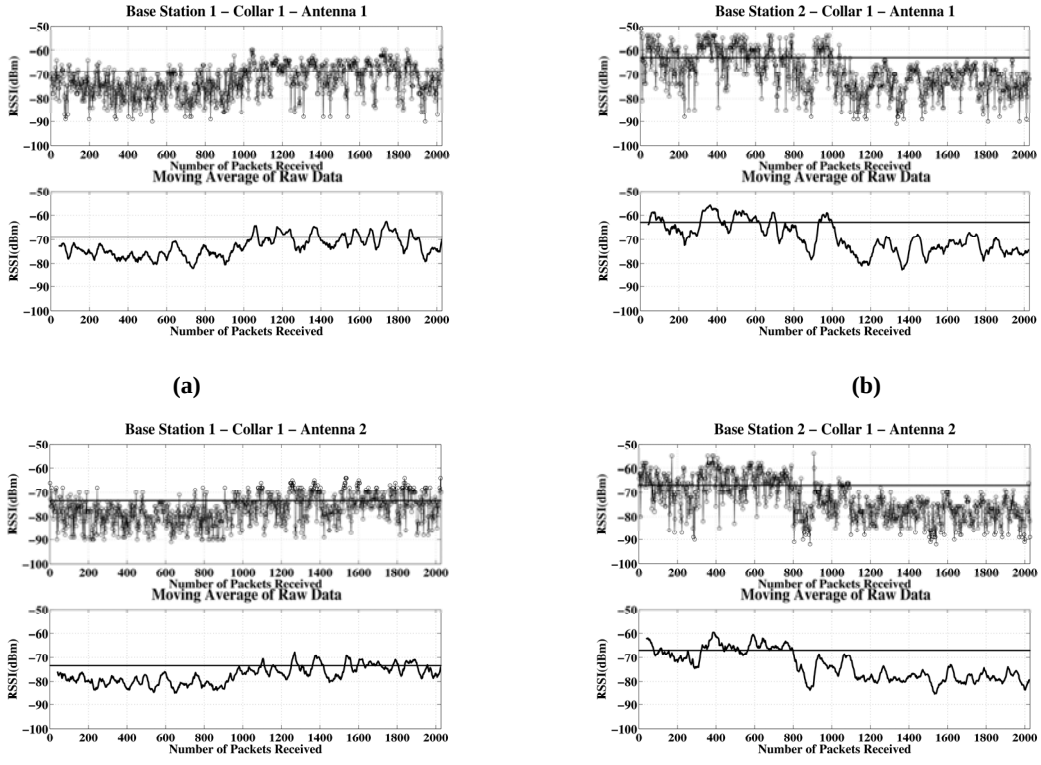



**Figure 16. Data received by A1 on C1 at (a) BS1 and (b) BS2, and data received by A2 on C1 at (c) BS1 and (d) BS2.**

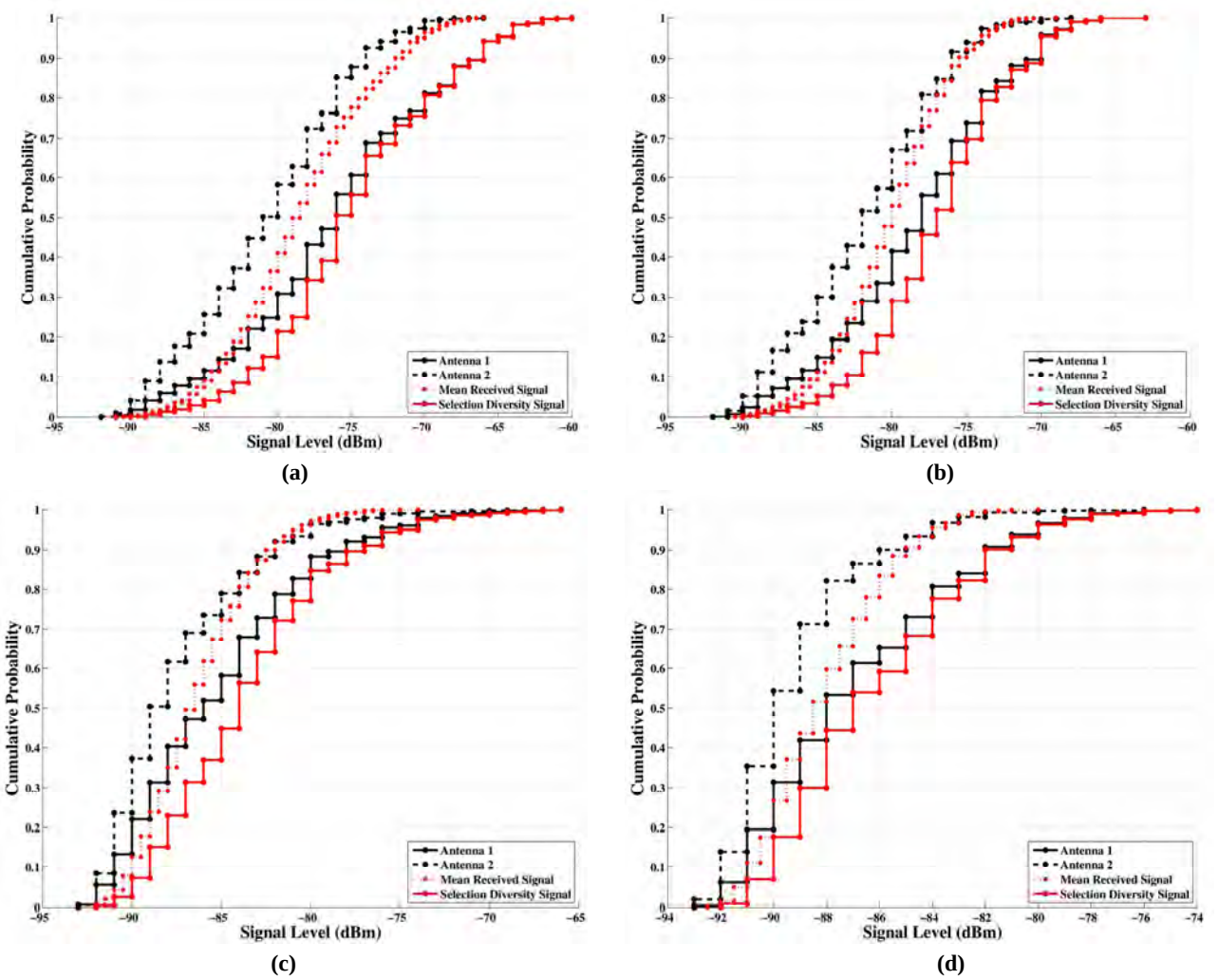



**Figure 17. Received signal power (dBm) for antenna 1 and antenna 2, mean received signal (dBm), and selection diversity signal (dBm) for (a) C1 to BS1, (b) C2 to BS1, (c) C1 to BS2 and (d) C2 to BS2.**

Figure 18 shows the cdfs of the expected diversity gain for (a) C1 to BS1, (b) C2 to BS1, (c) C1 to BS2 and (d) C2 to BS2.

The expected diversity gain (dB) is the difference between the selected diversity signal power (dBm) and the mean signal (dBm).

The median diversity gain averaged over all four instances is 4.4 dB. The 10% and 90% diversity gain exceedances averaged over all four instances are 5.7 dB and 0.5 dB, respectively.

## 4.2. Base-Station Diversity

There are four instances of base-station diversity gain. These are A1 on C1 to BS1 and BS2, A2 on C1 to BS1 and BS2, A1 on C2 to BS1 and BS2, and A2 on C2 to BS1 and BS2. The base-station diversity cumulative distributions are calculated in an identical way to the antenna diversity cumulative distributions. The results are shown in Figures 19 and 20.

The median diversity gain averaged over all four instances is 4.5 dB. The 10% and 90% diversity gain exceedances averaged over all four instances are 8.5 dB and 1.1 dB, respectively.

## 4.3. Overall Diversity

There are four instances of antenna and base-station diversity gain with regards to each collar. These are A1 to BS1, A2 to BS1, A1 to BS2 and A2 to BS2.

Figure 21 shows the four curves for each instance, themean received signal (dBm) and the maximum signal received for the first collar.

Figure 22 shows the four curves for each instance, the mean received signal (dBm) and the maximum signal received for the second collar.

The overall diversity gain (i.e. the diversity gain available from the combined antenna diversity and

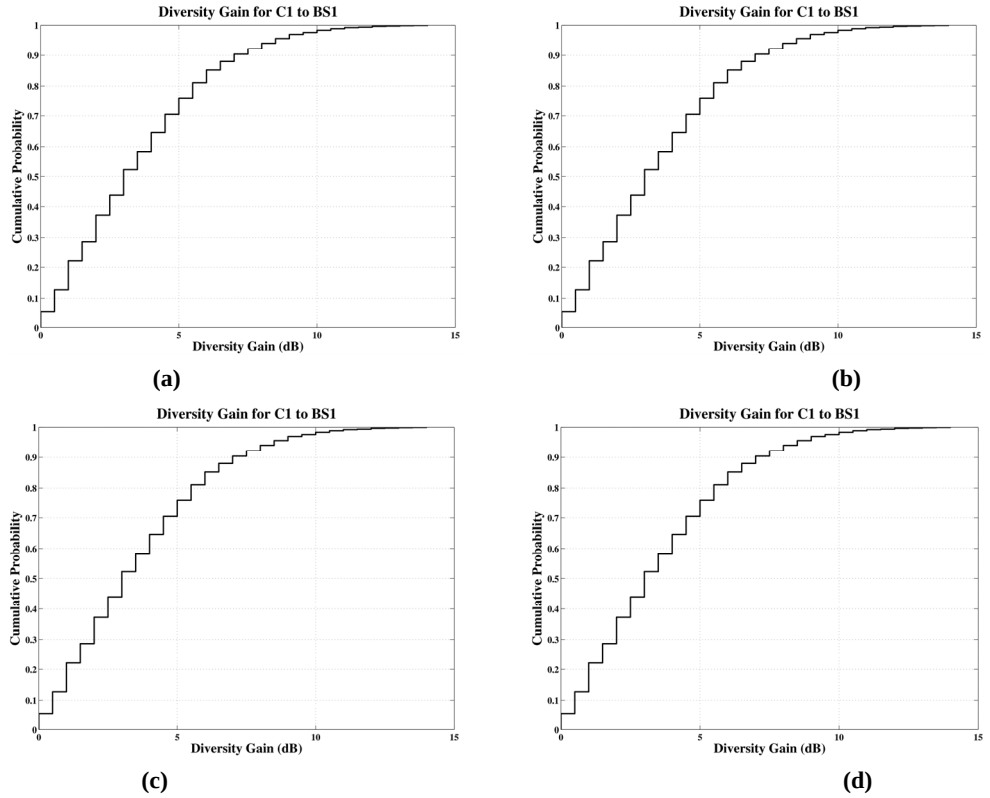



**Figure 18. Antenna diversity gain for each instance: (a) C1 to BS1, (b) C2 to BS1, (c) C1 to BS2 and (d) C2 to BS2.**

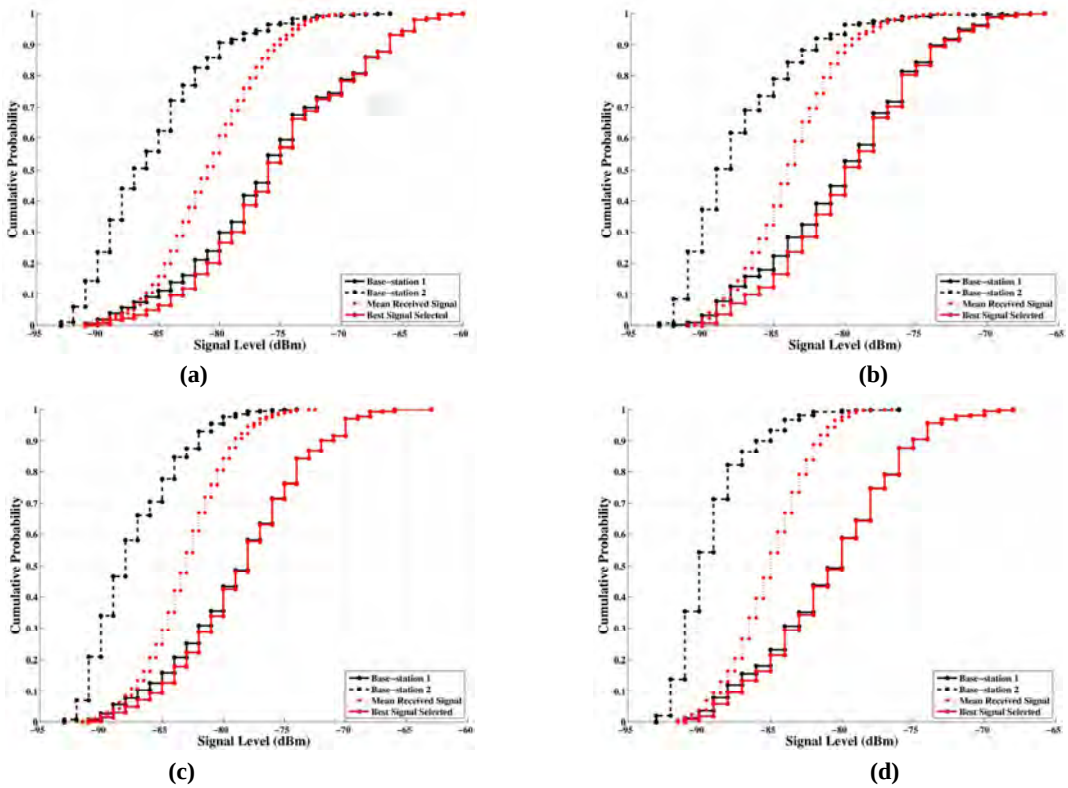



**Figure 19. Received signal power (dBm) for BS1 and BS2, mean received signal (dBm), and selection diversity signal (dBm) for (a) A1 of C1 to BS1 and BS2, (b) A2 of C1 to BS1 and BS2, (c) A1 of C2 to BS1 and BS2 and (d) A2 of C2 to BS1 and BS2.**

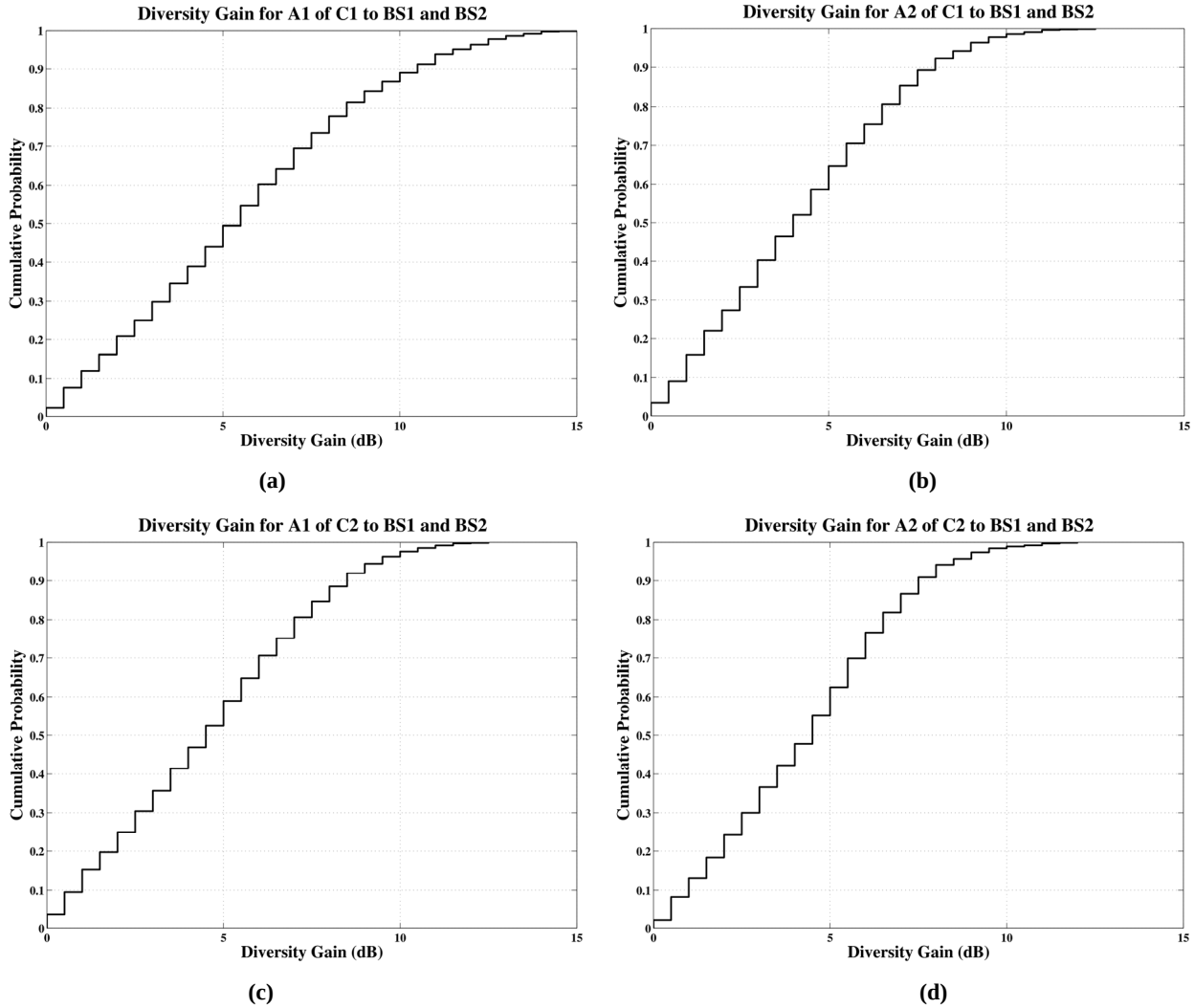



**Figure 20. Base-station diversity gain for each instance: (a) A1 of C1 to BS1 and BS2, (b) A2 of C1 to BS1 and BS2, (c) A1 of C2 to BS1 and BS2 and (d) A2 of C2 to BS1 and BS2.**

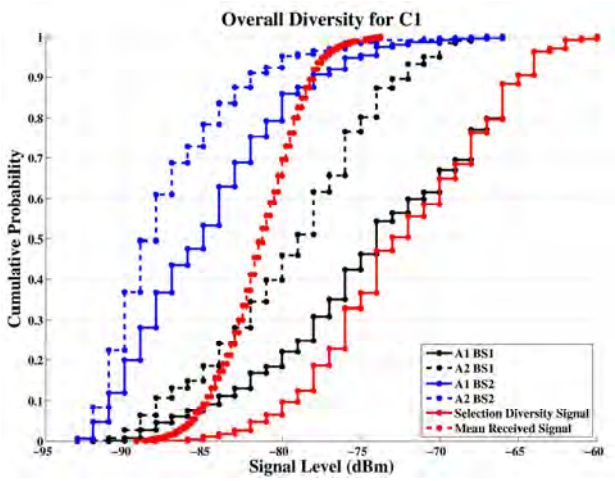



**Figure 21. Overall selection and mean diversity for the first collar (two antennas and two base-stations).**

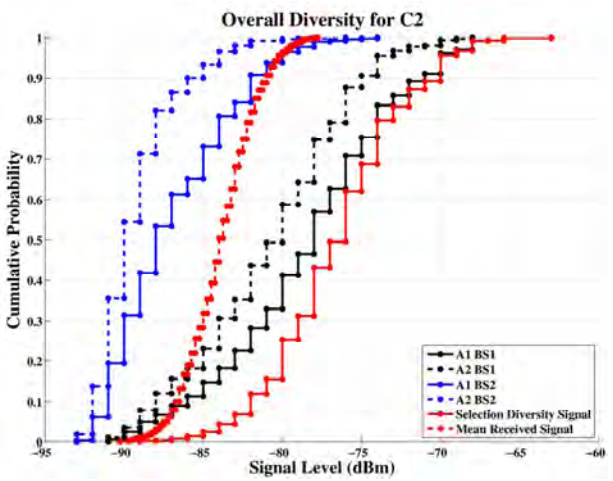



**Figure 22. Overall selection and mean diversity for the second collar (two antennas and two base-stations).**

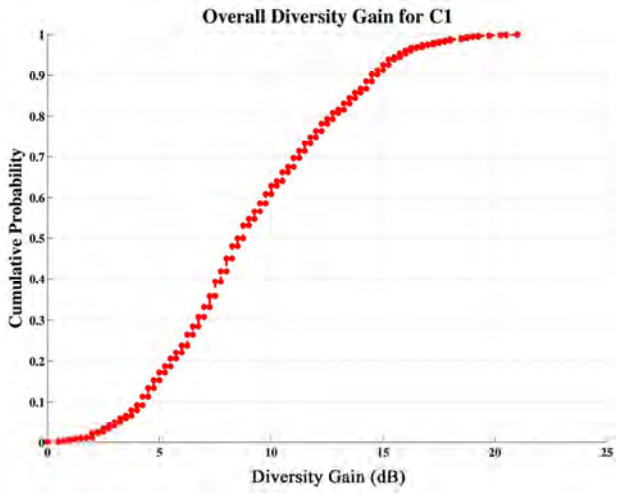



**Figure 23. Overall diversity gain for the first collar.**

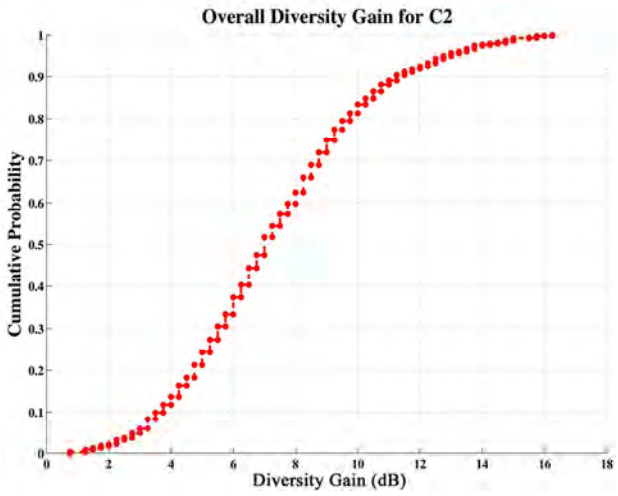



**Figure 24. Overall diversity gain for the second collar.**

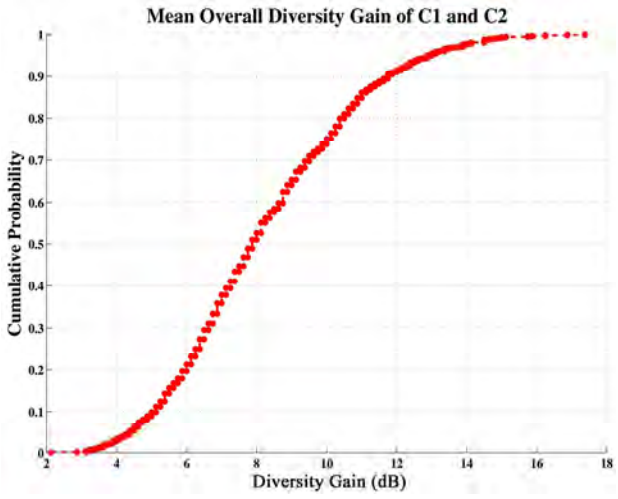



**Figure 25. Mean overall diversity gain for the two collars.**

base-station diversity) for the first and second collar is shown in Figure 23 and 24, respectively.

The median overall diversity gain for the first and second collar is 8.1 dB and 7 dB, respectively. The 10% exceedances are 14.3 and 11.2 dB for the first and second collar, respectively. The 90% exceedances are 4.1 dB for the first and 3.5 dB for the second collar, respectively.

The mean overall diversity gain is shown in Figure 25. This is the mean value calculated from the two individual collar diversity gains. The median overall mean diversity is 7.9 dB. The 10% and 90% diversity gain exceedances are 11.6 dB and 5.1 dB, respectively.

## 5. Conclusions

Antenna and base-station diversity has been applied to the wireless monitoring of farm animals. The statistical distributions of received signals and antenna/base-station signal correlations have been summarised. The advantage offered by selection diversity has been evaluated. The overall (antenna and base-station) diversity gain offered at each collar has also been studied.

## 6. References

[1] I. F. Akyildiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Vol. 38, No. 4, pp. 393–422, March 2002.

[2] The Defense Advanced Research Projects Agency (DARPA), Self-healing minefield, http://www.darpa.mil/ sto/ smallunitops/shm/index.htm.

[3] K. S. Low, W. N. N. Win, and M. J. Er, "Wireless sensor networks for industrial environments," In Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, Vol. 2, pp. 271–276, November 28–30, 2005.

[4] S. Y. Lau, T. H. Chang, S. Y. Hu, H. J. Huang, L. de Shyu, C. M. Chiu, and P. Huang, "Sensor networks for everyday use: The bl-live experience," In Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Vol. 1, pp. 336–343, 2006.

[5] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, M. Naeve, B. Heile, and V. Bahl, "Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks," Vol. 40, No. 8, pp. 70–77, August 2002.

[6] J. K. Hart, K. Martinez, and R. Ong, "Environmental sensor networks," Vol. 37, No. 8, 2004.

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," In Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applica-

tions (WSNA'02), Atlanda, USA, 2002.

[8] H. G. Goh, M. L. Sim, and H. T. Ewe, "An overview of rice field monitoring using wireless sensor networks and mobile internet application," In Proceedings of the 8th International Conference on Electronics, Information and Communication (ICEIC'06), Vol. 1, pp. 114–117, Ulaanbaatar, Mongolia, June 2006.

[9] Wireless mobile ad-hoc sensor networks for very large scale cattle monitoring, Berlin, Germany, August 2006.

[10] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," Vol. 3, No. 1, pp. 38–45, 2004.

[11] K. Mayer, K. Taylor, and K. Ellis, "Cattle health monitoring using wireless sensor networks," In The 2nd IASTED International Conference on Communication and Computer Networks, November 2004.

[12] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences on zebranet," In Proceedings of SenSys, Baltimore, Maryland, USA, 2004.

[13] I. McCauley, B. Matthews, L. Nugent, A. Mather, and J. Simons, "Wired pigs: Ad-hoc wireless sensor networks in studies of animal welfare," In Proceedings of EmNetS-II Embedded Networked Sensors The Second IEEE Workshop on, pp. 29–36, May 30–31, 2005.

[14] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain, "Animal behaviour understanding using wireless sensor networks," In Local Computer Networks, Proceedings 2006 31st IEEE Conference on, pp. 607–614, November 2006.

[15] P. Corke and P. Sikka, "Results from the farm," http://eecs.harvard.edu/emnets/papers/corkeEmnets06.pdf (Last accessed on 24/02/08).

[16] K. Sasloglou, I. A. Glover, Kae-Hsiang Kwong, and I. Andonovic, "Wireless sensor network for animal monitoring using both antenna and base-station diversity," In Proceedings of 11th IEEE Singapore International Conference on Communication Systems ICCS 2008, pp. 27–33, November 19–21, 2008.

[17] http://www.xbow.com, (Last accessed on 18th May 2008).

[18] http://www.hyperlinktech.com, (Last accessed on 18th May 2008).

[19] A. Papoulis, "Probability, random variables, and stochastic processes," Mc-Graw Hill, third Edition, 1991.

[20] T. S. Rappaport, "Wireless communications principles and practices," Prentice Hall PTR, second Edition, 2002.

*Wireless Sensor Network*, **2009, 1, 397-406**
doi:10.4236/wsn.2009.15048 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Modeling and Analysis of Random Periodic Spectrum Sensing for Cognitive Radio Networks

**Caili GUO, Zhiming ZENG, Chunyan FENG, Qi LIU**
*School of Information and Communication Engineering,*
*Beijing University of Posts and Telecommunications, Beijing, China*
*Email*: *guocaili@bupt.edu.cn*
*Received June* 18, 2009; *revised August* 20, 2009; *accepted August* 21, 2009

## Abstract

A random periodic spectrum sensing scheme is proposed for cognitive radio networks. The sensing period, the transmission time for primary users and cognitive radios are extended to general forms as random variables. A generalized Markov analytical model for sensing period optimization is presented, and the applications of the proposed analytical model by using examples involving primary user systems with both voice and data traffic are illustrated. The analysis and numerical results show that sensing period does affect the maximum rewards of the channel, and the analytical model is justified by its flexibility since it uses general forms of the sensing period, the transmission time for primary users and cognitive radios. Hence the model can be easily adapted for the analysis of many different applications.

**Keywords:** Cognitive Radio Networks, Random Periodic Spectrum Sensing, Generalized Markov Process, the Optimal Sensing Period

## 1. Introduction

Due to energy consumption and hardware implication of Cognitive Radios (CRs), it is undesirable and impractical to assume the spectrum sensing to be continuous. In a practical CR network, such as an IEEE 802.22 network [1], a periodic spectrum sensing scheme where the spectrum is sensed periodically to determining the presence/absence of Primary Users (PUs) is preferable. The sensing time and sensing period are two key sensing parameters for periodic sensing scheme. The former is a pre-defined amount of time used to achieve the desirable level of detection quality and is mainly depended on PHY-layer sensing methods, such as energy detection, matched filter and feature detection. And the latter defined as the interval between two successive detection processes has a significant impact on the sensing efficiency of CRs. In the case of the sensing period is relatively large, both some opportunities may go undiscovered and interference to PUs may occur, whereas blindly reducing the sensing period is not desirable either, as it increases the sensing overhead. Thus the design of any periodic sensing scheme involves balancing the tradeoffs among spectrum utilization, interference to PUs, and sensing overhead by selecting an appropriate sensing period. We usually consider a spectrum consist of several channels, and each channel can be a frequency band with certain bandwidth, a spreading code in a CDMA network or a set of tones in an OFDM system. Here we use the term channel broadly.

In CR networks, the control of quiet period, during which all CRs should suspend their transmissions so that any CR monitoring the channel may observe the presence/absence of PU signals without interference, can be synchronous or asynchronous [1–2]. Accordingly there are two kinds of periodic sensing schemes: One is synchronous sensing period and the other is asynchronous sensing period. Most of the existing works focused on the synchronous sensing period schemes [3–4]. As a simple solution for design and implementation, the synchronous sensing period scheme sets a pre-determined fixed sensing period for all channels. While it does not need the scheduling of quiet period for each channel among CRs, it shows less flexible. Recent researches [5–7] showed that the asynchronous sensing period scheme is more favorable, in which sensing period can be adjusted adaptively according to the channel-usage characteristics of each channel by the MAC-layer sensing protocol or through a dedicated control channel [8]. Kim [5] proposed an adaptation algorithm in which the

The material in this paper is based on "Random Periodic Spectrum Sensing with Sensing Period Optimization for Cognitive Radio Networks", by Caili Guo, Zhiming Zeng, Chunyan Feng and Qi Liu which appeared in the proceedings of 11th IEEE International Conference on Communications Systems, ICCS 2008, Guangzhou, China, November 2008.

optimal sensing period is uniquely determined for each channel to maximize the discovery of opportunities as well as minimize the delay in locating an idle channel. However, this approach clearly did not consider the impact of sensing period selection on interference to PUs. In [6], we extended [5] to a Flexible Sensing Period (FSP) mechanism that introduces the "period control factor" to control each channel's sensing period adaptively to tradeoff undiscovered opportunities and interference to PUs with sensing overhead effectively, but as far as each channel is concerned, the FSP also consider that sensing period is still fixed. In order to combat the fluctuation of sensing period induced by the varying of channel-usage characteristics, in [7] we described a Fuzzy Spectrum Sensing Period Optimization (FSSPO) algorithm where each channel's sensing period is adaptively adjusted in real time with fuzzy logic and parameters optimization.

Existing approaches of asynchronous sensing period in [5,6] and [7] only considered how to adjust sensing period which is usually regarded as a constant once it is determined, and they all assumed that the sensing results are perfect. In this paper, the random periodic sensing scheme we proposed extends the sensing period, the time of transmission for primary users and cognitive radios to random variables and more practical situation where sensing error exists is considered. As the PUs have the highest priority, we also introduce a back-off mechanism where a random back-off time is generated whenever PUs release the channel and CRs have to delay for the back-off time before occupying the channel. Here we focus on how to model the proposed random periodic sensing scheme to a generalized Markov process and how to derive the optimal sensing period. To support the proposed analytical model for sensing period optimization, we also illustrate the applications of the analytical model by using examples involving PU systems with both voice and data traffic.

The rest of the paper is organized as follows. Section 2 introduces the random periodic spectrum sensing scheme. A generalized Markov analytical model for sensing period optimization is constructed in Section 3. Then in Section 4, how to obtain performance measures of channels is considered. Example applications of the proposed analytical model for real networks are illustrated in Section 5. Numerical examples are presented and discussed in Section 6. Finally, we conclude the paper and suggest future directions in Section 7.

## 2. Random Periodic Spectrum Sensing Scheme

In CR networks, a channel usually could be modeled as an ON-OFF source alternating between ON (busy) and OFF (idle) periods depending on PUs' channel-usage pattern. The sojourn time of a ON period is used for transmission of PUs themselves and that of a OFF period captures the time period in which the channel can be utilized by CRs' transmission without causing any harmful interference to PUs. The distribution of the sojourn time in the ON state can assumed to be general, and so is that in the OFF state. Thus the ON-OFF channel-usage stochastic process describing the behavior of the channel occupation can form an alternative renewal process. A renewal period models a time period in which the PUs and CRs occupy the channel once alternatively. Hence there are only busy and idle two possible states in a renewal period accordingly.

Considering that the sensing period of each channel is a random variable and sensing errors are possible present at any moment as well as a back-off mechanism is introduced, in this paper we further subdivide the channel in a renewal period into five kinds of states: normal busy, available idle, delay idle, false alarm and miss detection. Below we will describe each state in detail.

Normal busy and available idle are two kinds of normal available states. The former denotes that PUs are being served normally and the latter stands for the channel are being utilized for the transmission of CRs. When the channel is in normal available, it is sensed once every random time interval $T$, i.e. sensing period, to make sure whether it is in normal busy or available idle.

As soon as the service of PUs completes, a random back-off time interval $T_0$ is generated to prevent CRs from occupying the channel at once, within which the channel is in delay idle state. The back-off mechanism can help decrease both the connection cost generated by switching channel state frequently and the short-term interference probability induced by non-negligible delay for relinquishing bands by CRs.

Corresponding to the binary hypotheses test of spectrum sensing: $B_0$ (null hypothesis indicating that the sensed channel is available for CRs) vs. $B_1$ (alternative), there are two kinds of sensing errors: false alarm (the overlook of an available channel) due to mistaking $B_0$ for $B_1$ and miss detection (the mistake of identifying an unavailable channel as an opportunity) due to mistaking $B_1$ for $B_0$. When false alarm or miss detection is present, the channel will transfer to false alarm or miss detection state. The presence of sensing errors also has significant effects on the performance of sensing schemes.

Common to most of the periodic spectrum sensing schemes, the sensing period selection of random periodic spectrum sensing scheme has strong impact on the sensing efficiency. In order to analyze the optimal sensing period effectively, an analytical model for sensing period optimization is constructed in next section.

## 3. Analytical Model

Owing to the multiplicity of conditionality and correlation that exists among the various random variable in-

volved in the random periodic sensing scheme, the analysis and performance evaluation, especially the determination of optimal sensing period is usually difficult. Therefore, a set of simplifying assumptions is to be made for analytical model to be tractable. We assume the following

- The sojourn time of normal busy and available idle are continuous random variables and drawn from general cumulative distribution functions (c.d.f.) represented by $F_1(t)$ and $F_2(t)$ respectively. Suppose the probability density functions (p.d.f.) for $F_i(t)$ ($i$=1,2) are $f_i(t)$, means are $\omega_i^{-1}$, and $F_i(t) = \int_0^t f_i(x)dx = 1 - \exp[-\int_0^t \omega_i(x)dx]$, $\omega_i^{-1} = \int_0^\infty t dF_i(t)$.
- The sensing period $T$ is also a continuous random variable and follows an arbitrary distribution with c.d.f., p.d.f. and mean are $G_1(t)$, $g_1(y)$ and $\upsilon_1^{-1}$, respectively, and $G_1(t) = \int_0^t g_1(y)dy = 1 - exp[-\int_0^t \upsilon_1(y)dy]$ $E(T) = \upsilon_1^{-1} = \int_0^\infty t dG_1(t)$.
- Delay idle state can transfer to normal busy or available idle. It is assumed that if a PU reappears during $T_0$ he can claim the channel and the delay idle will transfer to normal busy with a constant rate $\gamma$. Otherwise, if the back-off timer expires and no one claims the channel, delay idle transfers to available idle instead. In this case, $T_0$ follows an arbitrary distribution, and its c.d.f., p.d.f. and mean are $G_2(t)$, $g_2(y)$ and $\upsilon_2^{-1}$, respectively, and

$$G_2(t) = \int_0^t g_2(y)dy = 1 - exp[-\int_0^t \upsilon_2(y)dy]$$

$$E(T_0) = \upsilon_2^{-1} = \int_0^\infty t dG_2(t)$$

- We assume that $\alpha$ and $\beta$ denote false alarm and miss detection probabilities respectively, and the sojourn times of false alarm and miss detection are arbitrarily distributed with c.d.f.s $H_i(t), i = 1,2$. Let $h_i(t)$ and $\varphi_i^{-1}$ be their p.d.f.s and means, and

$$H_i(t) = \int_0^t h_i(z)dz = 1 - exp[-\int_0^t \varphi_i(z)dz],\ \varphi_i^{-1} = \int_0^\infty t dH_i(t)$$

When miss detection occurs the channel transfers to delay idle, whereas when false alarm occurs, in order to render mathematical tractability, we also assume that the channel skips normal occupy and transfers to delay idle too, for the normal occupy time included is small enough compared to total time of the long-run channel and can be neglected.

- We assume that the sensing time is small relative to distribution parameters $\omega_1, \omega_2, \gamma, E(T), E(T_0)$ and $\varphi_i^{-1}(i = 1,2)$, and can be negligible. It is also assumed that channel is in delay idle initially, and all random variables are mutually independent.

Consider the stochastic process $\{S(t), t \geq 0\}+$, where $S(t)$ denotes the state of the channel at time $t$, as following

- 0 stands for the channel is in delay idle, and the channel is occupied neither by PUs nor by CRs.
- $(i,k)$ means that the channel is in normal available states, where $i$ =0/1 represents that the channel is in available idle/normal busy and $k$ stands for the times of the channel has been sensed, k =0,1,2, ⋯ .
- $(2, j)$ denotes that the sensing error is present, where $j$ =1/2 means that the system is in false alarm/miss detection.

It is easy to see that, $\{S(t), t \geqslant 0\}$ is a non-Markovian stochastic process. In order to make the process Markovian, we need to incorporate the missing information by adding "supplementary variables" to the state description. Hence at time t, let $X_i(t)$ ($i$ =1/2) be the remaining normal busy/available idle time, $Y_i(t)$ ($i$ =1/2) the remaining sensing/delay time, and $Z_i(t)$ ($i$ =1/2) the remaining false alarm /miss detection time. Formally, the evolution of the stochastic process describing the dynamic behavior of the channel can be fully characterized by a generalized Markov process $\{S(t), X_i(t), Y_i(t), Z_i(t) \mid t \geq 0\}$, and the following state probabilities are defined

$$\begin{cases} P_{ik}(t,x,y)dx = P\{S(t) = (i,k), x < X_i(t) \leq x + dx, \\ \quad y < Y_1(t) \leq y + dy\}, i = 0,1; k = 0,1,2,\cdots \\ P_0(t,y)dy = P\{S(t) = 0, y < Y_2(t) \leq y + dy\} \\ P_{2j}(t,z)dz = P\{S(t) = (2,j), z < Z_i(t) \leq z + dz\}, j = 1,2 \end{cases}$$

Notations:

$$\bar{\cdot}: \bar{F}_i(t) = 1 - F_i(t), *: f_i^*(s) = \int_0^\infty f_i(t)e^{-st}dt, \bar{F}_i^*(s) = [1 - f_i^*(s)]/s$$

$$f_{\omega 1}, f_{\omega 2}, g_\gamma : f_{\omega 1} = f^*(s + \omega_1), f_{\omega 2} = f^*(s + \omega_2), g_\gamma = g^*(s + \gamma)$$

$$\hat{f}_{\omega 1}, \hat{f}_{\omega 2}, \hat{g}_\gamma : \hat{f}_{\omega 1} = f^*(\omega_1), \hat{f}_{\omega 2} = f^*(\omega_2), \hat{g}_\gamma = g^*(\gamma)$$

$$\bar{\alpha}, \bar{\beta} : \bar{\alpha} = 1 - \alpha, \bar{\beta} = 1 - \beta$$

$$M_1, M_2, M_3, M_4 : M_1 = \omega_1 - \omega_2, M_2 = s + \omega_2, M_3 = s + \omega_1, M_4 = s + \gamma$$

$$E_{\omega 1x}, E_{\omega 2x}, E_{\gamma y} : E_{\omega 1x} = \exp[-(s + \omega_1)x], E_{\omega 2x} = \exp[-(s + \omega_2)x],$$

$$E_{\gamma y} = \exp[-(s + \gamma)y]$$

The possible states of channels and the transitions among them are shown in Figure 1.

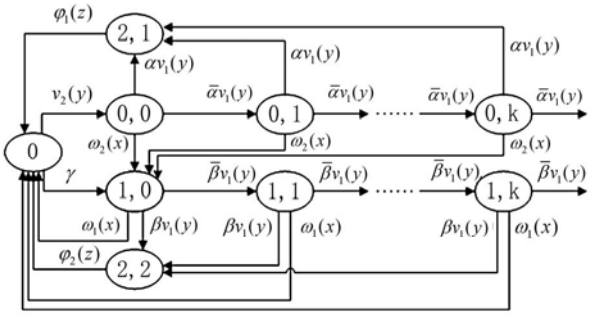



**Figure 1. The state transition model of the generalized Markov process.**

According to Figure 1, a few new performance measures could be defined. The probability of the channel in available idle state, normal busy state and delay idle state are Available Idle Probability, Normal Busy Probability and Delay Idle Probability, respectively. Sensing Frequency is defined the frequency of the occurrence of channel sensing when the channel is in state $(i,k), i = 0,1$. False alarm occurs if and only if the channel transfers from state $(0,k)(k = 0,1,2\cdots)$ to state (2,1); while miss detection occurs if and only if the channel transfers from state $(1,k)(k = 0,1,2\cdots)$ to state (2,2). So False Alarm Frequency and Miss Detection Frequency are defined accordingly. Also define $R_1(t)$, $R_2(t)$, $L_0(t)9$, $L_1(t)$, $L_2(t)$ and $L_3(t)$ are the Instantaneous Probability of Available Idle, Normal Busy, Delay Idle, Instantaneous Frequency of Sensing, False Alarm and Miss Detection at an arbitrary time *t*, respectively. Then define $R_1$, $R_2$, $L_0$, $L_1$, $L_2$ and $L_3$ are the steady state forms of $R_1(t)$, $R_2(t)$, $L_0(t)$, $L_1(t)$, $L_2(t)$ and $L_3(t)$, respectively, e.g., $R_1 = \lim_{t\to\infty} R_1(t)$.

In CR networks, each channel will go through one or all kinds of states in which available idle and normal busy generate rewards by the using for the transmission of CRs and PUs, respectively, delay idle and false alarm waste opportunities, miss detection induces interference to PUs, and sensing has overhead. It is assumed that the expected rewards per unit time generated by the channel are $e_1$ or $e_2$ when the channel is in available idle or normal busy, respectively. And the expected losses per unit time induced by the delay of channel is $c_0$, the expected cost of each sensing is $c_1$, the expected false alarm and miss detection expenses every time are $c_2$ and $c_3$, respectively. We also assumed that expected total rewards generated by the channel during (0,t] are *R*, then

$$R(t) = \sum_{i=1}^{2} e_i \int_0^t R_i(t)dt - c_0 \int_0^t L_0(t)dt - \sum_{j=1}^{3} c_j \int_0^t L_j(t)dt \quad (1)$$

where $e_1, e_2, c_0, c_1, c_2$ and $c_3$ are weighting factors ($e_1 + e_2 + c_0 + c_1 + c_2 + c_3 = 1$). Taking Laplace transform on both sides of (1), we have

$$R^*(s) = [\sum_{i=1}^{2} e_i R_i^*(s) - c_0 L_0^*(s) - \sum_{j=1}^{3} c_j L_j^*(s)] / s \quad (2)$$

And the expected rewards per unit time in the steady state is

$$\begin{aligned} R &= \lim_{t\to\infty} R(t)/t = \lim_{s\to 0} s^2 R^*(s) \\ &= \sum_{i=1}^{2} e_i R_i - c_0 L_0 - \sum_{j=1}^{3} c_j L_j \end{aligned} \quad (3)$$

where performance measures $R_i$ ($i$ = 1, 2), $L_0$ and $L_j$ ($j$ = 1, 2, 3) could be expressed as functions of the mean sensing period $E(T)$ (denoted by $\bar{x}$) and all of them will be illustrated how to obtain in Section 4. Then taking suitable value for $\bar{x}$ to make ***R*** maximum can bring out the optimal sensing period $\bar{x}'$. That is

$$\bar{x}' = \arg\max_{\bar{x}} (\sum_{i=1}^{2} e_i R_i - c_0 L_0 - \sum_{j=1}^{3} c_j L_j) \quad (4)$$

## 4. Derivation of Performance Measures

In order to derive performance measures for searching the optimal sensing period, the steady state probabilities with exact solutions in closed form should be calculated. How to calculate them using probability analysis and supplementary variables method is discussed below.

According to Figure 1, we have the following differential difference equations

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial x} + \omega_2(x) + v_1(y)\right] p_{0k}(t,x,y) = 0 \quad k = 0,1,2\cdots \quad (5)$$

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial x} + \omega_1(x) + v_1(y)\right] p_{10}(t,x,y) = \sum_{k=0}^{\infty} \mu p_{0k}(t,x,y) \quad (6)$$

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial x} + \omega_1(x) + v_1(y)\right] p_{1k}(t,x,y) = 0 \quad k = 1,2\cdots \quad (7)$$

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial y} + \gamma + v_2(y)\right] p_0(t,y) = 0 \quad (8)$$

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial z} + \varphi_1(z)\right] p_{21}(t,z) = 0 \quad (9)$$

$$\left[\frac{\partial}{\partial t} + \frac{\partial}{\partial z} + \varphi_2(z)\right] p_{22}(t,z) = 0 \quad (10)$$

The above equations are to be solved under the boundary conditions

$$\begin{aligned} &p_0(t,0) \\ &= \sum_{k=0}^{\infty} \int_0^{\infty} \lambda p_{1k}(t,x,y)dxdy + \int_0^{\infty} \varphi_1(z) p_{21}(t,z)dz \\ &\quad + \int_0^{\infty} \varphi_2(z) p_{22}(t,z)dz + \delta(t) \end{aligned} \quad (11)$$

$$p_{00}(t,0) = \int_0^{\infty} v_2(y) p_0(t,y)dy \quad (12)$$

$$\begin{aligned} &p_{0k}(t,x,0) \\ &= \int_0^{\infty} \bar{\alpha}\omega_2(x) p_{0k-1}(t,x,y)dx\,dy \quad k = 1,2\cdots \end{aligned} \quad (13)$$

$$p_{10}(t,0) = \int_0^{\infty} \gamma p_0(t,y)dy \quad (14)$$

$$p_{1k}(t,x,0) = \int_0^{\infty} \bar{\beta} v_1(y) p_{1k-1}(t,x,y)dx\,dy \quad k = 1,2\cdots \quad (15)$$

$$p_{21}(t,0) = \sum_{k=0}^{\infty} \int_0^{\infty} \alpha v_1(y) p_{0k}(t,x,y)dx \quad (16)$$

$$p_{22}(t,0) = \sum_{k=0}^{\infty} \int_0^{\infty} \beta v_1(y) p_{1k}(t,x,y)dxdy \quad (17)$$

And the initial conditions

$$p_0(0,y)=\delta(y) \tag{18}$$

Taking Laplace transforms of Equations (5)–(18) with respect to $t$ and solving the equations, we can derive

$$P_{0k}^*(s,x,y) = M_1M_4(1-\bar{\alpha}f_{\omega 2})(1-\bar{\beta}f_{\omega 1})\bar{\alpha}^k(f_{\omega 2})^k g_\gamma E_{\omega 2x}\overline{F_2}(x)\times\overline{G_1}(y)/D(s) \quad k=0,1,2... \tag{19}$$

$$P_{10}^*(s,x,y) = \{[\gamma M_1(1-g_\gamma)(1-\bar{\alpha}f_{\omega 2})(1-\bar{\beta}f_{\omega 1})-\mu M_4 g_\gamma(1-\bar{\beta}f_{\omega 1})]\times E_{\omega 1x}\overline{F}_1(x)\overline{G}_1(y)+\mu M_4 g_\gamma(1-\bar{\beta}f_{\omega 1})E_{\omega 2x}\overline{F}_1(x)\overline{G}_1(y)\}/D(s) \tag{20}$$

$$P_{1k}^*(s,x,y) = [\bar{\beta}\gamma f_{\omega 1}(1-g_\gamma)M_1(1-\bar{\alpha}f_{\omega 2})(1-\bar{\beta}f_{\omega 1})+\omega_2\bar{\beta}g_\gamma(f_{\omega 2}-f_{\omega 1})\times M_4(1-\bar{\beta}f_{\omega 1})](\bar{\beta}f_{\omega 1})^{k-1}E_{\omega 1x}\overline{F}_1(x)\overline{G}_1(y)/D(s) \quad k=1,2... \tag{21}$$

$$P_0^*(s,y) = (M_4M_1-M_4M_1\bar{\alpha}f_{\omega 2}-M_4M_1\bar{\beta}f_{\omega 1}+M_4M_1\bar{\alpha}\bar{\beta}\times f_{\omega 2}f_{\omega 1})E_{\gamma y}\overline{G_2}(y)/D(s) \tag{22}$$

$$P_{21}^*(s,z) = [\beta\gamma M_1 f_{\omega 1}-(\beta\gamma M_1+\beta\omega_2M_4)f_{\omega 1}g_\gamma-\beta\gamma M_1\bar{\alpha}f_{\omega 2}f_{\omega 1}+\beta\omega_2M_4f_{\omega 2}g_\gamma+\beta\gamma M_1\bar{\alpha}f_{\omega 2}f_{\omega 1}g_\gamma]\exp(-sz)\overline{H_1}(z)/D(s) \tag{23}$$

$$P_{22}^*(s,z) = (\alpha M_1M_4f_{\omega 2}g_\gamma-\alpha M_1M_4\bar{\beta}f_{\omega 2}f_{\omega 1}g_\gamma)\exp(-sz)\times\overline{H_2}(z)/D(s) \tag{24}$$

where

$$\begin{aligned}D(s) &= (M_1M_4-M_1\omega_1\gamma/M_3)+(M_1\omega_1\gamma/M_3-M_1M_4\bar{\beta}f_{\omega 1})+(M_1\bar{\alpha}\times\omega_1\gamma/M_3-M_1\bar{\alpha}M_4)f_{\omega 2}+(M_1\bar{\alpha}\bar{\beta}M_4-M_1\bar{\alpha}\omega_1\gamma/M_4)f_{\omega 2}f_{\omega 1}\\ &-(M_1\beta\gamma\times f_{\omega 1}-M_1\beta\gamma\bar{\alpha}f_{\omega 1}f_{\omega 2})h_1^*(s)-\{[M_4\omega_1\omega_2/M_2-(M_4\omega_1\omega_2+M_1\omega_1\gamma)/M_3]+[(M_3\omega_1\omega_2+M_1\omega_1\gamma)/M_3-(M_4\omega_1-\mu\bar{\beta})/M_2]f_{\omega 1}\\ &+[(M_4\omega_1\omega_2\bar{\beta}+M_1\times\omega_1\gamma\bar{\alpha})/M_3-M_4\omega_1\omega_2/M_2]f_{\omega 2}+[M_4\omega_1\omega_2\bar{\beta}/M_2-(M_4\omega_1\omega_2\bar{\beta}+\gamma\bar{\alpha}M_1)/M_3]f_{\omega 2}f_{\omega 1}\}g_\gamma\\ &-[M_4\beta\omega_2f_{\omega 2}-(s\beta\omega_2-\omega_1\beta\gamma)f_{\omega 1}+M_1\beta\gamma\bar{\alpha}f_{\omega 2}f_{\omega 1}]h_1^*(s)\times g_\gamma-(M_1M_4\alpha f_{\omega 2}-M_1M_4\alpha\bar{\beta}f_{\omega 2}f_{\omega 1})h_2^*(s)g_\gamma\end{aligned} \tag{25}$$

In accordance with the definitions of $R_1(t)$ and $R_2(t)$, we obtain

$$R_1(t)=\sum_{k=0}^{\infty}\int_0^{\infty}P_{0k}(t,x)dx \tag{26}$$

$$R_2(t)=\sum_{k=0}^{\infty}\int_0^{\infty}P_{1k}(t,x)dx \tag{27}$$

Taking Laplace transform on both sides of Equations (26), (27) and using Equations (19)–(25), we get

$$R_1^*(s) = (M_1M_4g_\gamma-M_1M_4\bar{\beta}f_{\omega 1}g_\gamma)\overline{F_2^*}(s+\omega_2)/D(s)s \tag{28}$$

$$\begin{aligned}R_2^*(s) &= [\gamma M_1-(\gamma M_1+\omega_2M_4)g_\gamma-\gamma M_1\bar{\alpha}f_{\omega 2}\\ &+(\gamma M_1\bar{\alpha}+\omega_2\bar{\beta}M_4)g_\gamma f_{\omega 2}]\overline{F_1^*}(s+\omega_1)\\ &+(\omega_2M_4g_\gamma-\omega_2M_4\times\bar{\beta}f_{\omega 1}g_\gamma)\overline{F_2^*}(s+\omega_2)\end{aligned} \tag{29}$$

where, D(s) is given by Equation (25). By applying the limiting theorem of Laplace transform and L'Hospital's rule, we get

$$\begin{aligned}R_1 &= \lim_{t\to\infty}R_1(t)=\lim_{s\to 0}sR_1^*(s)\\ &= (M_1\gamma\hat{g}_\gamma-M_1\gamma\bar{\beta}\hat{f}_{\omega 1}\hat{g}_\gamma)\overline{F_2^*}(\omega_2)/D\end{aligned} \tag{30}$$

$$\begin{aligned}R_2 &= \lim_{t\to\infty}R_2(t)=\lim_{s\to 0}sR_2^*(s)\\ &= [\gamma M_1-\gamma\omega_1\hat{g}_\gamma-\gamma M_1\bar{\alpha}\hat{f}_{\omega 2}+(\gamma M_1\bar{\alpha}+\gamma\omega_2\bar{\beta})\hat{g}_\gamma\hat{f}_{\omega 2}]\\ &\times\overline{F_1^*}(\omega_1)+[(\omega_2\gamma\hat{g}_\gamma-\omega_2\gamma\bar{\beta}\hat{g}_\gamma\hat{f}_{\omega 1})\overline{F_2^*}(\omega_2)]/D\end{aligned} \tag{31}$$

where

$$\begin{aligned}D &= D'(0)=M_1(\omega_1+\gamma)/\omega_1-[M_1(\gamma+\bar{\beta}\omega_1)/\omega_1]\hat{f}_{\omega_1}\\ &-[M_1\bar{\alpha}(\omega_1+\gamma)/\omega_1]\hat{f}_{\omega_2}+[M_1\bar{\alpha}(\gamma+\bar{\beta}\omega_1)/\omega_1]\hat{f}_{\omega_2}\hat{f}_{\omega_1}\\ &-\{M_1(\omega_2-\gamma)/\omega_2+\{[\bar{\beta}(\omega_2^2-\omega_1\omega_2+\gamma\omega_1)-\omega_2\gamma]/\omega_2\}\hat{f}_{\omega_1}\\ &+\{[\gamma\omega_2^2(\beta-\alpha)-\omega_1\gamma\omega_2\bar{\alpha}-\omega_1\omega_2M_1\bar{\alpha}+\gamma\omega_1^2]/\omega_1\omega_2\}\times\hat{f}_{\omega_2}\\ &+\{[\omega_1\omega_2\bar{\alpha}\bar{\beta}M_1+\gamma\bar{\beta}(\omega_2^2-\omega_1^2)+\gamma\omega_2\bar{\alpha}M_1]/\omega_1\omega_2\}\hat{f}_{\omega_2}\hat{f}_{\omega_1}\}\hat{g}_\gamma\\ &+(\beta\gamma\times M_1\hat{f}_{\omega_1}-\beta\gamma M_1\bar{\alpha}\hat{f}_{\omega_2}\hat{f}_{\omega_1})(1/\varphi_1)+(\beta\gamma\omega_2\hat{f}_{\omega_2}-\beta\gamma\omega_1\hat{f}_{\omega_1}\\ &+\beta\gamma M_1\bar{\alpha}\hat{f}_{\omega_2}\hat{f}_{\omega_1})\times\hat{g}_\gamma(1/\varphi_1)\\ &+(\gamma\alpha M_1\hat{f}_{\omega_2}-\gamma\alpha M_1\bar{\beta}\hat{f}_{\omega_2}\hat{f}_{\omega_1})\hat{g}_\gamma(1/\varphi_2)\end{aligned} \tag{32}$$

With the same derivation of $R_1$ and $R_2$, we can get

$$\begin{aligned}L_0 &= (\gamma M_1-\gamma M_1\bar{\alpha}\hat{f}_{\omega_2}-\gamma M_1\bar{\beta}\hat{f}_{\omega_1}\\ &S+\gamma M_1\bar{\alpha}\bar{\beta}\hat{f}_{\omega_2}\hat{f}_{\omega_1})\overline{G_2^*}(\gamma)/D\end{aligned} \tag{33}$$

According to Figure 1, as the channel is in state $(i,k), i=0,1$, it is being sensed. Using state transfer frequency formula shown in [9], we get

$$L_1(t)=\sum_{k=0}^{\infty}\int_0^{\infty}\omega(x)[P_{0k}(t,x)+P_{1k}(t,x)]dx \tag{34}$$

Taking Laplace transform on Equation (34) and using Equations (19)–(25), we obtain

$$\begin{aligned}L_1^*(s) &= \{(M_1M_4g_\gamma-M_1M_4\bar{\beta}f_{\omega_1}g_\gamma)f_{\omega_2}\\ &+\{\gamma M_1-(\gamma M_1+\omega_2M_4)]g_\gamma-\gamma M_1\bar{\alpha}f_{\omega_2}\\ &+(\gamma M_1\bar{\alpha}+\omega_2M_4\bar{\beta})g_\gamma f_{\omega_2}\}\times f_{\omega_1}\\ &+(\omega_2M_4g_\gamma-\omega_2M_4\bar{\beta}g_\gamma f_{\omega_1})f_{\omega_2}\}/D(s)\end{aligned} \tag{35}$$

Then

$$L_1 = \lim_{t\to\infty} L_1(t) = \lim_{s\to 0} sL_1^*(s)$$
$$= [\gamma M_1 \hat{f}_{\omega 1} + \omega_2 \gamma \hat{f}_{\omega_2} \hat{g}_\gamma - \gamma\omega_1 \hat{f}_{\omega 1}\hat{g}_\gamma - \gamma M_1 \bar{\alpha} \hat{f}_{\omega_2} \hat{f}_{\omega 1} \quad (36)$$
$$- \gamma M_1(\alpha - \beta)\hat{f}_{\omega_2}\hat{f}_{\omega 1}\hat{g}_\gamma]/D$$

Similarly, we can derive

$$L_2 = (\beta\gamma M_1 \hat{f}_{\omega 1} - \beta\gamma\omega_1 \hat{f}_{\omega 1}\hat{g}_\gamma - \beta\gamma M_1 \bar{\alpha} \hat{f}_{\omega_2}\hat{f}_{\omega 1}$$
$$+ \beta\omega_2\gamma \hat{f}_{\omega_2}\hat{g}_\gamma + \beta\gamma M_1 \bar{\alpha}\hat{f}_{\omega_2}\hat{f}_{\omega 1}\hat{g}_\gamma)/D \quad (37)$$

$$L_3 = (\alpha\gamma M_1 \hat{f}_{\omega_2}\hat{g}_\gamma - \alpha\gamma M_1 \bar{\beta}\hat{f}_{\omega_2}\hat{f}_{\omega 1}\hat{g}_\gamma)/D \quad (38)$$

Substituting $R_i(i=1,2)$, $L_0$ and $L_j(j=1,2,3)$ by (30), (31), (33), (36), (37) and (38) into (4), respectively, the optimal sensing period $\bar{x}'$ is determined by $F_i(t)$, i.e., the distributions of sojourn time of normal busy and available idle. For the sojourn time of normal busy is used for transmission of PUs themselves and that of available idle can be utilized by CRs' transmission when PUs have no data to transmit, $F_i(t)$ are usually determined by the traffic generated by PUs services in practical networks.

## 5. The Applications of Sensing Period Optimization

This section illustrates the applications of the analytical model developed here by using examples involving PU systems with both voice and data traffic.

### 5.1. Optimization for PUs with Voice Traffic

A typical phone conversation is marked by periods of active talking/talk spurts (or ON periods) interleaved by silence/ listening periods (or OFF periods). The duration of each period is exponentially distributed, i.e., the sojourn time of normal busy and available idle follow exponential distributions with probability density functions $f_i(t) = \lambda_i e^{-\lambda_i t}$, and means $\lambda_i^{-1}$ are constants.

It is a special case for analytical model mentioned above in which $\omega_i^{-1}$s can be substitute by $\lambda_i^{-1}$z. The transient rate of available idle to normal busy and that of normal busy to delay idle are thus reduced to constants $\lambda_i$.

### 5.2. Optimization for PUs with Data Traffic

In the past, exponential distributions are also frequently employed to model interarrival times of data calls for its simplicity, but exponential distributions may not be appropriate in modeling data traffic. Taking Email, an important application that constitutes a high percentage of internet traffic, as an example, its traffic can also be characterized by ON/OFF states. During the ON-state an email could be transmitted or received, and during the OFF-state a client is writing or reading an email. According to traffic models included in the UMTS Forum 3G traffic and ITU RM.2072, the Pareto distribution, which is one of popular heavy-tailed distributions, can be used to close capture the nature of Email traffic for both ON and OFF state, i.e., the sojourn time of normal busy and available idle follow Pareto distributions with probability density functions $f_i(t) = \frac{\alpha_i \beta_i^{\alpha_i}}{t^{\alpha_i+1}}, t > \alpha_i$, and means $\alpha_i\beta_i/(\alpha_i - 1)$, where $\alpha_i > 0$ is called the shape parameter and $\beta_i > 0$ is called the scale parameter.

In order to derive performance measures to search the optimal sensing period, $\frac{\alpha_i \beta_i^{\alpha_i}}{t^{\alpha_i+1}}$ and means $\alpha_i\beta_i/(\alpha_i - 1)$ should substitute $f_i(t)$ and $\omega_i^{-1}$ in Equations (5)–(18).

## 6. Numerical Results and Discussions

Through numerical experiments, we examine the impact of sensing period selection on maximum expected rewards of the channel for different traffic types under various channel parameters in this section.

### 6.1. Performance Analysis

According to the characteristics of channels, we first set the channel parameters as the following $\gamma = 0.02$, $\varphi_1 = 0.033$, $\varphi_2 = 0.1, \alpha = 0.2, \beta = 0.1$, $e_1 = e_2 = 0.1$, $c_0 = c_2 = 0.05$, $c_1 = 0.2$, $c_3 = 0.5$, and let us assume that $T_0$ follows the uniform distribution with parameters $\bar{y} - 10$ and $\bar{y} + 10$, written $T_0 \sim U(\bar{y} - 10, \bar{y} + 10)$, and $\bar{y} = 10$.

Two traffic types of PUs are considered, i.e., Type I: voice traffic and Type II: data traffic. In the case of voice traffic, the sojourn time of normal busy and available idle follow exponential distributions with $\lambda_1 = 0.04$, $\lambda_2 = 0.01$; for data traffic, the sojourn time of normal busy and available idle follow Pareto distributions with $\alpha_1 = 0.8, \beta_1 = 5$ and $\alpha_2 = 2.5, \beta_2 = 60$, i.e., means are $\omega_1 = 0.04$ and $\omega_2 = 0.01$, respectively. Here $\omega_1 = \lambda_1$ and $\omega_2 = \lambda_2$ are selected for easy comparison.

To validate the feasibility of the proposed analytical model for sensing period optimization, numerical examples are carried out for the following three sensing schemes, i.e.,

1) Scheme 1: $T = \bar{x}$, that is, it is exactly a fixed period sensing scheme that sensing is performed at once where $T = \bar{x}$.

2) Scheme 2: $T$ follows the uniform distribution with parameters $\bar{x} - 1$ and $\bar{x} + 1$, written $T \sim U = (\bar{x} - 1, \bar{x} + 1)$.

3) Scheme 3: $T$ follows the exponential distribution with parameter $\bar{x}$, written $T \sim EP(\bar{x})$.

With MATLAB, from (4) we can obtain the optimal sensing periods $\bar{x}'$ and the maximum expected rewards $\boldsymbol{R}$ of the channel per unit time in the steady state for each sensing scheme. Figure 2 and Figure 3 illustrate the expected rewards of the channel for various values of $\bar{x}$ under Type I and Type II traffic types respectively.

For Type I voice traffic, Figure 2 shows that the maximum expected rewards of the channel is varied according to the distribution of sensing period *T*. The optimal sensing periods for each scheme are $\bar{x}' = \{6.5, 13.2, 7.8\}$ and the maximum expected rewards are $R = \{109.8, 106.1, 114.9\}$. So the optimal sensing scheme is sensing randomly with sensing period *T* following the exponential distribution when $\bar{x} = 7.8$ under the above-mentioned channel parameters, for the scheme obtained $R = 114.9$ is maximum compared to other two schemes.

For Type II data traffic, Figure 3 shows that the optimal sensing periods for each scheme are $\bar{x}' = \{10.7, 17.9, 11.3\}$ and the maximum expected rewards are $R = \{77.7, 113.7, 90.2\}$. From Figure 3, it can easily find that the optimal sensing scheme is sensing randomly with sensing period *T* following the uniform distribution. One point that deserves mention is that, compared to the voice traffic, there are significant differences in maximum expected rewards among the three schemes in the case of data traffic.

## 6.2. Analysis of Channel Parameters

In practical, the optimal sensing scheme is different with different sets of channel parameters. Next, we study effects of the setting of various channel parameters on the optimal sensing scheme.

1) *Effects of Distribution Parameters*: in both voice traffic and data traffic, the distribution parameters $\lambda_1, \lambda_2, \gamma, \varphi_1, \varphi_2$ and $\omega_1, \omega_2, \gamma, \varphi_1, \varphi_2$ directly reflect the sojourn time of normal busy, available idle, delay idle, false alarm and miss detection state, respectively. The distribution parameters $\omega_1 / \lambda_1 = 0.04$, $\omega_2 / \lambda_2 = 0.01$ ($\gamma = 0.02, \varphi_1 = 0.033, \varphi_2 = 0.1$) show that the sojourn time of available idle is longest, which means low or moderate PU traffics are chosen. Other three distribution parameters are chosen for 3 sensing schemes, and the optimal rewards *R* vs. the optimal mean sensing period $\bar{x}'$ are shown in Table 1, respectively. a) ($\omega_1 / \lambda_1, \omega_2 / \lambda_2, \gamma, \varphi_1, \varphi_2$) is (0.01, 0.04, 0.02, 0.033, 0.1). The sojourn time of normal busy is relatively long and less opportunities can be used by transmission of CRs; b) ($\omega_1 / \lambda_1, \omega_2 / \lambda_2, \gamma, \varphi_1, \varphi_2$) is (0.04, 0.01, 0.05, 0.033, 0.1). The sojourn time of delay idle is reduced so that more opportunities can be used by CRs than 1); and c) ($\omega_1 / \lambda_1, \omega_2 / \lambda_2, \gamma, \varphi_1, \varphi_2$) is (0.04, 0.01, 0.02, 0.1, 0.033). The sojourn time of miss detec-

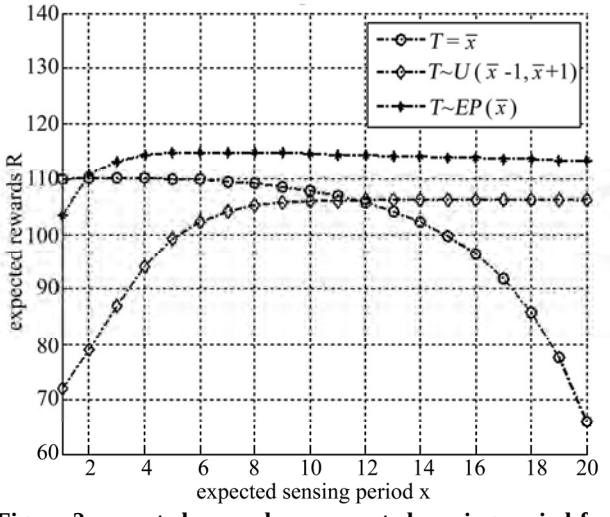



**Figure 2. expected rewards vs. expected sensing period for voice traffic.**

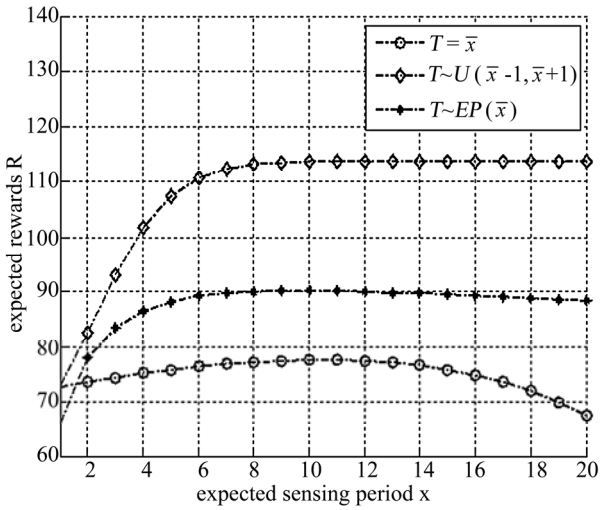



**Figure 3. expected rewards vs. expected sensing period for data traffic.**

**Table 1. The Optimal Rewards *R* Vs. *O*ptimal Mean Sensing Period $\bar{x}'$ for Various Distribution Parameters under $e_1 = e_2 = 0.1$, $c_0 = c_2 = 0.05$, $c_1 = 0.2, c_3 = 0.5$, $\alpha = 0.2, \beta = 0.1$, $T_0 \sim U(\bar{y} - 10, \bar{y} + 10)$ and $\bar{y} = 10$.**

| $\{\omega_1 / \lambda_1, \omega_2 / \lambda_2, \gamma, \varphi_1, \varphi_2\}$ | $T$ | Type I | | Type II | |
|---|---|---|---|---|---|
| | | $\bar{x}'$ | $R$ | $\bar{x}'$ | $R$ |
| (0.01,0.04,0.02,0.033,0.1) | $T = \bar{x}$ | 7.8 | 108.1 | 6.5 | 121.3 |
| | $T \sim U = (\bar{x} - 1, \bar{x} + 1)$ | 10.2 | 106.3 | 9.2 | 86.5 |
| | $T \sim EP(\bar{x})$ | 11.8 | 105.5 | 7.4 | 94.6 |
| (0.04,0.01,0.05,0.033,0.1) | $T = \bar{x}$ | 9.4 | 110.5 | 10.7 | 79.8 |
| | $T \sim U = (\bar{x} - 1, \bar{x} + 1)$ | 11.9 | 109.8 | 8.8 | 99.6 |
| | $T \sim EP(\bar{x})$ | 10.2 | 112.3 | 5.9 | 128.9 |
| (0.04,0.01,0.02,0.1,0.033) | $T = \bar{x}$ | 10.5 | 97.5 | 15.1 | 108.2 |
| | $T \sim U = (\bar{x} - 1, \bar{x} + 1)$ | 9.3 | 99.3 | 17.5 | 132.4 |
| | $T \sim EP(\bar{x})$ | 8.9 | 101.6 | 3.9 | 86.7 |

tion is longer than that of false alarm. From Table 1, we clearly see that the optimal sensing scheme is different with regard to different distribution parameters.

2) *Effects of Weighting Factors*: Weighting factors $e_1$, $e_2$, $c_0$, $c_1$, $c_2$ and $c_3$ can be treated as indexes regarding the importance of $R_1, R_2, L_0, L_1$ and $L_2$. Generally speaking, the PU applications, such as GPS, can only endure minor interference for acceptable Quality of Service (QOS). For such applications, the cost of interference induced by miss detection is more important and $c_3$ is obviously bigger than all other factors as shown in above-chosen setting $e_1 = e_2 = 0.1,\ c_0 = c_2 = 0.05,\ c_1 = 0.2,\ (c_3 = 0.5)$. Table 2 shows optimal rewards $R$ and the optimal mean sensing period $\bar{x}'$ for two different network applications : a) In the case where the PU is not sensitive to interference, the weighting factors $c_0$ and $c_2$ are both larger than other factors in order to maximize the utilization of existing opportunities. Here ($e_1$, $e_2$, $c_0$, $c_1$, $c_2$, $c_3$) is set to (0.1, 0.1, 0.35, 0.05, 0.35, 0.05). b) For energy-constrained CR networks such as sensors and mobile ad hoc applications, frequent sensing is undesirable for high energy overhead, so the weighting factor $c_1$ should be set relatively large and (0.1, 0.1, 0.05,0.5,0.05,0.1) is chosen. This table demonstrates that the optimal sensing scheme is also different with regard to different weighting factors and from the Table 2 one can easily find which scheme performs best.

### 6.3. Performance Comparison for Different Traffic Types

According to the proposed analytical model for sensing period optimization, a sensing scheme with a bigger expected reward performs better. Figure 2, Figure 3, Table 1 and Table 2 show that the optimal sensing scheme is different for different traffic types. In Figure 4, we show the expected rewards for different schemes under $\gamma = 0.02,\ \varphi_1 = 0.033,\ \varphi_2 = 0.1,\ \beta = 0.1, e_1 = e_2 = 0.1,$ $c_0 = c_2 = 0.05, c_1 = 0.2, c_3 = 0.5$, and two different situations for voice traffic: 1) fixed $\lambda_2$ (=0.15) and various $\lambda_1$; and 2) fixed $\lambda_1$ (= 0.2) and various $\lambda_2$, and two different situations for data traffic: 1) fixed $\omega_2$ (=0.15 ) and various $\omega_1$; and 2) fixed $\omega_1$ (=0.2) and various $\omega_2$.

**Table 2. The optimal rewards $R$ vs. optimal mean sensing period $\bar{x}'$ for various weighting factors under $\lambda = 0.04$, $\mu = 0.01,\ \gamma = 0.02, \varphi_1 = 0.033, \varphi_2 = 0.1,\ \alpha = 0.2, \beta = 0.1$ $T_0 \sim U(\bar{y} - 10, \bar{y} + 10)$, and $\bar{y} = 10$.**

| ($e_1$, $e_2$, $c_0$, $c_1$, $c_2$, $c_3$) | $T$ | Type I | | Type II | |
|---|---|---|---|---|---|
| | | $\bar{x}'$ | $R$ | $\bar{x}'$ | $R$ |
| (0.1,0.1, 0.35,0.0 5,0.35,0. 05) | $T = \bar{x}$ | 10.4 | 107.6 | 9.7 | 73.9 |
| | $T \sim U = (\bar{x} - 1, \bar{x} + 1)$ | 11.3 | 111.4 | 14.5 | 98.2 |
| | $T \sim EP(\bar{x})$ | 8.9 | 105.3 | 10.3 | 114.6 |
| (0.1,0.1, 0.05,0.5, 0.05,0.1) | $T = \bar{x}$ | 7.5 | 98.8 | 11.8 | 89.2 |
| | $T \sim U = (\bar{x} - 1, \bar{x} + 1)$ | 15.2 | 95.6 | 6.2 | 125.5 |
| | $T \sim EP(\bar{x})$ | 12.6 | 101.2 | 4.9 | 75.4 |

As shown in Figure 4(a) and Figure 4(b), varying $\lambda_1$ ($\lambda_2$) among 0.1 and 1, we observed that the performance of voice traffic is not sensitive to which sensing scheme is preferred. However, for data traffic, with the variation of $\omega_1$ and $\omega_2$, the optimal sensing scheme is changing among three sensing schemes and the maximum expected rewards have big different, as shown in Figure 4(c) and Figure 4(d). The possible reason is that the optimal sensing period for voice traffic is depending only on a constant mean value of the transmission time for primary users and cognitive radios whereas performance of data traffic are affected by not only mean value but also the distributions of the time of transmission for primary users and cognitive radios.

From the above discussions, we suggest that: 1) for voice traffic, the maximum expected rewards of three sensing scheme is close to each other. In real system, a fixed period sensing scheme is preferred for simplicity; and 2) however, the situation changes dramatically for data traffic, the maximum expected rewards of three sensing scheme are far different. The analytical model proposed can be used to search the optimal sensing scheme, and the analysis can be easily extended to that of any other distribution of sensing period $T$.

## 7. Conclusions and Future Work

This paper dealt with the random sensing period scheme in CR networks. An efficient generalized Markov analytical model for sensing period optimization was proposed and studied. How our proposed analytical model can be applied to PU systems with both voice and data traffic was also discussed. Both numerical results and analysis for various channel parameters and traffic types of PUs were obtained and compared. We found that sensing period does affect the maximum expected rewards of the channel, and the proposed analytical model is valid for the analysis of the case where the sensing period, the transmission time for primary users and cognitive radios are all following arbitrary distributions.

In this work, we assume that the sensing period for each channel is different, i.e. the sensing period is asynchronous for all channels. The proposed scheme is suitable for the scenario where each CR only sense the channel for its operating. If each CR is responsible for sensing more than one channel, the intelligence schedule algorithm of sensing period should be used to negotiate among CRs because the quiet period of each channel is also asynchronous. In future, we would like to develop practical schedule mechanisms or protocols, which deal

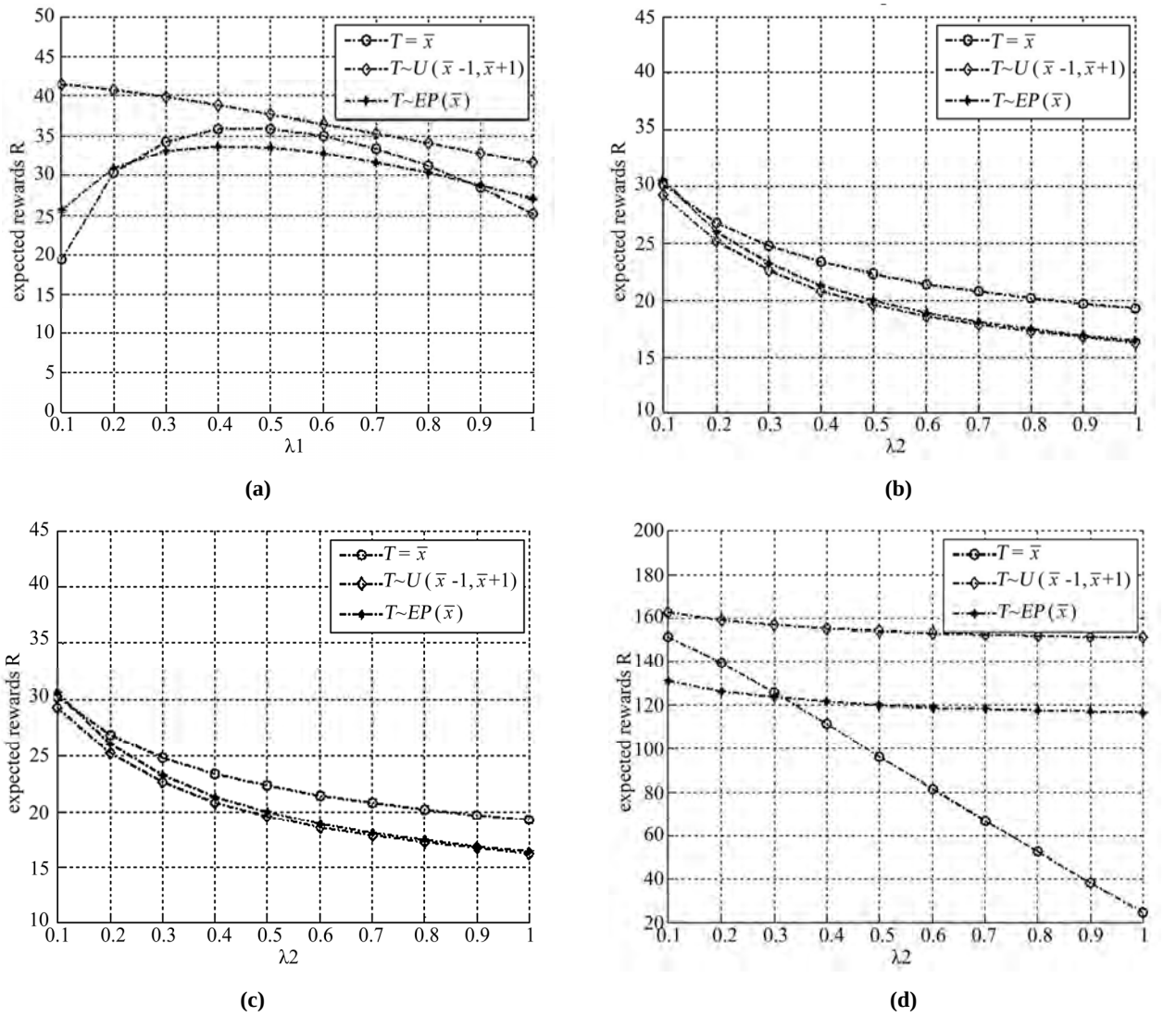



**Figure 4. Performance comparison of different traffic types under** $\gamma = 0.02$, $\varphi_1 = 0.033$, $\varphi_2 = 0.1$, $\alpha = 0.2$, $\beta = 0.1$, $e_1 = e_2 = 0.1$, $c_0 = c_2 = 0.05, c_1 = 0.2, c_3 = 0.5$ **with (a)** $\lambda_2$ **=0.15, (b)** $\lambda_1$ **= 0.2, (c)** $\omega_2$ **= 0.15, and (d)** $\omega_1$ **= 0.2.**

with the coordination of channel sensing among CRs, to make the proposed sensing scheme more effective in real CR networks.

## 8. Acknowledgements

This work was supported by Natural Science Foundation of China under Grant No. 60772110.

## 9. References

[1] Y. C. Liang, W. S. Leon, Y. H. Zeng, *et al*., "System description and operation principles for IEEE 802.22 WRANs," IEEE 802.22–05/0094r4, http://www.ieee802.org/22/.

[2] V. K. N. Lau , R. S. Cheng, R. D. Murch, *et al*., "Adaptive quiet period control," IEEE 802.22–06/0082r0, http://www.ieee802.org/22/.

[3] W. D. Hu, D. Willkomm, M. Abusubaih, *et al*., "Cognitive Radios for dynamic spectrum access-dynamic frequency hopping communities for efficient IEEE 802.22 operation," IEEE Communications Magazine, Vol. 45, No. 5, pp. 88–87, 2007.

[4] T. Shu, S. G. Cui, and M. Krunz, "Medium access control for Multi-Channel parallel transmission in Cognitive Radio networks," In proceedings of IEEE GLOBECOM, San Francisco, pp. 1–5, 2006.

[5] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-Layer sensing in Cognitive Radio networks," IEEE Transactions on Mobile Computing, Vol. 7, No. 5, pp. 533–545, 2008.

[6] Y. Zhang, C. Y. Feng, and C. L. Guo, "A flexible sensing period mechanism of spectrum sensing in cognitive radio

Networks," The Journal of China Universities of Posts and Telecommunications, Vol. 31 No. 2, pp. 128–131, 2008.

[7] C. L. Guo, Z. M. Zeng, C. Y. Feng, and Z. Q. Liu, "An asynchronous spectrum sensing period optimization model and adaptive fuzzy adjustment algorithm," Journal of Electronics & Information Technology, Vol. 31, No. 4, pp. 920–924, 2009.

[8] C. Han, J. Wang, and S. Li, "A spectrum exchange mechanism in cognitive radio contexts," In proceedings of IEEE PIMRC, Finland, pp. 1–5, 2006.

[9] D. H. Shi, "A new method for calculating the mean failure numbers of a repairable system during (0, t)," Acta Mathematicac Applicatae Sinica, Vol. 8, No. 1, pp. 101–110, 1985.

*Wireless Sensor Network*, **2009, 1, 407-416**
doi:10.4236/wsn.2009.15049 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# An Energy-Efficient MAC Protocol for Ad Hoc Networks

**Yongsheng SHI, T. Aaron GULLIVER**
*Department of Electrical and Computer Engineering University of Victoria*
*P.O. Box 3055 STN CSC Victoria, B.C. V8W 3P6 CANADA*
*Email*: {*yshi, agullive*}*@ece.uvic.ca*
*Received July* 21, 2009; *revised August* 26, 2009; *accepted August* 28, 2009

## Abstract

A mobile ad hoc network (MANET) is a collection of nodes equipped with wireless communications and a networking capability without central network control. Nodes in a MANET are free to move and organize themselves in an arbitrary fashion. Energy-efficient design is a significant challenge due to the characteristics of MANETs such as distributed control, constantly changing network topology, and mobile users with limited power supply. The IEEE 802.11 MAC protocol includes a power saving mechanism, but it has many limitations. A new energy-efficient MAC protocol (EE-MAC) is proposed in this paper. It is shown that EE-MAC performs better than IEEE 802.11 power saving mode and exceeds IEEE 802.11 with respect to balancing network throughput and energy savings.

**Keywords:** Energy-Efficient, MAC Protocol, IEEE 802.11, Ad Hoc Networks

## 1. Introduction

Energy efficiency is a major challenge in wireless networks. In order to facilitate untethered communication, most wireless network devices are portable and battery-powered and thus operate on an extremely constrained energy budget. However, progress in battery technology shows that only small improvements in battery capacity can be expected in the near future [1]. Furthermore, since recharging or replacing batteries is costly or, under some circumstance, impossible, it is desirable to keep the energy-dissipation level of devices as low as possible.

A mobile ad hoc network is a collection of two or more nodes equipped with wireless communications and networking capabilities without central network control, i.e. an infrastructure-less mobile network. Energy-efficient design in MANETs is more important and challenging than with other wireless networks. First, due to the absence of an infrastructure, mobile nodes in an ad hoc network must act as routers and participate in the process of forwarding packets. Therefore, traffic loads in MANETs are heavier than in other wireless networks with fixed access points or base stations and thus MANETs have more energy consumption. Second, energy-efficient design needs to consider the trade-offs between different network performance criteria. For example, routing protocols usually try to find a shortest path from sources to destinations. It is likely that some nodes will over-serve the network and their energy will be drained quickly, and thus cause the network to be partitioned. Therefore simple solutions that only consider power constraints may cause a severe performance degradation. Third, no centralized control implies that energy-efficient management in MANETs must be done in a distributed and cooperative manner, which is difficult to achieve.

At the wireless interface, energy consumption in idle mode is only slightly less than transmit mode and almost equal to receive mode [2]. Therefore, it is desirable to build a network protocol that maximizes the time the device is in sleep mode (the wireless interface turned off), and also maximizes the number of wireless devices in sleep mode. Many protocols have been proposed to deal with this challenge [3–6].

In this paper, a new energy-efficient MAC protocol, EE-MAC, is proposed. The design is based on the fact that most applications of ad hoc networks are data-driven, which means that the sole purpose of forming an ad hoc network is to collect and disperse data. Hence, keeping all network nodes awake is costly and unnecessary when some nodes do not have traffic to carry. The proposed protocol conserves energy by turning off the radios of specific nodes in the network. The goal is to reduce energy consumption without significantly reducing network performance. EE-MAC is based on IEEE 802.11 and its power saving mode, and can provide useful information to the network layer for route discovery.

The rest of this paper is organized as follows. Section 2 introduces related work and gives an overview of current energy-efficient protocols for MANETs. Section 3 introduces IEEE 802.11 power saving mode (PSM). Sec-

tion 4 describes the proposed protocol, EE-MAC. In Section 5, performance results are given and EE-MAC is compared to 802.11 and 802.11 PSM. Finally, some conclusions are given in Section 6.

## 2. Related Work

Energy-efficient protocol design is a cross-layer issue and usually spans the network layer and MAC layer. These two layers have different approaches to dealing with power management. At the network layer, energy-efficient routing is a very active research topic. The aim is to choose routes for unicast sessions so as to maximize the overall network lifetime. Essentially, the design principle of energy-efficient routing is to equally balance energy expenditure among network nodes rather than directly reduce power consumption at each node. On the other hand, the MAC layer approach is to turn off the device network interface when it does not have any traffic. Thus, a design combining routing and MAC considerations is appropriate for energy-efficient protocols. We discuss some of the proposed solutions in the remainder of this section.

Local energy-aware routing (LEAR) [4] is an energy-efficient routing protocol that does not consider the MAC layer, while the dynamic power saving mechanism (DPSM) [3] and the on-demand power management [5] protocols are MAC layer approaches. Geographic adaptive fidelity (GAF) [6] is a cross-layer design, but it needs geographic position devices to provide location information.

LEAR is based on the dynamic source routing (DSR) protocol, where route discovery requires flooding of route-request messages. The basic idea of LEAR is to consider the willingness of each mobile node to participate in the routing and forwarding of data packets on behalf of others. This is based on the local information of a mobile node. When a routing path is being established, each mobile node relies on information on remaining battery power to decide whether or not to participate in the selection process of a route path. When a node's remaining battery power is higher than a certain threshold, route-request messages are forwarded and the node joins in the route path selection process; otherwise, the message is discarded. Thus, all intermediate nodes along the route path have sufficient power and the first arriving route message is considered to have followed an energy-efficient as well as a reasonably short path. If any of the intermediate nodes drop the route-request message, which means no nodes are willing to join the route path, the source will not receive a single reply even though a route may exist. To prevent this, the source node will resend the same route request message with a lower threshold.

Observing that the fixed beacon interval in IEEE 802.11 PSM wastes energy, DPSM uses adaptively changed ad hoc traffic indication messages (ATIMs). Coupled with a separate DATA window, DPSM can control the transition to the low-power state in the middle of a beacon interval. Therefore, a node is allowed to enter sleep mode after completing any transmissions that are explicitly announced in the ATIM window, and a longer sleep mode time is achieved.

On-demand power management for ad hoc networks bases power management decisions on traffic in the network. The key idea is that transitions from power-saving mode to active mode are triggered by communication events instead of the established beacon interval used in IEEE 802.11 PSM. On the other hand, transitions from active mode to power-saving mode are determined by a soft-state timer which is refreshed by the same communication events that trigger a transition to active mode. A node uses HELLO messages to track its neighbor's power management state to decide whether or not to send packets to them.

The GAF protocol identifies redundant nodes with respect to routing and turns them off without sacrificing routing fidelity. Each node uses location information based on GPS to associate itself with a virtual grid, where nodes in a particular grid square are redundant with respect to forwarding packets. One master node in each grid stays awake to route packets. With GAF, nodes can be in three states, sleep, discover or active. Initially a node is in the discover state and exchanges discovery messages including grid IDs to find other nodes within the same grid. A node becomes a master if it does not hear any discovery messages for a given period of time. If more than one node can become a master, the one with the longest expected lifetime becomes the master and handles the routing for that grid square.

## 3. An Overview of IEEE 802.11 Power Saving Mode

Power management can achieve great savings in infrastructure networks. All traffic for mobile stations must go through access points, so they are ideal locations to buffer traffic. However, in ad hoc networks, far more of the burden is placed on the sender to ensure that the receiver is active or awake. Receivers must also be more available and cannot sleep for as long as in infrastructure networks.

Power management in IEEE 802.11 power saving mode (PSM) is based on traffic indication messages. Nodes use ATIMs to notify other nodes to prepare to receive data. All nodes have to wake up periodically to listen for ATIMs and check whether they have packets to receive.

In PSM [7,8], time is divided into beacon intervals and each beacon interval starts with an ATIM window. This

window is the period during which nodes must remain active and no stations are permitted to power down their wireless interface. The ATIM window size is a parameter that can be adjusted. Setting it to 0 means no power management is used. There are four possibilities for a node in terms of ATIMs: the node has transmitted an ATIM, received an ATIM, neither transmitted nor received, or both transmitted and received. Nodes that transmit ATIM frames do not sleep because this indicates an intent to transmit buffered traffic. Nodes to which an ATIM is addressed must also keep awake so they can receive data packets from the ATIM sender. A node that both transmits and receives of course needs to be active. Thus, only those nodes that neither transmit nor receive an ATIM can go to sleep after the ATIM window. Figure 1 illustrates the basic PSM operations. Nodes *A* and *B* have advertised packets in the ATIM window by sending ATIMs and receiving ATIM-ACKs, both of which are subject to the DCF rules described earlier. Therefore nodes *A* and *B* remain awake for the rest of the beacon interval. The transmission of data packets from nodes *A* and *B* takes place during the beacon interval. The node that has no packets to transmit can go into sleep mode at the end of the ATIM window if it does not receive an ATIM during the window. In Figure 1, node *C* enters sleep mode after the ATIM window, thus saving energy. All sleeping nodes wake up again at the start of the next beacon interval.

The beacon and ATIM window sizes can affect the performance of PSM. Since no data packets are transmitted in the ATIM window, overhead in terms of energy consumption and bandwidth is incurred. If we use a small ATIM window to improve energy savings, there may not be enough time to advertise all buffered data packets. Conversely, using a large ATIM window may unnecessarily waste bandwidth and not leave enough time to transmit buffered data. Moreover, PSM also suffers from long packet delivery latency: for each hop that a packet traverses, the packet is expected to be delayed for at least a beacon period. PSM was originally designed for single-hop networks, which means all nodes in the network are fully connected. However, ad hoc networks are usually multi-hop networks, and thus PSM is not an ideal solution.

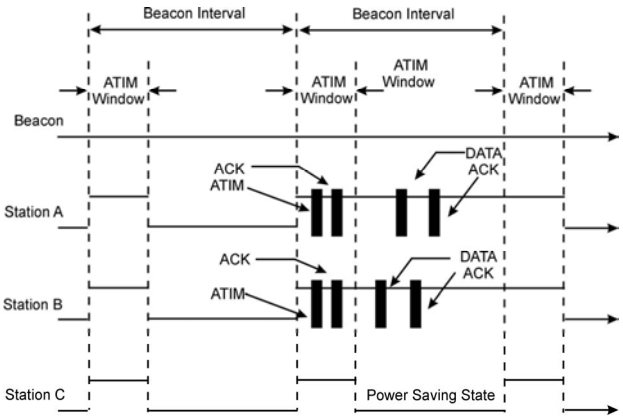



**Figure 1. IEEE 802.11 PSM operation.**

## 4. The Proposed EE-MAC Protocol

The key idea of EE-MAC is to elect master nodes from all nodes in the network. Master nodes stay awake all the time and act as a virtual backbone to route packets in the ad hoc network. Other nodes, called slave nodes, remain in an energy-efficient mode and wake up periodically to check whether they have packets to receive. To be fair, a rotation mechanism between masters and slaves is used. EE-MAC uses some features of PSM, such as periodically waking up at the beginning of the beacon interval.

EE-MAC can provide knowledge and guidance to the route lookup process, because only master nodes can be selected along a routing path. On the other hand, EE-MAC requires a mechanism to awaken a sleeping node when packet delivery is imminent. This is usually handled by low-level mechanisms at the MAC or physical layers. In EE-MAC, if a node has been asleep for a while, packets addressed to it are not lost but are stored at one of its upstream nodes, usually a master. When the node awakens, the buffered data is sent to it (this is a PSM feature which is used in our protocol).

### 4.1. Design Criteria

We consider the following design criteria.

- The protocol must ensure enough master nodes are elected to build the backbone of the network so that every node has at least one master in its vicinity. A collection of masters can be described as a connected dominating set (CDS). All nodes are either a member of the CDS or a direct neighbor of at least one of the members of the CDS. Nodes in the CDS serve as the routing backbone and remain active all the time. All other nodes are slave nodes and can choose to sleep. Since slave nodes do not join in the process of route discovery or packet forwarding, network connectivity is decreased. To prevent a dramatic decrease in throughput, an acceptable set of masters is required to maintain global connectivity with some redundancy.
- The master node election algorithm is based on local information, which is a distributed approach. Each node only employs local information to determine whether it will become a master. Due to the characteristics of distributed management in ad hoc networks and the two essential requirements, low overhead and fast convergence, the algorithm for finding a CDS should be localized. The election algorithm is given in the next section.

• The algorithm must have a fair way to rotate masters and slaves in order to ensure that nodes equally share the job of providing global connectivity. Over-using some critical nodes will severely decrease the network lifetime. Thus, if alternative nodes appear, masters can step down and give the new nodes a chance to serve as masters to balance node energy consumption.

### 4.2. Master Election and Forming a Connected Dominating Set

To form a CDS, many researchers have proposed solutions [9–11]. In this paper, we use the algorithm in [12] modified for the energy saving condition.

Given a simple graph $G=(V,E)$, where $V$ is a set of nodes and $E$ is a set of links, a link from $u$ to $v$ is denoted by a pair ($u$, $v$). According to [12], a set $V'\subset V$ is a dominating set of $G$ if every node $v\in V-V'$ is connected by at least one node $u\in V'$. For example, in Figure 2, the node sets $u$, $v$ in *a* and $u$, $v$ in *b* are dominating sets of the corresponding graphs. If all nodes in a dominating set are connected together, it forms a CDS.

To quickly elect masters in an ad hoc network, we use the following steps:

1) Initially assign the marker $F$ to each node $u$ in $V$.

2) Each node $u$ exchanges its neighbor set $N(u)$ with all its neighbors.

3) $u$ changes its marker to $T$ if there exist nodes $v$ and $w$ such that $(w,u)\in E$ and $(u,v)\in E$, but $(w,v)\notin E$.

The $T$-marked nodes form a connected dominating set and become masters, while the $F$-marked nodes become slaves. However, we may not need all $T$-marked nodes elected to act as the backbone of the network because there are redundancies in this set. We say a node is covered if its neighbors can reach each other directly or via other connected $T$-marked nodes. We establish a rule to reduce the number of masters based on the idea that if a node is covered by no more than $k$ connected $T$-marked nodes, we can change the marker of this node to $F$. In general, assuming that $V_k^{'}=\{v_1,v_2,...,v_k\}$ is the node set of a connected subgraph in $G'$ and if $N(u)\subseteq N(V_k^{'})$ in $G$, then $u$ can change its marker from $T$ to $F$. This rule can be simply described as: if every pair of neighbors of a $T$-marked node can be connected directly or via no more than $k$ other connected $T$-marked nodes, this node is marked as $F$.

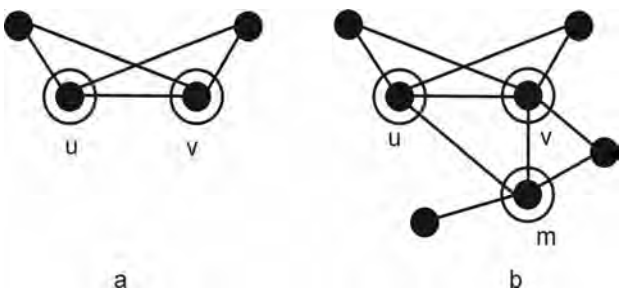



**Figure 2. Examples of connected dominating sets.**

Two more issues need to be considered, node connectivity and node energy. We denote the connectivity level of a node $i$ as $CL_i$. Let $N_i$ be the number of neighbors of node $i$ and $C_i$ be the number of pairs of nodes among these neighbors that can be connected via $i$ if $i$ becomes a $T$-marked node. Clearly, $0\le C_i\le\binom{N_i}{2}$, and define the maximum as $CL_i\le C_i/\binom{N_i}{2}$. The energy level of node $i$ can be expressed as $EL_i=E_{ri}/E_i$, where $E_{ri}$ is the remaining node energy and $E_i$ is the initial node energy. Finally, the node id, $id_i$, will be considered if the two factors given above are identical.

Overall, the rule to reduce redundant $T$-marked nodes is as follows:

Assuming $V_k^{'}=\{v_1,v_2,\ldots,v_i\ldots v_k\}$ is the node set of a connected subgraph in $G'$, the marker of $u$ is changed to $F$ if one of the following conditions holds:

1) $N(u)\subseteq N(V_k^{'})$ in $G$, and for any node $v_i\in V_k^{'}$, $N(v_i)\not\subseteq N(V_k^{'}-v_i)\cup N(u)$.

2) $N(u)\subseteq N(V_k^{'})$ in $G$, and for some nodes $v_1,...,v_i\in V_k^{'}$, $N(v_1,...,v_i)\subseteq N(V_k^{'}-v_1,...,v_i)\cup N(u)$

- $EL_u<\min\{EL_1,EL_2,...,EL_i\}$ or
- $CL_u<\min\{CL_1,CL_2,...,CL_i\}$ if $EL_u=\min\{EL_1,EL_2,...,EL_i\}$ or
- $id_u<\min\{id_1,id_2,...,id_i\}$ if $EL_u=\min\{EL_1,EL_2,...,EL_i\}$ and $CL_u=\min\{CL_1,CL_2,...,CL_i\}$

After connected dominating set selection and reduction, all $T$-marked nodes will become masters and the other nodes will become slaves. We use periodically broadcasted HELLO messages to make each node in the network aware of its neighbors' status, including whether or not they are masters, their current masters and their current neighbors. Using a small value for $k$ will increase network connectivity but there will be many redundant masters which will consume more energy. Conversely, a large value for $k$ will save energy but decrease the robustness of the network. In addition, a large $k$ will usually require more frequent HELLO messages to collect information. To balance the energy savings and network throughput, we use $k=3$ in this paper.

As mentioned above, rotation of masters and slaves is an important design requirement. The rotation of masters and slaves is done to allow every possible node to have a chance to become a master, and let current masters

change their role to save energy. Each master periodically checks if it should withdraw as a master. The conditions to trigger a withdrawal are essentially the same as for CDS reduction given above. However, in order to balance the network load, we force some masters to quit even if the conditions to withdraw are not met. After a node has served as a master for some period of time or if its energy level is below a certain value of $EL_i$ and the average of its neighbors, it will withdraw even if there are no masters nearby. The only exception is if some neighbors can only be connected to the network via that node.

### 4.3. Features of EE-MAC

In EE-MAC, since masters do not operate in power saving mode and can forward packets all the time, the packet delivery ratio and packet delay can be improved greatly compared to PSM. In this section, we present the important features of EE-MAC.

#### 4.3.1. Entering Sleep Mode Earlier

In the original PSM, a node with packets to transmit will send an ATIM frame to the destination, and both source and destination will stay awake in that beacon interval, no matter how many packets need to be transmitted. While this approach has its advantages, it may result in much higher energy consumption than necessary. For example, if a source only has one packet pending, they have to waste the whole beacon period to deal with this packet. To avoid this, we add the number of data packets remaining at the sender to every data packet sent to the destination. This information allows the destination to know when it has received all pending packets for it. When the source or destination have sent or received all their packets, they can enter sleep mode until the beginning of the next beacon interval.

#### 4.3.2. Priority Processing of Packets to Slaves

When nodes are trying to send packets, they first deal with those to be sent to slave nodes. After transmitting all packets to slave nodes, packets between masters can be sent. By using this method, slaves can be in sleep mode as long as possible.

#### 4.3.3. Prolonging the Sleep Period for Slaves

In EE-MAC, most packets are forwarded by masters and packet routing via slaves is kept to a minimum. To take advantage of this, each slave uses history information to decide their sleep time. When a node observes two consecutive beacon intervals without any packets addressed to it, it will decide to sleep through the next beacon interval. The corresponding master must store this information since failure to get an ACK does not guarantee a broken link. If the master does not know a slave's situation, it just buffers the packets to that slave. Only when the master does not hear from a neighboring slave for two consecutive beacon intervals does it discard these packets.

#### 4.3.4. Additional MAC Layer Control

Nodes in an ad hoc network may move randomly. Thus, to quickly adapt to network topology changes, a node informs its neighbors of its status, master or slave, by using the power management bit in the MAC header. Since the MAC header can be heard anywhere in the network, including RTS/CTS packets, this information will help neighbors to know each other's situation.

## 5. Performance Results

### 5.1. Simulation Environment

Our conclusions are based on the results gathered by extensive simulation of a network model which implements EE-MAC. For the simulations, we used Network Simulator-2 (NS-2) [13,14]. NS-2 is a popular package which has been widely used in mobile ad hoc network studies. For comparison with EE-MAC, we also implemented IEEE 802.11 and its PSM mode.

We consider 25, 50 and 75 nodes moving in a square area of 500m×500m, 750m×750m and 1000m×1000m based on a mobility model called *random waypoint* [15]. Initially, each node chooses a random position in the area, chooses a random destination, chooses a speed at random uniformly distributed between 0m/s and 10m/s, and moves towards the destination at the chosen speed. The node then pauses for a period of time before repeating the same process. Longer pause times reflect lower node mobility and shorter pause times reflect higher mobility. Simulations were performed for 400 seconds, so a 400 second pause time means no node mobility.

The nodes have 2 Mbps bandwidth and 250m radio range. Each source node generates a Constant-Bit-Rate (CBR) flow to the destination with 256 byte packets. We vary the number of sources and the number of packets sent per second to change the network load. A network load of 10% means that the total bit rate of all traffic sources is 2×10% = 0.2 Mbps. DSR [16] is used as the routing protocol. For the energy model, we use the data shown in Table 1. All performance results shown in this paper are an average of 10 runs.

We use the following metrics to evaluate network performance:

- **Data packet delivery ratio:** The data packet delivery ratio is the ratio of the number of packets generated at

**Table 1. Power Consumption Model [2]**

| Transmit Mode | Receive Mode | Idle Mode | Sleep Mode |
|---|---|---|---|
| 1400mW | 1020mW | 890mW | 70mW |

the sources to the number of packets received by the destinations. This metric reflects the network throughput. One of our goals is to design an energy-efficient MAC protocol which can improve energy consumption without suffering a significant capacity loss. Thus, this metric is useful to measure any degradation in network throughput.

- **End-to-end delay:** This metric not only includes the delays due to data propagation and transfer, but also those caused by buffering, queuing and retransmitting data packets.
- **Energy efficiency:** We define energy efficiency as

$$\text{Energy efficiency} = \frac{\text{Total bits transmitted}}{\text{Total energy consumed}}$$

where the total bits transmitted is calculated using application layer data packets only and total energy consumption is the sum of the energy consumption in the nodes during the simulation time. The unit of energy efficiency is bit/Joule and the greater the number of bits per Joule, the better the energy efficiency achieved.

### 5.2. Performance Evaluation

We now present our simulation results. The figures in this section show three curves labeled 802.11, PSM and EE-MAC. The curves labeled 802.11 correspond to the IEEE 802.11 protocol without using power saving mode. The curves labeled PSM indicate the IEEE 802.11 protocol with power saving mode. The curves labeled EE-MAC represent the protocol proposed in this paper.

#### 5.2.1. Impact of the Network Load

From the simulation results, we observe that network load has a significant impact on all three protocols. However, we show that varying the network load affects these protocols differently in terms of our performance metrics.

In Figures 3 and 4 we show the packet delivery ratio under different network loads from 10% to 40%. When the network load is low (10%), 802.11 performs a little better than EE-MAC while EE-MAC provides a significant improvement over PSM. As the network load increases to 40%, all three protocols become worse due to the higher collision rate. However, the performance differences between 802.11 and EE-MAC, and EE-MAC and PSM also increase, which means heavier traffic has more impact on EE-MAC than 802.11 because under a heavy network load, the master election algorithm operates more frequently to rotate masters and slaves. Among the three protocols, PSM always performs worst. PSM drops significantly more packets than the others because of the existence of a fixed ATIM window, which wastes bandwidth. When the traffic is high, it is possible that the

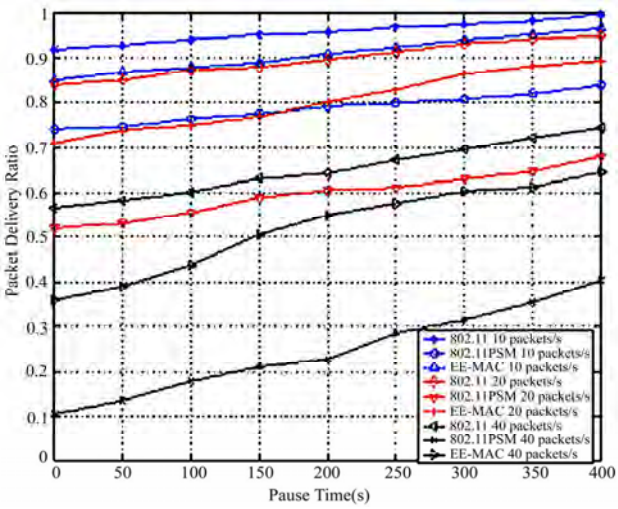



**Figure 3. Packet delivery ratio with 50 nodes and 10 sources in an area of 750m×750m.**

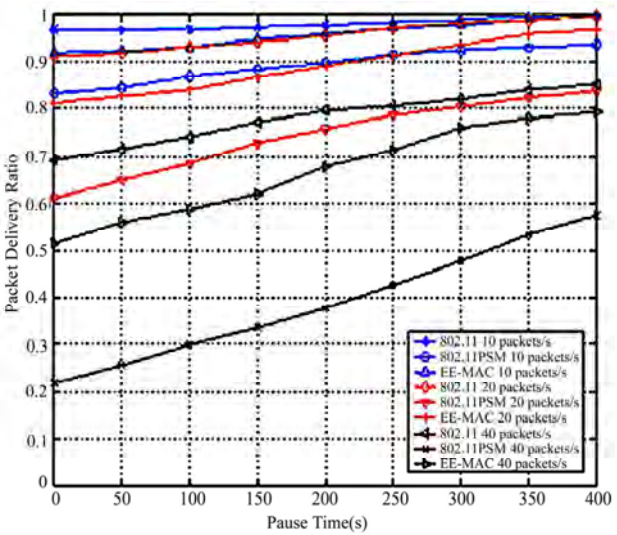



**Figure 4. Packet delivery ratio with 75 nodes and 10 sources in an area of 750m×750m.**

ATIM window is not long enough to advertise all pending packets, or the buffered data packets cannot all be sent out during a beacon interval. On the other hand, EE-MAC has the advantage of masters which never enter sleep mode, so traffic between masters does not need to be advertised. Coupled with the fact that most of the network traffic is data traffic between masters, EE-MAC can use a shorter ATIM window than PSM and thus provide better performance than PSM. EE-MAC is worse than 802.11 because it still uses an ATIM window in every beacon interval which wastes some bandwidth. Moreover, the overhead of the master election algorithm and using fewer nodes to forward packets also decreases the packet delivery ratio.

In Figures 5 and 6 we present the average packet delay. Again, 802.11 performs the best among the three techniques, and as the network load becomes heavier this advantage increases. EE-MAC is not much worse than 802.11, but is far superior to PSM. PSM suffers from long packet delays mainly because of its mechanism of *receiving-buffering-advertising-sending*. Thus, each hop in a PSM network corresponds to the length of the beacon interval. In addition, if the network load is high, some packets have to be buffered up to 3 beacon intervals before being sent out. Note that packets are dropped if they have been kept in the buffer for 3 beacon intervals. These factors cause PSM to have poor packet delay performance. Similarly, the overhead due to master elections, using ATIM windows, and fewer routing nodes, results in EE-MAC having higher packet delays than 802.11.

In Figures 7 and 8, the metric of most interest in this paper, energy efficiency, is presented. The results show that EE-MAC performs best among all protocols. This is because EE-MAC allows slave nodes to enter sleep mode when no packets are addressed to them, but there always exist awake nodes (masters) to forward packets. Furthermore, EE-MAC can tell slaves to enter sleep mode once they have finished receiving all data addressed to them in a beacon interval. These benefits allow EE-MAC to nicely balance energy consumption and packet delivery ratio, resulting in much better energy efficiency. PSM does perform better than 802.11 and is comparable to EE-MAC under light network load conditions.

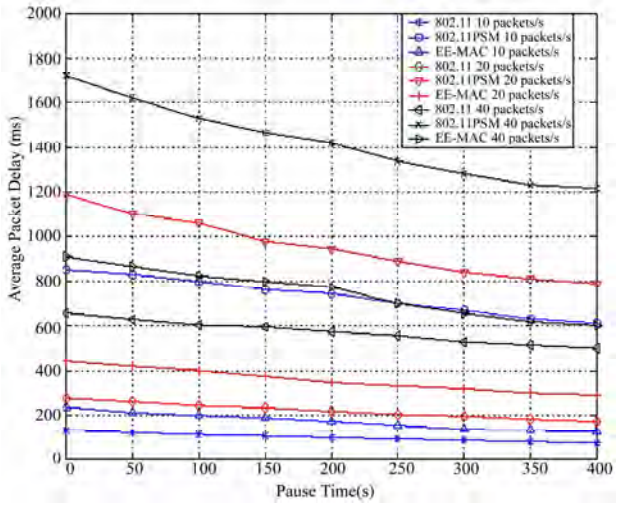



**Figure 5. Average packet delay with 50 nodes and 10 sources in an area of 750m×750m.**

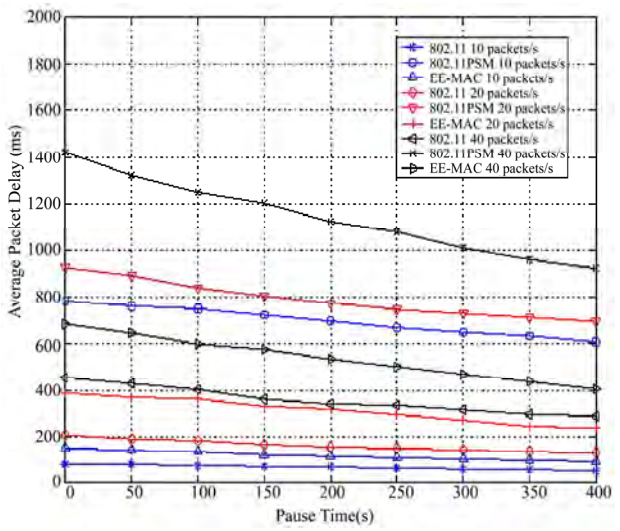



**Figure 6. Average packet delay with 75 nodes and 10 sources in an area of 750m×750m.**

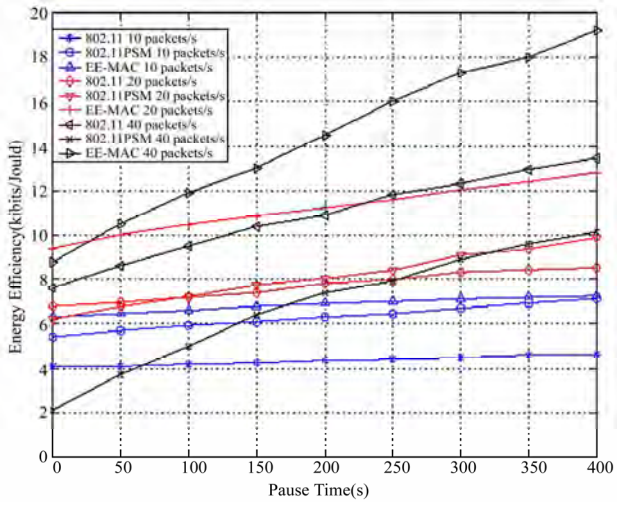



**Figure 7. Energy efficiency with 50 nodes and 10 sources in an area of 750m×750m.**

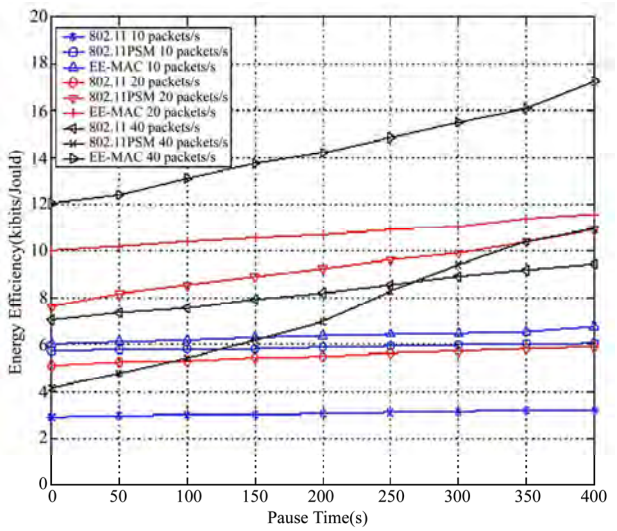



**Figure 8. Energy efficiency with 75 nodes and 10 sources in an area of 750m×750m.**

As the network load increases, PSM becomes worse very quickly due to high data packet loss. Moreover, more nodes need to participate in packet forwarding under a heavy network load, which means more nodes must stay awake all the time, causing high energy consumption.

Comparing energy efficiency between EE-MAC and 802.11 under different network loads is somewhat complicated because it is related to both network throughput and energy consumption. Since the difference in power consumption among transmit, receive and idle modes is not significant, the energy savings achieved is highly dependent on the network node density, the ratio of time in sleep mode to other modes, and the ratio of masters to slaves. EE-MAC gains by reducing the number of awake nodes. In some cases, 802.11 can outperform EE-MAC. In Figure 9, the performance is given for 50 nodes, 20 sources and 20 packets/s, and 75 nodes, 5 sources and 20 packets/s. With 50 nodes, EE-MAC is sometimes worse than 802.11 because at least 20 of the 50 nodes in the network can never enter sleep mode. Conversely, with 75 nodes and 5 sources, EE-MAC is approximately 3 times better than 802.11. Figure 9 also indicates that with 5 sources and 75 nodes, PSM is slightly better than EE-MAC because in this situation, the cost of maintaining a CDS is higher than the advantages it brings. As the network load increases, EE-MAC will improve relative to PSM.

### 5.2.2. Impact of Mobility

From the results shown, it is clear that high mobility decreases the performance of all three protocols. Overall, mobility has a greater impact on EE-MAC than the other two protocols. The reason is that with high mobility, the network topology changes rapidly and links between nodes can break often. Thus, the master election algorithm has to operate frequently, which introduces more overhead than with low mobility. Although mobility impacts EE-MAC in terms of packet delivery ratio, it still performs better than PSM. In terms of energy efficiency, PSM performs very badly because under high mobility, frequent route discovery messages cause a node to stay awake much of the time.

### 5.2.3. Varying Node Density

Clearly, high density can significantly improve network performance with all three protocols. They will have more options to choose a better route, and if a route breaks, it is easier and quicker to find another one. As mentioned above, EE-MAC relies more on node density to enhance its performance than the other protocols because if the number of sources is constant, with high node density, only a small fraction of the nodes need to be elected as masters and most nodes can remain in power saving-mode. This will result in significant energy savings. Furthermore, with high node density, the impact of mobility on EE-MAC is reduced. In other words, the higher the node density, the better EE-MAC performs.

### 5.2.4. Changing Network Area

Reducing the network area from 750m×750m to 500m×500m results in increased packet delivery ratio, decreased average packet delay and increased energy efficiency for all three protocols. In a smaller network area, the advantages of EE-MAC are not as prominent because the weaknesses of the other two protocols are reduced. Simulation results show that in an 500m×500m area, most routes are 2–4 hops long, while in an 750m×750m area, routes are often 4–7 hops long, so the routing overhead and packet delay are much less in small networks. The results show that EE-MAC only provides a slight benefit in energy efficiency and is not as superior to PSM as in a 750m×750m area. The performance with the network area increased to 1000m×1000m was also evaluated. Not only is the node density decreased, but also forming a CDS requires more nodes in general and the CDS can be broken more easily. These factors cause a degradation in performance with EE-MAC, especially in a high mobility network. As node mobility increases, the packet delivery ratio and energy efficiency with EE-MAC is reduced more compared to 802.11 and PSM.

### 5.2.5. Static Network

Figures 10 and 11 show the performance under static network conditions. We fix the number of sources at 10 and vary the CBR to change the network load. Packet delivery ratio and energy efficiency are given corresponding to different network loads. It is clear that as the network load increases, the packet delivery ratio of PSM drops much more quickly than with EE-MAC and 802.11. The decreased difference between EE-MAC and 802.11 with 50 nodes, compared to that with 75 nodes, shows that EE-MAC benefits more from a higher node density.

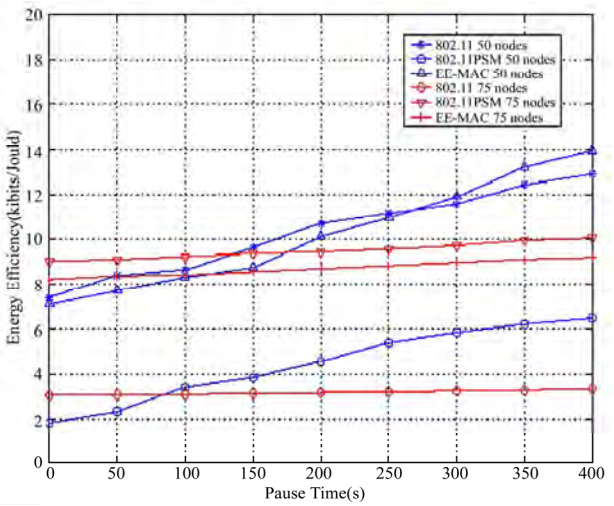



**Figure 9. Energy efficiency with 50 nodes and 20 sources, and 75 nodes and 5 sources, in an area of 750m×750m.**

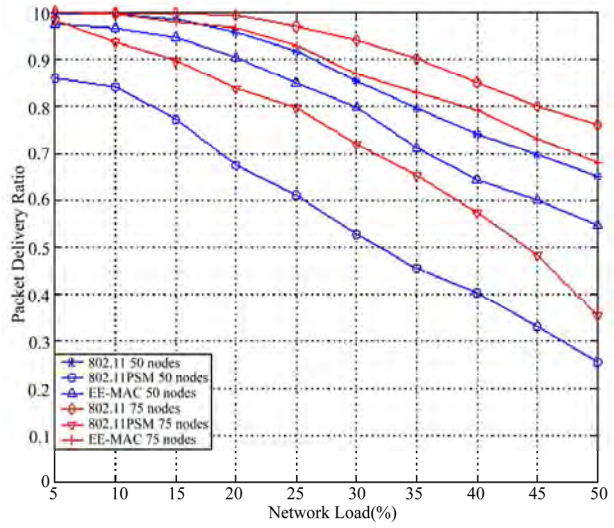



**Figure 10. Packet delivery ratio with 50 and 75 nodes, 10 sources and 5% to 50% network load without mobility in an area of 750m×750m.**

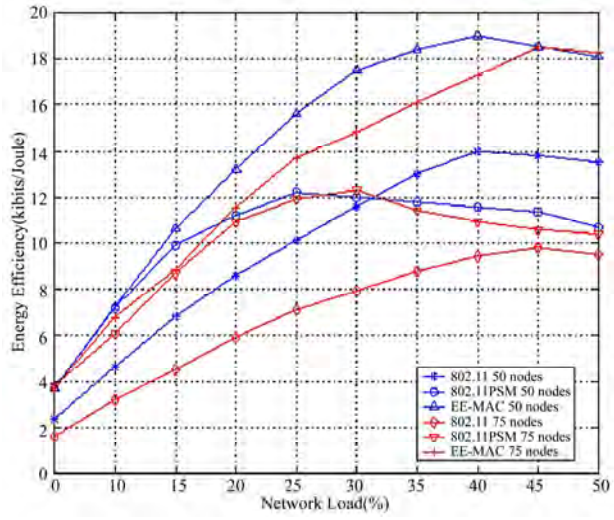



**Figure 11. Energy efficiency with 50 and 75 nodes, 10 sources and 5% to 50% network load without mobility in an area of 750m×750m.**

Note that in terms of energy-efficiency, under certain conditions, PSM performs slightly better than EE-MAC for two reasons. First, PSM has good network throughput under a light network load. Second, EE-MAC needs an almost constant number of nodes to form a CDS even when the network load is very light and thus has constantly awake nodes with little traffic through them.

## 6. Conclusions

This paper presented EE-MAC, an energy-efficient MAC protocol for mobile ad hoc networks. The goal was to reduce energy consumption in an ad hoc network without significantly reducing network performance. The key idea of EE-MAC is to elect some nodes to form a connected dominating set and use this as a virtual backbone to route packets, while other network nodes, called slaves, stay in power-saving mode. EE-MAC is a cross-layer design which spans the network layer and the MAC layer.

The performance of EE-MAC was evaluated using the NS-2 network simulator, and compared to IEEE 802.11 with and without power saving mode. The results show that IEEE 802.11 performs better than EE-MAC in terms of packet delivery ratio and average packet delay. However, EE-MAC exceeds IEEE 802.11 in energy efficiency and is much better than PSM in overall terms. The network load has a great impact on the behavior of EE-MAC. Under a light network load, EE-MAC is only slightly worse than IEEE 802.11, but as the network load increases, the difference in performance between EE-MAC and IEEE 802.11 increases because EE-MAC needs to rotate masters and slaves more frequently with high traffic and EE-MAC still uses the ATIM window. The results also show that the higher the node density, the better EE-MAC performs. In summary, a mid-sized network with relatively high node density is the best environment to utilize EE-MAC.

## 7. References

[1] S. Sheng, A. Chandrakasan, and R. W. Brodersen, "A portable multimedia terminal," IEEE Communications Magazine, Vol. 30, No. 12, pp. 64–75, December 1992.

[2] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," Proceedings of IEEE INFOCOM, pp. 1548–1557, April 2001.

[3] E. S. Jung and N. H. Vaidya, "An energy efficient MAC protocol for wireless LANs," Proceedings of IEEE INFOCOM, pp. 1756–1764, June 2002.

[4] K. Woo, C. Yu, D. Lee, H. Y. Youn, and B. Lee, "Non-blocking, localized routing algorithm for balanced energy consumption in mobile ad hoc networks," Proceedings of International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 117–124, August 2001.

[5] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," Proceedings of IEEE INFOCOM, pp. 481–491, March 2003.

[6] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," Proceedings of ACM International Conference on Mobile Computing and Networking, pp. 70–84, 2001.

[7] Supplement to IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer

(PHY) Specifications: High-speed Physical Layer Extension in 2.4 GHz Band, IEEE Std 802.11b, 2000.

[8] M. S. Gast, 802.11 Wireless Networks: The Definitive Guide, O'Reilly & Associates, 2002.

[9] P. J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," Mobile Networks and Applications, Vol. 9, No. 2, pp. 141–149, April 2004.

[10] Y. P. Chen and A. L. Liestman, "Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks," Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 165–172, June 2002.

[11] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," Proceedings of ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, pp. 7–14, August 1999.

[12] J. Wu and H. Li, "Domination and its applications in ad hoc wireless networks with unidirectional links," Proceedings of IEEE International Conference on Parallel Processing, pp. 189–197, August 2000.

[13] UCB LBNL VINT Group, Network simulator (V. 2), http://www.isi.edu/nsnam/ns.

[14] K. Fall and K. Varadhan, Network Simulator Notes and Documentation, The VINT Project, UC Berkeley, USC/ ISI, LBL and Xerox PARC, 1999, http:// www.isi.edu/nsnam/ns.

[15] J. Broth, D. Maltz, D. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," Proceedings of International Conference on Mobile Computing and Networking, pp. 85–97, October 1998.

[16] D. Johnson, D. Maltz, and Y. C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," IETF draft, July 2004, http://www.ietf.org/ rfc/rfc3684.txt.

*Wireless Sensor Network*, 2009, 1, 417-424
doi:10.4236/wsn.2009.15050 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Voronoi-Based Coverage Optimization for Directional Sensor Networks

**Jing LI[1], Ruchuan WANG[1,2,3], Haiping HUANG[1], Lijuan SUN[1,3]**
[1]*College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China*
[2]*State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Beijing, China*
[3]*Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing, China*
*Email*: *leejnjupt@126.com*, {*wangrc, hhp*}*@njupt.edu.cn*
*Received July* 25, 2009; *revised August* 3, 2009; *accepted August* 4, 2009

## Abstract

Sensing coverage is a fundamental problem in sensors networks. Different from traditional isotropic sensors with sensing disk, directional sensors may have a limited angle of sensing range due to special applications. In this paper, we study the coverage problem in directional sensor networks (DSNs) with the rotatable orientation for each sensor. We propose the optimal coverage in directional sensor networks (OCDSN) problem to cover maximal area while activating as few sensors as possible. Then we prove the OCDSN to be NP-complete and propose the Voronoi-based centralized approximation (VCA) algorithm and the Voronoi-based distributed approximation (VDA) algorithm of the solution to the OCDSN problem. Finally, extensive simulation is executed to demonstrate the performance of the proposed algorithms.

**Keywords:** Directional Sensor Networks, Coverage, Deployment, Voronoi Diagram

## 1. Introduction

In recent years, wireless sensor networks (WSNs) have attracted intense interests due to their wide applications in military and civilian operations, such as environmental monitoring, battlefield surveillance, and health care [1–3]. The conventional research of WSNs is always based on isotropic sensors which can equally detect the environment in each orientation. However, sensors may have a limited angle of sensing range due to special applications, which are denoted as directional sensors. There are many kinds of directional sensors, such as video sensors [4,5], ultrasonic sensors [6] and infrared sensors [2]. The most familiar directional sensors are the video sensors widely used in wireless multimedia sensor networks (WMSNs). Note that the directional characteristic we discuss in this paper is from the point of view of the sensing, but not from the communicating activity of sensors.

Sensing coverage is a fundamental issue in all sensor networks, which has been studied by many works. Due to the large variety of sensors and applications, coverage is subject to a wide range of interpretations. In general, coverage can be considered as the measure of quality of service (QoS) in a sensor network. For example, the famous Art Gallery Problem [7] deals with determining the number of observers necessary to cover an art gallery room such that every point is detected by at least one observer.

One approach to sensing coverage is the deployment of sensors, which is a critical issue in existing works. A well-planned deployment can help maximize the sensing coverage while activating as few sensors as possible. A. Ghosh [8] studies the problem of coverage holes under random deployment. An algorithm is proposed to achieve a tradeoff between the cost of deployment and the percentage of area covered. In [9] and [10], two algorithms are proposed to efficiently deploy sensors and to maximize the coverage.

Different from traditional sensors, the coverage of a directional sensor is determined by both its location and orientation. Ma *et al* [11] provide a directional sensor model where each sensor is fixed to one direction and analyzes the probability of full area coverage. In [12,13], Tao presents two coverage-enhancing algorithms to minimize the overlapping sensing area of directional sensors, according to the characteristic of adjustable orientations of directional sensors. In [14–16], some algorithms are proposed to cover maximal number of targets. Unfortunately, they limit orientations of a directional sensor, i.e. the sensor they assume cannot rotate its orientation continuously. The previous works assume commonly that after directional sensors are deployed randomly, they adjust orientations to achieve efficient coverage.

Given a directional sensor network, we are interested

in designing a deployment algorithm that is able to optimize coverage. S. Megerian *et al* [17] present the worst-case coverage in WSNs. They attempt quantifying the QoS by finding areas of lower observability from sensors and detecting *breach* regions, where breach is defined as the minimum Euclidean distance from any point on a path to any sensor. Due to find a path, where minimize the probability of detecting the target moves along this path, they define *maximal breach path*. Similarly, J. Adriaens *et al* [18] put forward an optimal polynomial time algorithm for computing the worst-case coverage in DSNs. Both of them present centralized methods using the Voronoi diagram to solve the worst-coverage problem. The use of Voronoi diagram, efficiently and without loss of optimality, transforms the continuous geometric problem into a discrete graph problem.

In this paper, we mainly address the problem of maximal coverage while activating as few sensors as possible, called optimal coverage in directional sensor networks (OCDSN) problem. For the solution to this problem, we will propose two optimizing coverage algorithms based on the Voronoi diagram. Both of two algorithms, called the Voronoi-based centralized approximation (VCA) algorithm and the Voronoi-based distributed approximation (VDA) algorithm, contribute to making edges of Voronoi cover as more as possible, which could keep the worst-case coverage. Further, it is approximately considered that if most paths are covered, most Voronoi polygons are covered, i.e. most of the given monitoring area is covered.

From the perspective of directional sensors, the effect of coverage has three parameters: the number of sensors, sensing range and the field of view (FOV). As shown in the following sections, we will discuss the impact of these parameters on the coverage of a directional sensor network.

The rest of this paper is organized as follows: We formally establish a mathematical model for representing the FOV in Section 2, and then define the OCDSN problem and prove that it is NP-complete in Section 3. Furthermore, we propose VCA and VDA of solution to the OCDSN in Section 4. Simulation results are presented to show the effectiveness of the proposed algorithms in Section 5. Finally, we conclude the paper in Section 6.

## 2. Preliminaries

### 2.1. Sensing model

A directional sensor has a finite angle of view. From the concept of FOV, its sensing region can be viewed as a sector in a two-dimensional plane denoted by 4-tuple $(L, R_S, \alpha, \beta)$ as shown in Figure 1. The sensing model of a directional sensor is defined as follows.

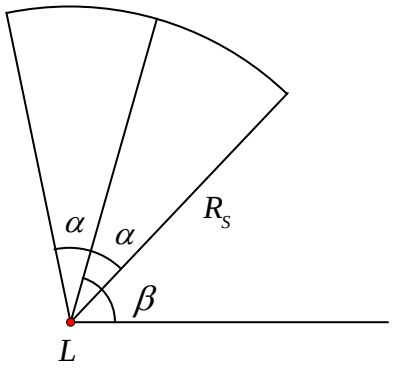



**Figure 1. Sensing model of a directional sensor.**

**Definition 1** *The directional sensing model:* 4-tuple $(L, R_S, \alpha, \beta)$

- $L$: the location of the sensor.
- $R_s$: sensing range of the sensor.
- $\alpha$: the offset angle of FOV , which equals to the half of the vertex angle of sensing sector.
- $\beta$: the horizontal angle to the midline of sensing sector, $0 \le \beta \le 2\pi$. This parameter defines the orientation of the directional sensor.

### 2.2. Notations and Assumptions

We adopt the following notations throughout the paper.

- $A$: the given specified area to be covered.
- $N$: the number of sensors.
- $S_i$: the i[th] sensor, $1 \le i \le N$.
- $S$: the set of sensors. $S = \{s_1, s_2, ..., s_N\}$.
- $\varphi_{i,\beta}$: the coverage of $S_i$ whose orientation is $\beta$, $1 \le i \le N$, $0 \le \beta \le 2\pi$.
- $\Phi$: the set of the coverage of all sensor. $\Phi = \{\varphi_{i,\beta} \mid i = 1,2,...N,\ 0 \le \beta \le 2\pi\}$.
- $P \in \varphi_{i,\beta}$: the point $P$ covered by $S_i$, $1 \le i \le N$, $0 \le \beta \le 2\pi$.

A point $P$ is said to be covered by sensor $S_i$ if and only if the following conditions are satisfied:

1) $dis(L_i, P) \le R_S$, where $dis(L_i, P)$ is the Euclidean distance between the location $L_i$ of sensor $S_i$ and point $P$. Since $S_i$ corresponds to $L_i$, throughout the rest of the paper, unless otherwise mentioned, $dis(s_i, P)$ means $dis(L_i, P)$.

2) The horizontal angle to $\overrightarrow{L_iP}$ is within [$\beta_i - \alpha$, $\beta_i + \alpha$].

An area $A$ is covered by sensor $S_i$, if and only if for any point $P_j \in A(j = 1,2,...,\infty)$, $P_j$ is covered by $S_i$, i.e. $P_j \in \varphi_{i,\beta}$.

For simplicity and computability, we make assumptions as below, however some of them can be relaxed.

- Directional sensors are homogeneous. Specifically, all sensors have the same omnidirectional communication range $R_c$ and shape of sensing sectors (i.e. $R_s$ and α).
- The communication range is at least twice as large as the sensing range, i.e. $R_c \geq 2R_s$.
- Every directional sensor knows its exact location information.
- Each direction of every sensor has a uniform sensing sector.

### 2.3. Geometry Notation

The Voronoi diagram is an important data structure in computational geometry, which is a fundamental construct defined by a discrete set of sites [19]. In a two-dimensional plane, the Voronoi diagram partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one site. This construction effectively produces polygons with edges that are equidistant from neighboring sites.

We define the Voronoi polygon of $S_i$ as $VP(S_i)$. According to the property of Voronoi diagram, if an arbitrary point $P \in VP(s_i)$ then $dis(P,s_i) \leq dis(P,s_j)$, where $i, j = 1,2,...,N$ and $i \neq j$. Therefore, if a sensor cannot detect the expected phenomenon in its Voronoi polygon, no other sensor can detect it (assume that the directional sensor can adjust its orientation circularly). The Voronoi diagram generated by the set of sensors $S$ is defined as $VD(S) = \bigcup_{s_i \in S} VP(s_i)$.

However, some of the lines at the boundaries of the Voronoi diagram extend to infinity. In this paper, the monitoring area with our supposal is finite, so we clip the Voronoi diagram within the boundary polygons. To do this, we introduce extra edges in the Voronoi diagram corresponding to the boundary edges and discard any line segments that lie outside the boundary of the field as shown in Figure 2. Let $BVD(S)$ be the graph by removing all edges of $VD(S)$ that no longer than the given area $A$, i.e. $BVD(S) = VD(S) \cap A$.

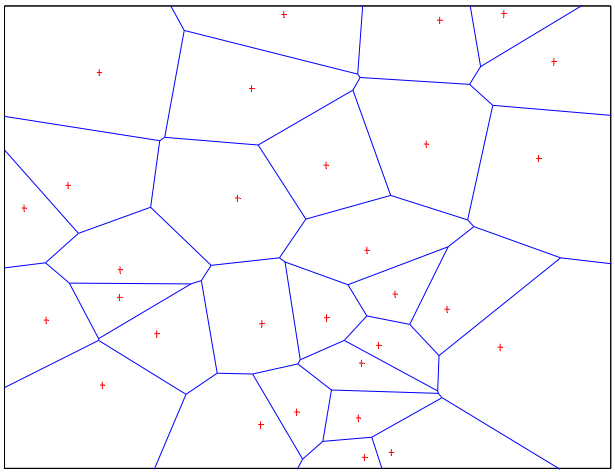


**Figure 2. Boundary Voronoi diagram.**

## 3. Problem Definition

Sensors are randomly deployed when we initialize the network, so the whole monitoring area is not always covered by this initial deployment. Further, it is unnecessary that all sensors are active. Our goal is to schedule the orientations in order to cover maximal area while activating as few sensors as possible, called the optimal coverage in directional sensor networks (OCDSN) problem. This problem can be defined as follows.

**Definition 2** *Optimal coverage in directional sensor networks (OCDSN) problem*: Given a specified area $A$, a set of directional sensors $S$, and each sensor with four parameters $L_i$ $R_s$, $\alpha$ and $\beta$, find a subset $Z$ of Φ, with the constraint that at most one $\varphi_{i,\beta}$ can be chosen for the same $i$ (i.e. an active sensor has only one orientation), to maximize the union of chosen $\bigcup \varphi_{i,\beta}$ (i.e. the covered area), while minimizing the cardinality of Z=$\{\varphi_{i,\beta} | (i, \beta)$ is chosen$\}$ (i.e. the number of active directional sensors).

**Definition 3** *Decision version of OCDSN*: Given a specified area $A$, a set of directional sensors $S$, and each sensor with four parameters $L_i$ $R_s$, $\alpha$ and $\beta$, determine if there exists a subset $Z$ of Φ with $u_0$ elements of Φ, with the constraint that at most one $L_i$ $R_s$, $\alpha$ and $\beta$, can be chosen for the same $i$, covering at least $p_0 A$, where $u_0$ is a given positive integer from 1 to $N$ and $p_0$ is a given positive pure decimal called the required ratio of coverage.

The following theorem shows the complexity of the OCDSN problem.

**Theorem 1** *The OCDSN problem is NP-complete.*

**Proof:** We follow two steps to prove this theorem.

First, we prove that OCDSN∈NP. The non-deterministic OCDSN algorithm is described as follows: Select $u_0$ elements of the set of directional sensors $S$ and assign a random orientation to each of these selected sensors, so that 2-tuple ($i$, $\beta$) corresponds to $\varphi_{i,\beta}$. Then check that if the union of chosen sensors covers at least $p_0 A$, i.e. $\bigcup \varphi_{i,\beta} \geq p_0 A$. It is easy to see this non-deterministic algorithm can be verified in a polynomial time. Therefore, OCDSN∈NP.

Second, to prove that the decision version of OCDSN is NP-hard, we show a polynomial time reduction 3-CNF-SAT to OCDSN, i.e. 3-CNF-SAT$\propto$OCDSN. Before the proof, we suppose a special case where sensors are isotropic (i.e. $\alpha = \pi$) and it requires the whole coverage (i.e. $p_0$ =1). Furthermore, a plane can be regarded as the set of infinite points, so we assume that our goal to cover the area $A$ is equivalent to covering infinite points.

For the 3-CNF-SAT problem, a Boolean formula $F$ consisting of infinite clauses and $N$ variables is in 3-conjunctive normal form, i.e. $F = \bigwedge_{j=1}^{\infty} c_j$, where each clause $c_j = l_{j,1} \vee l_{j,2} \vee l_{j,3}$ and each literal $l_{j,k} \in \left\{l_1, \overline{l_1}, ..., l_N, \overline{l_N}\right\}$. $(k = 1,2,3)$ From the given formula $F$, an instance of OCDSN is constructed as follows:

- $A = \left\{c_j \mid j = 1,2,...\right\}$.
- $\varphi_{i,0} = \left\{c_j \mid l_i \text{ is a literal in } c_j,\ j = 1,2,...\right\}$.
- $\varphi_{i,\pi} = \left\{c_j \mid \overline{l_i} \text{ is a literal in } c_j,\ j = 1,2,...\right\}$.

This construction can be finished in a polynomial time.

Now we prove that $F$ is satisfiable if and only if $A$ is covered. If $F$ is satisfiable, then there is a set of truth values for $l_i, i = 1,2,...,N$, such that each clause is true with this assignment. Thus, with this assignment at least one literal is true in each clause. Since each literal corresponds to a $\varphi_{i,0}$ or $\varphi_{i,\pi}$, picking these true literals yields a subset $Z$ of $\Phi$, where the cardinality of $Z$ is the number of active sensors, i.e. $u_0 = \|Z\|$. Note that $l_i$ and $\overline{l_i}$ cannot both be true, so the corresponding $\varphi_{i,0}$ and $\varphi_{i,0}$ in $\Phi$ cannot both be chosen into $Z$. Therefore, $A$ is covered by $Z$.

Conversely, if $A$ is covered by a subset $Z$ of $\Phi$ and we suppose that $\|Z\| = N$, then we assign true to the literals which elements in $Z$ correspond to. Obviously, every clause is true because at least one of its literals is true. Therefore, $F$ is satisfiable.

We conclude that 3-CNF-SAT$\propto$OCDSN. The 3-CNF-SAT problem is known to be NP-complete, so the OCDSN problem is NP-hard.

Since the OCDSN problem is NP and NP-hard, the result follows.

## 4. Solution

Since the OCDSN problem is NP-complete, we propose two algorithms, the centralized algorithm and the distributed algorithm based on Voronoi diagram, both of which solve the OCDSN problem efficiently.

### 4.1. Voronoi-Based Centralized Approximation (VCA) Algorithm

To solve the NP-complete OCDSN problem as well as possible, we present the VCA algorithm that needs the global information.

The main idea of VCA is based on the greedy principle and can be described as follows. Initially, we deploy a set of directional sensors $S$ randomly in the monitoring area $A$, all of which are inactive (i.e. not active). Secondly generate $BVD(S)$ and construct $F$={*maxlen* ($S_i$),$i$ =1,2,…, $N$}, where *maxlen* ($S_i$) is the maximal length of uncovered edges of $BVD(S)$ which $S_i$ can be possible to cover. VCA runs in loops. In each loop, calculate the maximal element of $F$ and let $S_j$, the corresponding sensor, be active. Rotate the orientation of $S_j$ to make it cover the maximal length of uncovered edges, remove $S_j$ from $F$ and update $F$. If there are no more edges of $BVD(S)$ to be covered or no more inactive directional sensors remaining, i.e. $F$ is empty, the algorithm terminates; otherwise, directional sensors are activated iteratively according to the above greedy rule.

The pseudo-code of the VCA algorithm is shown below.

**VCA algorithm**

Deploy $N$ sensors randomly in monitoring area
location$(i) = (x_i, y_i)$
state$(i)$ = 'inactive' // Initialize the network, and let all sensors be inactive.
Generate $BVD(S)$
Construct $F = \{maxlen(s_i), i = 1,2,...,N\}$
while $F$ is not empty && there is one edge uncovered
  Calculate the maximal element of $F$
  $s_j = \arg\max\{maxlen(s_i)\}$
  state$(j)$ = 'active'
  Rotate $S_j$ orientation to make it cover the maximal length of uncovered edges
  Remove $S_j$ from $F$
  Update $F$
end while

Given $N$ sensors, the best known algorithms for the generation of the Voronoi diagram have $O(N\log N)$ complexities. In 2D, Voronoi diagrams are essentially linear complexity in terms of vertices and edges. For $N$ sensors, $\|E\|$ (numbers of edges) in the Voronoi diagram is $O(N)$, so constructing and updating $F$ need $O(N^2)$. The best order algorithms have $O(N\log N)$ complexities, thus calculating the maximal element needs $O(N\log N)$. The complexity of while loop is $O(N(N\log N + N^2)$. Therefore, the total time of the VCA algorithm is as follows:

$$\begin{aligned} &O(N\log N) + O(N^2) + O(N(N\log N + N^2)) \\ &= O(N(N\log N + N^2)) \end{aligned}$$

### 4.2. Voronoi-Based Distributed Approximation (VDA) Algorithm

As above, we can achieve the solution to OCDSN problem approximately in polynomial time. However, the VCA algorithm needs the global information, which in-

curs high cost. Without global information available in a centralized location, each sensor must make its decision independently based only on local information gathered from the *neighbors*. In this subsection, we present the VDA algorithm.

**Definition 5** *The set of neighbors of* $S_i$:

$$Neighbor(i) = \{s_j \in S \mid dis(s_i, s_j) \le 2R_S,\ j = 1, 2, ..., N\}.$$

**Definition 6** *The weight of* $S_i$:

$$w_i = \begin{cases} \dfrac{\sum_{j=1}^{\|E_i\|} \|e_{i,j}\|}{\|E_i\|}, & \text{if } \|E_i\| \neq 0, \\ 0, & \text{otherwise} \end{cases}$$

where $\|e_{i,j}\|$ is the length of $e_{ij}$ and $e_{ij}$ is the $j^{\text{th}}$ uncovered edge of $BVD(S)$ that $S_i$ has the capability of sensing, and $E_i$ is the set of all $e_{ij}$. Indeed, $w_i$ represents the priority among neighbors of $S_i$.

**Definition 7** *Local Voronoi diagram of* $S_i$:

$$LVD(s_i) = C(s_i) \cap BVD(S),$$

where $C(S_i)$ is the communication region of $S_i$, i.e. $C(s_i) = \{P \in A \mid dis(s_i, P) \le R_C\}$.

As shown in Figure 3, each directional sensor can calculate its local Voronoi diagram by collecting locations of all sensors in its communication region. Note that we assume $R_C \ge 2R_S$, so $\varphi_{i,\beta} \in C(s_i)$. Therefore we can discuss the coverage of each sensor in its local Voronoi diagram respectively.

The VDA algorithm is simply described as follows.

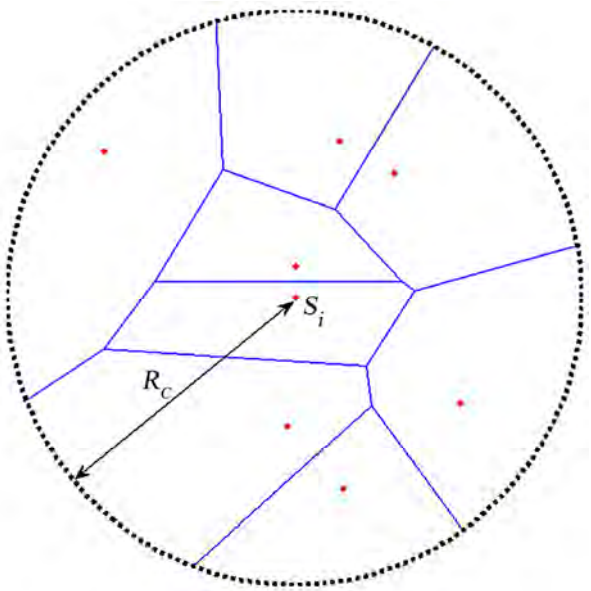



**Figure 3. Local Voronoi diagram of $S_i$.**

Initially, each directional sensor is in the active state and collects locations of all sensors in its communication region. Then each sensor generates local Voronoi diagram respectively, and calculates its weight. Each sensor starts to collect the information of its neighbors, i.e. weights, covered edges, and states. Upon receiving the information and updating its weight, each sensor makes its decision independently as follows. If its weight is maximal, it chooses the orientation for the purpose of covering the maximal edge and sends out a new message to inform its neighbors. If its weight is zero, it triggers a transition timer, with random duration $T_r$. The timer is canceled if new information from the neighbors arrives and changes the weight to a non-zero value. Note that the purposes of setting the transition timer $T_r$. are 1) to prevent a sensor finalizing its decision before its neighbors with higher weight and 2) to transfer its state to inactive in time.

The pseudo-code of the VDA algorithm is shown below.

**VDA algorithm**

**1. Initialization phase (only performed once)**
Deploy $N$ sensors randomly in $A$
$\text{location}(i) = (x_i, y_i)$
$\text{state}(i) = \text{'active'}$
$S_i$ collects locations of sensors in $C(S_i)$, and generates $LVD(s_i)$
Calculate $w_i$
Send the information message to neighbors

**2. Decision phase**
for receiving message from neighbors
  Update $w_i$
  if $w_i$ is maximal among neighbors of $S_i$
    Rotate the orientation to make covering the maximal edge
    Send the information message to neighbors
    if transition timer of $S_i$ is on
      turn the transition timer off
    end if
    continue
  end if
  if $w_i$ =0 and transition timer of $S_i$ is off
    turn the transition timer on
  end if
  if transition timer of $S_i$ remains on for longer than $T_r$
    $\text{state}(i) = \text{'inactive'}$
    return
  end if
end for

**Table 1. Parameters setting.**

| Parameter | Default | Variation |
|---|---|---|
| *A* Area | 500*500 | 500*500 |
| Sensor number *N* | 200 | 70-1000 |
| Sensing range $R_S$ | 50 | 30-120 |
| Offset angle $\alpha$ | $\pi/3$ | $\pi/6$–$\pi$ |

It is clear that the VDA algorithm terminates in finite time when all edges are covered or all sensors make their decisions. By analyzing the pseudo-code, the initialization phase has $O(N\log N)$ complexities, while the time of decision phase is $O(N)$.

## 5. Simulations

In this section, we evaluate the performance of two algorithms by simulations executed in MATLAB 7.4.0. The simulation parameters are summarized in Table 1.

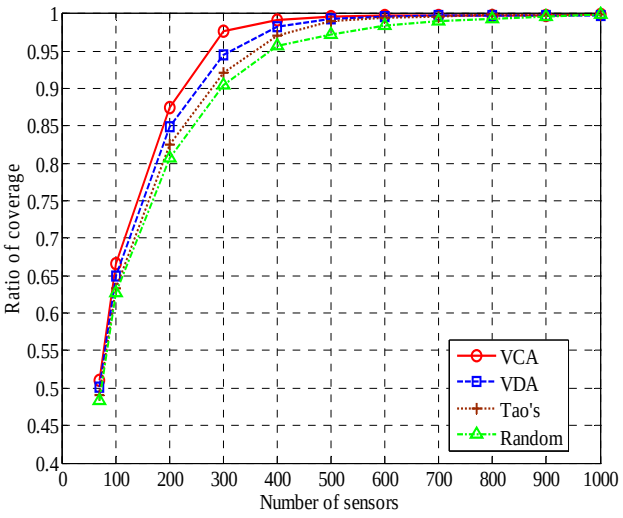
We compare the performance of VCA and VDA with the coverage-enhancing algorithm which Tao [13] presented and the random algorithm in which every sensor separately selects its orientation randomly. For random algorithm, we run it 100 times and get the average value.

First, we examine the effect of the number of sensors *N*. As shown in Figure 4, with the increase of sensors deployed, both the ratio of coverage and number of active sensors for all three algorithms increase linearly until *N* approaches 400; upon passing such a value, the number of active sensors increases slowly or even decreases whereas the ratio of coverage continuously increases and then becomes saturated when *N* is above 400 or so. To state the difference: for the ratio of coverage, VCA always behaves better than VDA, obviously Tao's and random algorithm; for the number of active sensors, VCA activates larger number of sensors than VDA, and the gap between two algorithm is nearly unfluctuating till *N* exceeds a value ($\geqslant 400$ in this simulation). The reasons that incur the above difference, are that 1) VCA and VDA decrease overlapped area in dense network by rotating orientations of directional sensors, so both of them work better than the random algorithm; and 2) VCA knows global information, while in VDA every sensor makes its decision independently based only on local information, so VCA achieves higher ratio of coverage with more sensors whereas VDA activates less sensors with lower ratio of coverage.

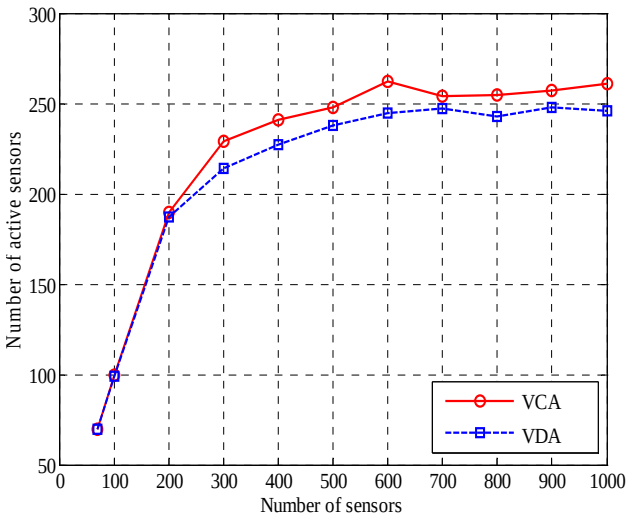
Figure 5 and Figure 6 show the influence of the sensing range and the offset angle on our two algorithms respectively. Clearly with the increase of $R_s$ or $\alpha$, the ratio of coverage increases for VCA, VDA and random algorithm, while the number of active sensors decreases. Also, for the ratio of coverage, VDA works better than VCA, Tao's and random algorithm, and VCA excels VDA in the number of active sensors. The intuitive reasons for results are similar to the effect of number of sensors. Notice that when $R_s$ or $\alpha$ passes such a great value ($R_s$ approaches 120 and $\alpha$ approaches $\pi$ in simulations), the number of active sensors of VDA is near to that of VCA. This is because the ratio of coverage is saturated till $R_s$ or $\alpha$ approaches a great value, and both of two algorithms need a certain number of sensors.

## 6. Conclusions and Future Work

Coverage is always an important issue for sensor networks. Different from many other papers designed for isotropic sensor networks, this paper has studied the problem of optimal coverage of directional sensor net



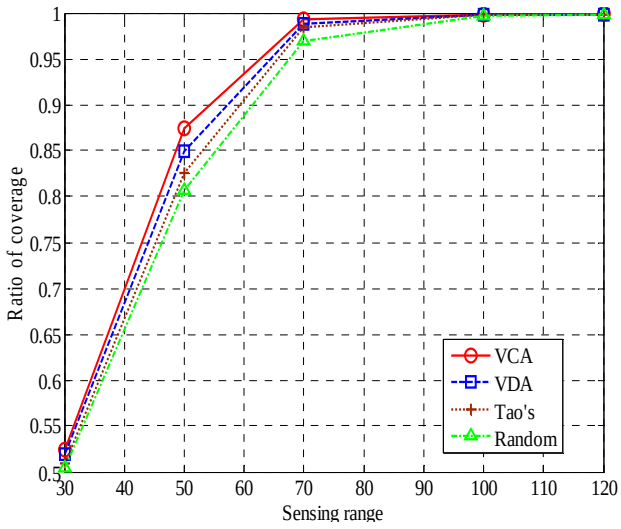

**(a) Ratio of coverage vs. number of sensors**



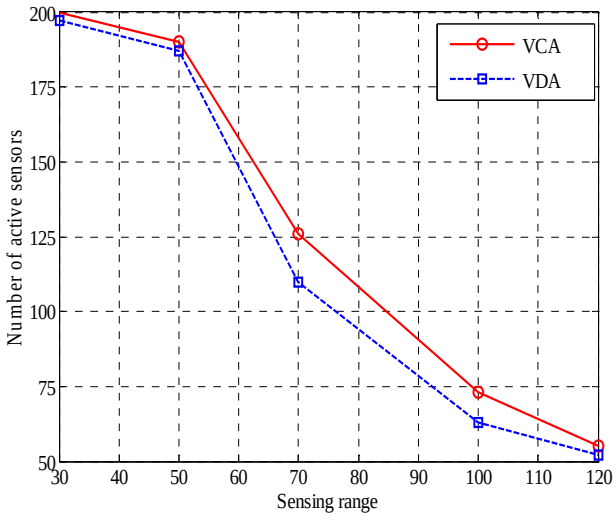

**(b) Number of active sensors vs. number of sensors**

**Figure 4. Effect of number of sensors with $R_S = 50, \alpha = \pi/3$.**

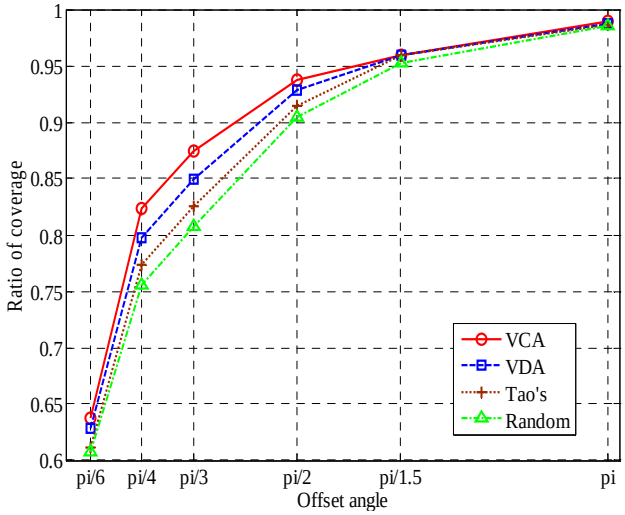



**(a) Ratio of coverage vs. sensing range**

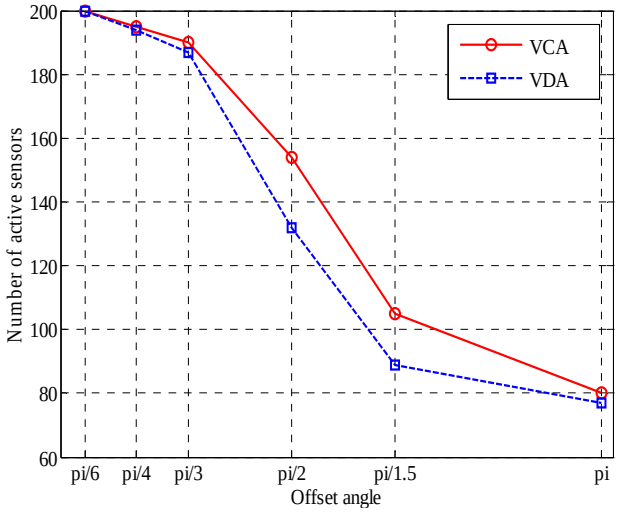



**(b) Number of active sensors vs. sensing range**

**Figutre 5. Effect of sensing range with $N = 200, \alpha = \pi/3$ .**




**(a) Ratio of coverage vs. offset angle**




**(b) Number of active sensors vs. offset angle**

**Figure 6. Effect of offset angle with $N = 200, R_S = 50$ .**

works and proved this problem is NP-complete. After the definition of sensing model and some assumptions, we present a centralized approximation algorithm and a distributed approximation algorithm to solve the OCDSN problem, based on the boundary Voronoi diagram. As our algorithms finish, we can obtain the optimizing covered area. Finally, we systematically evaluate the performance of two algorithms through extensive simulations. The results show that VCA and VDA work better than the coverage-enhancing algorithm which Tao presented and the random algorithm, and fewer sensors can cover more area so that more sensors can be inactive. Moreover, we analyze the advantage and disadvantage of VCA and VDA respectively. As a future work, we plan to design energy efficient algorithms to prolong the lifetime of directional sensor networks.

## 7. Acknowledgments

We are grateful that the subject is sponsored by the National Natural Science Foundation of P. R. China (No. 60773041), the Natural Science Foundation of Jiangsu Province(BK2008451), National 863 High Technology Research Program of P. R. China (No. 2006AA01Z219), High Technology Research Program of Nanjing City (No. 2007RZ106, 2007RZ127), Foundation of National Laboratory for Modern Communications (No. 9140C11-05040805), Jiangsu Provincial Research Program on

Natural Science for Higher Education Institutions (No. 07KJB520083) and Special Fund for Software Technology of Jiangsu Province. Postdoctoral Foundation of Jiangsu Province(0801019C), Science & Technology Innovation Fund for higher education institutions of Jiangsu Province(CX08B-085Z, CX08B-086Z ).

## 8. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," ACM Trans. on Multimedia Computing, Communications and Applications, Vol. 40, No. 8, pp. 102–114, August 2002.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 214–226, 2004.

[3] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," Information Processing in Sensor Networks, IPSN, 6th International Symposium on, pp. 69–78, April, 2007.

[4] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In situ image sensing and interpretation in wireless sensor networks," SenSys, pp. 192–204, 2005.

[5] W. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, "Panoptes: Scalable low-power video sensor networking technologies," ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 1, No. 2, pp. 151–167, May 2005.

[6] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Rangeonly slam for robots operating cooperatively with sensor networks," Robotics and Automation, ICRA, Proceedings 2006 IEEE International Conference on, pp. 2078–2084, May 2006.

[7] M. Marengoni, B. A. Draper, A. Hanson, and R. A. Sitaraman, "A System to Place Observers on a Polyhedral Terrain in Polynomial Time," Image and Vision Computing, Vol. 18, pp. 773–80, December 1996.

[8] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," Local Computer Networks, 29th Annual IEEE International Conference on, pp. 68–76, November 2004.

[9] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," Information Fusion, Proceedings of the Fifth International Conference on, Vol. 2, pp. 1581–1587, July 2002.

[10] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in Proceedings of IEEE WCNC, pp. 1609–1614, March 2003.

[11] H. Ma and Y. Liu, "On coverage problems of directional sensor networks," in International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Vol. 3794, pp. 721–731, 2005.

[12] D. Tao, H. Ma, and L. Liu, "A virtual potential field based coverage-enhancing algorithm for directional sensor networks," Journal of Software, Vol. 18, No. 5, pp. 1152−1163, May 2007.

[13] D. Tao, H. Ma, "Coverage-enhancing algorithm for directional sensor networks," Stojmenovic I, Cao JN, eds. Proceedings of the 2nd International Conference on Mobile Ad-Hoc and Sensor Networks, pp. 256−267, 2006.

[14] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," Journal of Combinatorial Optimization, Vol. 11, No. 1, pp. 21–41, February 2006.

[15] Y. Cai, W. Lou, and M. Li, "Cover set problem in directional sensor networks," Proceedings of the Future Generation Communication and Networking (FGCN 2007), Vol. 01, pp. 274–278, 2007.

[16] J. Wen, L. Fang, J. Jiang, and W. Dou, "Coverage optimizing and node scheduling in directional wireless sensor networks," Wireless Communications, Networking and Mobile Computing, WiCOM'08, 4th International Conference on, pp. 1–4, October 2008.

[17] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Worst- and best-case coverage in sensor networks," IEEE Transactions on Mobile Computing, Vol. 4, No. 1, pp. 84–92, February 2005.

[18] J. Adriaens, S. Megerian, and M. Potkonjak, "Optimal worst-case coverage of directional field-of-view sensor networks," Sensor and Ad Hoc Communications and Networks, SECON'06, 3rd Annual IEEE Communications Society on, Vol. 1, pp. 336–345, September 2006.

[19] F. Aurenhammer, "Voronoi diagrams-A survey of a fundamental geometric data structure," ACM Computing Surveys, Vol. 23, No. 3, pp. 345–405, September 1991.

*Wireless Sensor Network*, 2009, 1, 425-433
doi:10.4236/wsn.2009.15051 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Investigations into Capacity of MIMO Ad Hoc Network Including Effects of Antenna Mutual Coupling

**Xia LIU[1], Marek E. BIALKOWSKI[2], Feng WANG[2] , Konglit HUNCHANGSITH[2]**
[1]*Correspondant author, School of ITEE, the University of Queensland, Brisbane, Australia*
[2]*School of ITEE, the University of Queensland, Brisbane, Australia*
*Email*: {*xialiu, meb, fwang, konglit*}*@itee.uq.edu.au*
*Received August* 6, 2009; *revised September* 27, 2009; *accepted September* 30, 2009

## Abstract

This paper reports on investigations into capacity of ad hoc network whose nodes are equipped with multiple element antennas (MEAs). The investigation of this multi-user Multiple Input Multiple Output (MIMO) system takes into account mutual coupling (MC) in addition to spatial correlation that is present in array antennas. A closed-form expression for an upper bound of mutual information (capacity) of MIMO ad hoc network is derived. An optimal signal transmission scheme is proposed to maximize the MIMO ad hoc network capacity. Simulation results for capacity of non-optimized and optimized cases of signal transmission are presented.

**Keywords:** MIMO, Ad Hoc Network, Channel Capacity, Mutual Information, Mutual Coupling, Spatial Correlation

## 1. Introduction

In recent years, the signal transmission technique employing multiple element antennas (MEAs) at two sides of a communication link has shown a great potential to significantly improve the transmission quality of wireless communication systems without the need for extra operational frequency bandwidth [1–3]. Because of this attribute, the multiple-input multiple-output (MIMO) technique is envisaged for the next generation of mobile communications.

Benefits of MIMO technique over the traditional single-input single-output (SISO) signal transmission with respect to the capacity and Bit Error Rate (BER) have been demonstrated for single-user (peer-to-peer) scenarios. An ultimate goal is to prove its advantages for multi-user wireless communication networks.

Wireless networks are usually formed by cells each with a base station (BS) and many mobile stations (MS). Mobile terminals communicate with the base station (BS) which organizes multiple-access to the intended MS users. To implement the MIMO concept for such a standard mobile communication system, MS and BS have to be equipped with multiple element antennas and suitable signal processing algorithms.

Nowadays, there is an increased interest in mobile ad hoc networks in which pairs of mobile nodes communicate directly with each other without the involvement of BS. The aim is to improve spectral efficiency and capacity of such networks [4,5]. In order to form a MIMO ad hoc network, multiple element antennas with associated signal processing algorithms have to be implemented at mobile nodes.

It is known that MIMO systems operate well in rich scattering environments. This is because such environments support virtual multiple channels which are statistically independent. Finite spacing of multiple element antennas introduces spatial correlation which decreases the effective degree of freedom (EDOF) [6] and thus reduces the MIMO system capacity. The adverse effect of spatial correlation has been pointed out both for single and multiple user scenarios [6–8]. The finite antenna spacing in array antennas is also responsible for mutual coupling which adversely affects power transmission and reception. The mutual coupling effect is especially pronounced in tightly spaced arrays. Because there is a considerable demand for compact size MS terminals, the effect of mutual coupling cannot be neglected and has to be taken into account while assessing the MIMO link performance. The effect of mutual coupling in MIMO systems for the case of peer to peer communication was addressed via simulations and measurements in [9–12] where it was demonstrated that in some cases the mutual coupling could reduce spatial correlation and improve the channel capacity.

In this paper, we report on investigations into the capacity of MIMO ad hoc network. The investigations take into account both the spatial correlation and the mutual coupling that are present in array antennas. A closed-form expression for an upper bound of the mutual infor-

mation (capacity) of MIMO ad hoc network taking into account spatial correlation and mutual coupling is derived. Assuming that the channel state information (CSI) is known both by the receiving node and the transmitting node, a cooperative communication is proposed to fight multi-user interference. It is shown that using the cooperative communication a higher MIMO channel capacity is achieved in comparison with the case when the CSI acquired at the receiving node is not shared with the other nodes of the network.

The rest of the paper is organized as follows. Section 2 describes the network configuration and the channel model that takes into account mutual coupling in transmitting and receiving array antennas. The capacity of non-cooperative and co-operative ad hoc network is given in Section 3. Section 4 presents the simulation results and Section 5 concludes the paper.

## 2. System Configuration and Channel Model

### 2.1. System Configuration

In this paper, a narrowband MIMO ad hoc network consisting of $N$ nodes is considered. Its configuration is shown in Figure 1. Each node is assumed to use a transceiver equipped with a uniform linear array of $M_{rt}$ vertically polarized wire dipole antennas. The choice of wire dipoles is justified by the fact that there are analytical solutions for the spatial correlation and mutual coupling for such type of array antenna. The network operates in a scattering environment in which each node is surrounded by scattering objects uniformly distributed within a circle, as shown in Figure 1.

A channel formed between a pair of node $i$ and the transmitting node $j$ is characterized by the $M_{rt}$ x $M_{rt}$ complex channel matrix $\mathbf{H}_{ij}$ whose elements are given by complex transmission coefficients between the individual antenna elements at nodes $j$ and $i$.

Under these assumptions, the signal received at node $i$ is described by the following equation,

$$\mathbf{y}_i = \mathbf{H}_{ij}\mathbf{s}_j + \sum_{k=1,k\neq j}^{N-1}\mathbf{H}_{ik}\mathbf{s}_k + \mathbf{n}_i \qquad (1)$$

The first term in (1) represents the signal received from the desired node as given by the channel matrix $\mathbf{H}_{ij}$; and the $M_{rt}$ x 1 transmitted signal vector $\mathbf{s}_j$. The second term is the sum of co-channel interference and $\mathbf{n}_i$ is the $M_{rt}$ x 1 noise vector. The co-channel interference-plus-noise (IPN) can be expressed as,

$$\mathbf{N}_i = \sum_{k=1,k\neq j}^{N-1}\mathbf{H}_{ik}\mathbf{s}_k + \mathbf{n}_i \qquad (2)$$

The correlation of IPN is given by the following expression,

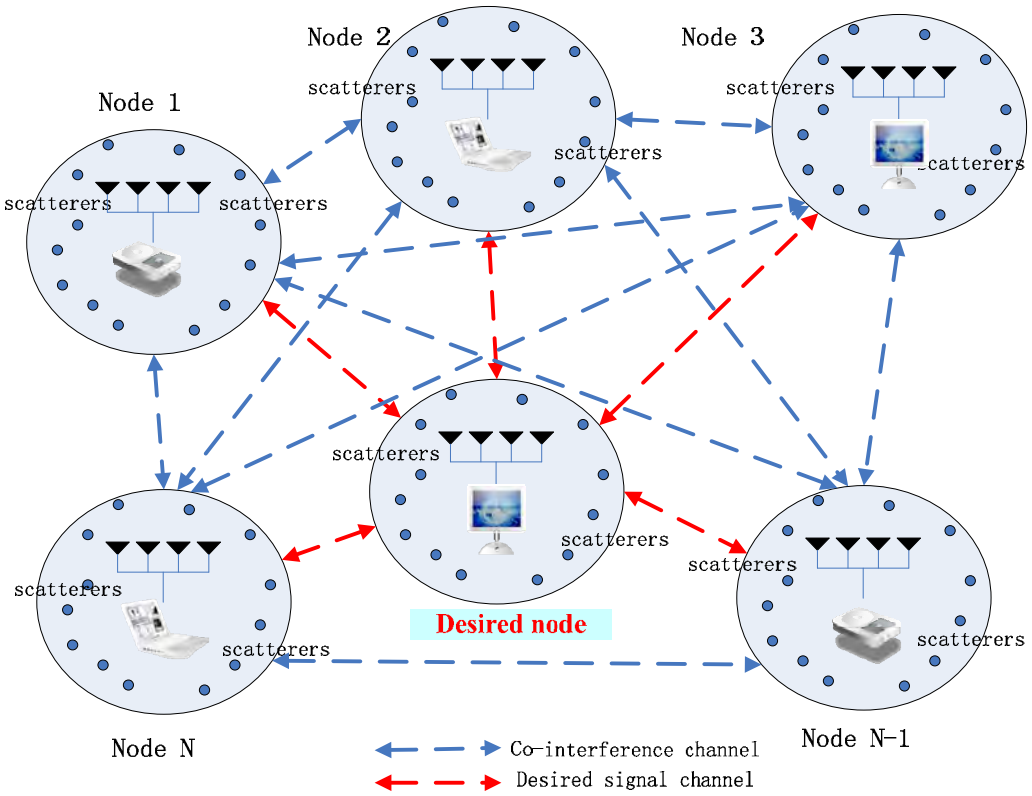



**Figure 1. MIMO ad hoc network.**

$$R_{N_i} = E\{N_i N_i^H\} = \sum_{k=1,k\neq j}^{N-1} \mathbf{H}_{ik} Q_k \mathbf{H}_{ik}^H + \sigma_n^2 I_{M_{rt}} \quad (3)$$

where $Q_k = E\{s_k s_k^H\}$ is the data covariance matrix representing the signal transmission scheme from the *k*-th inference node to node *i*. $E\{\}$ denotes the statistical expectation.

For the case of non-optimized ad hoc network, only receivers have the knowledge of CSI. In such a case, the best strategy for the nodes is to transmit equal power over the individual transmitting antenna elements. In this case, the $Q_{ik}$ of the *k*-th interference node can be expressed as:

$$\begin{aligned} &Q_{ik} = \frac{p_k}{N_t} I, \\ &subject \quad to: P_I = \sum_{k=1}^{N-1} p_k \,,\, p_1 = p_2 = ... = p_k \end{aligned} \quad (4)$$

where $p_k$ is the transmit power from node *k* and $P_I$ is the total transmitted power of all interferences.

By using (4), Equation (3) can be rewritten as:

$$R_{N_i} = \frac{P_I}{N-1} \sum_{k=1,k\neq j}^{N-1} \mathbf{H}_{ik} \mathbf{H}_{ik}^H + \sigma_n^2 I_{M_{rt}} \quad (5)$$

The correlation of received signal at node *i* that is transmitted from the desired node is obtained as:

$$R_i^y = E\{y_i y_i^H\} = \mathbf{H}_{ij} Q_j \mathbf{H}_{ij}^H + R_{N_i} \quad (6)$$

in which $Q_j = E\{s_j s_j^H\}$ is the data covariance matrix representing the signal transmission scheme from node *j* to node *I*, which is also the mutual information for the channel formed between the desired node *j* and node *i*. $tr\{Q_j\} = P_j$ and $P_j$ is the total transmitted signal power from the desired node.

The capacity of the system can be improved by optimizing the signal transmission scheme. The optimization process requires the knowledge of CSI by transmitters which is usually obtained by a feedback loop from receivers. In this case, the new data correlation matrix, subject to constraint of total transmitted power $tr\{Q_j\} = P_j$, has to be worked out.

## 2.2. Channel Model Neglecting Mutual Coupling

In the undertaken investigations, a narrow-band flat block-fading MIMO channel is assumed between the network nodes. For each pair of nodes, the channel matrix (**H**) can be used to describe the channel properties. Its entries depend on the signal propagation environment and properties of the antenna arrays used at the two sides of communication link.

In the initial stage, the Kronecker representation of channel [13,14] neglecting the effects of mutual coupling is assumed. In this representation, the transmitter and receiver correlations are separable and the channel matrix **H** (for brevity the subscripts are dropped here) is represented as:

$$\mathbf{H} = R_R^{1/2} G_H R_T^{1/2} \quad (7)$$

where $G_H$ is the matrix including identical independent distributed (i.i.d) Gaussian entries with zero mean and unit variance, and $R_R$ and $R_T$ are the spatial correlation matrices at the receiver and transmitter, respectively. The channel correlation is expressed as,

$$R_H = E\{\mathbf{H}\mathbf{H}^H\} \quad (8)$$

Because each node includes vertically polarized wire dipole antennas and the scattering environment is represented by circles of uniformly distributed scattering objects surrounding nodes as shown in Fig.1, the spatial correlation matrix elements can be obtained using the Clark's model as given by.

$$\rho_{l,m}^{R(T)} = J_0(\kappa d_{l,m}) \quad (9)$$

where $J_0$ stands for the zero-order Bessel function, $\kappa$ is a wave number and $d_{lm}$ is the distance between elements *l* and *m* of the uniform array antenna.

The correlation matrices $R_T$ and $R_R$ can be generated using (9) as,

$$R_{R(T)} = \begin{bmatrix} \rho_{1,1}^{R(T)} & \cdots & \rho_{1,M_{rt}}^{R(T)} \\ \vdots & \ddots & \vdots \\ \rho_{M_{rt},j}^{R(T)} & \cdots & \rho_{M_{rt},M_{rt}}^{R(T)} \end{bmatrix} \quad (10)$$

Having defined $R_T$ and $R_R$, the channel matrix for each pair of nodes (**H**) can be calculated using Equation (7).

## 2.3. Channel Model Including Mutual Coupling

The mutual coupling in an array of collinear side-by-side wire dipoles can be modeled using the theory described in [16]. Assuming the array is formed by $M_{rt}$ wire dipoles, the mutual matrix can be calculated using the following expression

$$\boldsymbol{C} = (Z_A + Z_T)(\mathbf{Z} + Z_T \boldsymbol{I}_{M_{rt}})^{-1} \quad (11)$$

where **Z** is the impedance matrix, $Z_A$ is the element input impedance in isolation, e.g. when the wire dipole is λ/2, its value is $Z_A = 73 + j42.5[\Omega]$; $Z_T$ is impedance of the receiver chosen as the complex conjugate of $Z_A$ to obtain the impedance match for best power transfer.

The impedance matrix **Z** is given by

$$\mathbf{Z} = \begin{bmatrix} Z_A + Z_T & Z_{12} & \cdots & Z_{1M_{rt}} \\ Z_{21} & Z_A + Z_T & \cdots & Z_{2M_{rt}} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{M_{rt}1} & Z_{M_{rt}2} & \cdots & Z_A + Z_T \end{bmatrix} \quad (12)$$

Note that this expression provides the circuit representation for mutual coupling in array antennas. It is valid for *single mode* antennas. Wire dipoles fall into this category.

For the side-by-side configuration of dipoles having length $l$ equal to 0.5λ, the expressions for $\{Z_{mn}\}$ can be adapted from [16,17] and are rewritten here as,

$$\mathbf{Z}_{mn} = \begin{cases} 30[0.5772 + \ln(2\kappa l) - C_i(2\kappa l)] \\ \quad + j[30S_i(2\kappa l)], \quad m = n \\ 30[2C_i(u_0) - C_i(u_1) - C_i(u_2)] \\ \quad - j[30(2S_i(u_0) - S_i(u_1) - S_i(u_2))], m \neq n \end{cases} \tag{13}$$

where $\kappa$ is the wave number equal to $2\pi/\lambda$,

$$\begin{aligned} u_0 &= \kappa d_h, \\ u_1 &= \kappa(\sqrt{d_h{}^2 + l^2} + l) \\ u_2 &= \kappa(\sqrt{d_h{}^2 + l^2} - l) \end{aligned} \tag{14}$$

$d_h$ is the horizontal distance between the two dipole antenna elements. Ci(u) and Si(u) are the cosine and sine integrals, respectively. They are given as,

$$\begin{aligned} C_i(u) &= \int_{\infty}^{u} (\cos(x)/x)dx \\ S_i(u) &= \int_{0}^{\infty} (\sin(x)/x)dx \end{aligned} \tag{15}$$

The expression for the coupling matrix (11) can be used both at the transmitting and receiving nodes to modify the channel matrices and the correlation matrices. When the mutual coupling is included, the channels matrices **H** shown in (7) has to be modified to the new form given by

$$\mathbf{H}' = C_R \mathbf{H} C_T \tag{16}$$

where the mutual coupling matrices are calculated using (11).

Similarly, the receiving and transmitting correlation matrices are modified using

$$R_{Rmu} = C_R R_R^{\frac{1}{2}} \tag{17}$$

$$R_{Tmu} = R_T^{\frac{1}{2}} C_T \tag{18}$$

## 3. Capacity of MIMO Ad Hoc Network

### 3.1. Channel Capacity between Individual Nodes

Having derived the channel matrices between the individual notes for the cases without and with mutual coupling, the next step is to obtain the channel capacities. The mutual information of the channel formed between node $i$ and the desired transmitting node $j$ of MIMO ad-hoc network can be obtained in a similar way as for the case of a multi-access MIMO system [15] and is given as:

$$\begin{aligned} & I(y_i; s_j \mid s_{k,k\neq j}) \\ & \leq I(y_i; s_j \mid s_{k,k\neq j}, \mathbf{H}_{ij}, \mathbf{H}_{ik}) \\ & = h(y_i \mid s_{k,k\neq j}, \mathbf{H}_{ij}, \mathbf{H}_{ik}) - h(y_i \mid s_{k,k\neq j}, s_j, \mathbf{H}_{ij}, \mathbf{H}_{ik}) \\ & = h(y_i \mid s_{k,k\neq j}, \mathbf{H}_{ij}, \mathbf{H}_{ik}) - h(N_i) \end{aligned} \tag{19}$$

The upper bound for capacity can be obtained from the following,

$$\begin{aligned} & I(y_i; s_j \mid s_{k,k\neq j}) \\ & \leq E\{\log_2 \det[\pi e(\mathbf{H}_{ij} Q_j \mathbf{H}_{ij}{}^H + R_{N_i})] - \log_2 \det(\pi e R_{N_i})\} \\ & = E\{\log_2 \det(I_{M_{rt}} + \frac{R_{N_I}^{-1} \mathbf{H}_{ij} Q_j \mathbf{H}_{ij}{}^H}{\sigma_n^2})]\} \\ & = E\{\log_2 \det(I_{M_{rt}} + \frac{Q_j}{\sigma_n^2}(R_{N_i}^{-\frac{1}{2}} \mathbf{H}_{ij})(R_{N_i}^{-\frac{1}{2}} \mathbf{H}_{ij})^H)]\} \end{aligned} \tag{20}$$

where $\sigma_n{}^2$ is the noise power.

When both the Kronecker representation (7) and the mutual coupling (11) are included in the MIMO channel model, the expression (20) for the upper bound of mutual information of the channel formed between node $i$ and the desired transmitting node $j$ of MIMO ad-hoc network is modified to:

$$\begin{aligned} & I(y_i; s_j \mid s_{ik}) = \\ & \quad E\{\log_2 \det[I_{M_{rt}} \\ & \qquad + \frac{Q_j}{\sigma_n^2}(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^j G_{H_{ij}} R_{Tmu}^j)(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^j G_{H_{ij}} R_{Tmu}^j)^H]\} \end{aligned} \tag{21}$$

The correlation of IPN with mutual coupling is given by (22),

$$R_{N_i}^{mu} = \frac{P_I}{N} \sum_{i=1}^{N-1} (R_{Rmu}^{ik} G_{H_{ik}} R_{Tmu}^{ik})(R_{Rmu}^{ik} G_{H_{ik}} R_{Tmu}^{ik})^H + \sigma_n^2 I_{M_{rt}} \tag{22}$$

The expression for the upper bound of mutual information (21) is given for a specified signal transmission scheme as described by the data covariance matrix $Q_j = E\{s_j s_j^H\}$.

The mutual information can be maximized by optimizing the signal transmission scheme subject to $tr\{Q_j\} = P_j$. As a a result, the capacity between the $i$-th node and the desired transmitting node for the optimized signal transmission scheme is given by

$$\begin{aligned} C_i = & \max_{tr(Q_j/\sigma_n^2) \leq P_j/\sigma_n^2} E\{\log_2 \det[I_{M_{rt}} \\ & + \frac{Q_j}{\sigma_n^2}(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij})(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij})^H]\} \end{aligned} \tag{23}$$

### 3.2. Capacity of Non-Cooperative Network

The capacity of ad hoc network that uses the optimized transmission scheme between individual nodes can be

written as the sum of individual optimized link capacities $C_i$ and therefore can be expressed as,

$$C = \sum_{i=1}^{N} C_i = \sum_{i=1}^{N} E\{\log_2 \det[I_{M_{rt}} + \frac{Q_j}{\sigma_n^2}(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij})(R_{N_i}^{-\frac{1}{2}} R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij})^H]\} \quad (24)$$

### 3.3. Capacity of Cooperative Network

The capacity of ad hoc network can be further improved by assuming cooperation between individual nodes. To accomplish this, first the desired transmitting node has to have the knowledge of channel state information (CSI) from all the remaining nodes of the network. This means that all of the complex channel matrices $\mathbf{H}_{ij}$ are perfectly known to the transmitting nodes. In practice, the required CSI is first acquired by receivers by using the training sequences between the pairs of nodes and by applying the channel estimation at receiver. Next, it is passed to transmitters using the feedback loops between the receivers and the transmitters.

To make the ad hoc network fully cooperative another assumption is made that each of the $N$ nodes not only feeds information about the channel properties but also provides the information about the interference to the desired transmitting node. Under these conditions, the desired transmitting node applies $N$ different signal transmission schemes to $N$ receiving nodes to maximize the mutual information (channel capacity) to each node. This strategy leads to optimization of the overall capacity of ad hoc network.

In order to obtain the capacity of the cooperative network, first we introduce the combined channel correlation in terms of channel $\mathbf{H}_{ij}$ as,

$$\begin{aligned} R_{comb} &= \mathbf{H}_{ij}{}^H R_{N_I}^{-1} \mathbf{H}_{ij} \\ &= (R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij})^H R_{N_i}^{-1} (R_{Rmu}^{ij} G_{H_{ij}} R_{Tmu}^{ij}) \end{aligned} \quad (25)$$

In order to optimize the channel capacity between node $i$ and the desired transmitting node $j$ a water-filling algorithm can be applied. Using this algorithm, the optimized channel capacity can be expressed as:

$$C_i^{opt} = \sum_{m=1}^{M_{rt}} \log_2(1 + \frac{p_m^{ij}}{\sigma_n^2} \lambda_m^{ij}) \quad (26)$$

in which $p_m{}^{ij}$ is the optimal transmitted power from the desired transmitting node $j$ to node $i$, which is the subject to the condition

$$P_j = \sum_{m=1}^{M_{rt}} p_m^{ij} \quad (27)$$

$\lambda_m{}^{ij}$ is the $m$-th eigenvalue of combined channel correlation $R_{comb}$ given by (25).

The optimal transmitted power at the $m$-th antenna is obtained using the following equation,

$$\begin{aligned} p_{ij} &= (\mu_{ij} - \frac{1}{\lambda_m^{ij}})^+ \\ subject \quad to: \sum_{m=1}^{M_{rt}} p_m^{ij} &= P_{ij} \end{aligned} \quad (28)$$

in which $(x)^+=\max(0, x)$ and $\mu$ is chosen to obey the power constraint. The capacity of the cooperative ad hoc network with the optimized signal transmission scheme is given as,

$$C^{opt} = \sum_{i=1}^{N} C_i^{opt} = \sum_{i=1}^{N} \sum_{m=1}^{M_{rt}} \log_2(1 + \frac{p_m^{ij}}{\sigma_n^2} \lambda_m^{ij}) \quad (29)$$

## 4. Simulation Results

In this section, computer simulations are performed to investigate the effects of spatial correlation and mutual coupling on capacity of ad hoc network. In the undertaken simulations, each node of ad hoc network is assumed to be equipped with a 4-element uniform array antenna. The elements are assumed to be wire dipoles having length of 0.5λ, where λ is the carrier wavelength. The transmit/receive spatial correlation matrices for each pair of nodes are obtained using Equations (9) and (10) as presented in Subsection 2.2. The mutual coupling matrices are obtained from expressions (17) and (18) as given in Subsection 4.3.

Figure 2 presents the ad hoc network capacity in a 2-dimentional (2D) manner. The X and Y axis represent the capacity as an empirical distribution function (EDF), which is a cumulative probability *1/M* at each of the *M* numbers in a sample,

$$F_M(x) = \frac{X_i \le x}{M} = \frac{1}{M} \sum_{i=1}^{M} I(X_i \le x), \; i = 1, ..., M \quad (35)$$

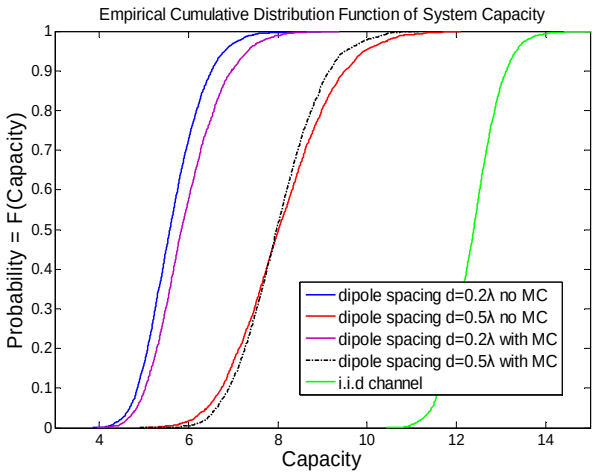



**Figure 2. The empirical distribution function (EDF) of capacity of single user vs multiple user MIMO system.**

where $X_i$ is the *i*-th element in the sample and *I(A)* is the indicator of event *A*. The X-axis represents the capacity while Y-axis indicates probability.

It can be seen in Figure 2 that when the spatial correlation exists, the capacity of ad hoc network is lower than when an ideal independent identical distribution (i.i.d) channel is assumed. This confirms the findings already obtained for single-user and multi-access MIMO systems reported in [8].

The presented results also show that when the dipole antenna spacing is small and equal to 0.2λ, the capacity with mutual coupling effect is larger compared to the one when mutual coupling is neglected. When the spacing is increased to 0.5λ the capacity curves, for with and without mutual coupling effect, are getting very close to each other. The gap between these curves is getting smaller.

The results obtained for the 4-element linear wire dipole arrays indicate that at the element spacing of 0.2λ, mutual coupling decreases spatial correlation level and improves capacity. When the spacing becomes larger, the mutual coupling effect becomes less pronounced and there is not much difference in capacity results when the mutual coupling is neglected or taken into consideration.

Figure 3 presents the ad hoc network capacity in a 3-dimentional (3D) manner. The X and Y axes represent the capacity as an empirical distribution function (EDF), similarly as in Figure 2. In comparison with Figure 2, the Z axis is added to indicate the antenna spacing between the adjacent dipoles. There are two surfaces representing the ad hoc network capacity. One surface represents the case of antenna arrays without mutual coupling while the other one stands for the case when the mutual coupling effect is included in calculations. From the results presented in Figure 3, one can see that when the dipole spacing is within 0.2λ to 0.4λ, at a fixed high probability, the capacity of ad hoc network with mutual coupling effect is higher than when mutual coupling is not taken into account. The trend becomes opposite when the spacing gets larger and is in the range of 0.4λ to 0.6λ. A cross-point occurs at the spacing equal to 0.4λ. For the element spacing of 0.6λ to 1λ, two surfaces overlap. The observed trends indicate that when the spacing is small (0.2λ to 0.4λ), the mutual coupling decreases spatial correlation and improves capacity. When the spacing is in the range of 0.4λ to 0.6λ, the mutual coupling increases the spatial correlation and decreases capacity. When the spacing exceeds 0.6λ, the mutual coupling weakly affects capacity and thus can be neglected because two sets of results without and with mutual coupling are almost identical.

Figure 4 shows the results similar to those of Figure 3 but for the case of non-optimized and optimized signal transmission schemes as given by expression (24) and (29), respectively.

The results shown in Figure 4 reveal that signal transmission optimization significantly improves ad hoc network capacity for both cases when mutual coupling is neglected or taken into account. The gap between the two capacity surfaces, with and without mutual coupling, is larger than for the non-optimized transmission scheme presented in Figure 3.

The differences between the non-optimized (expression (24))

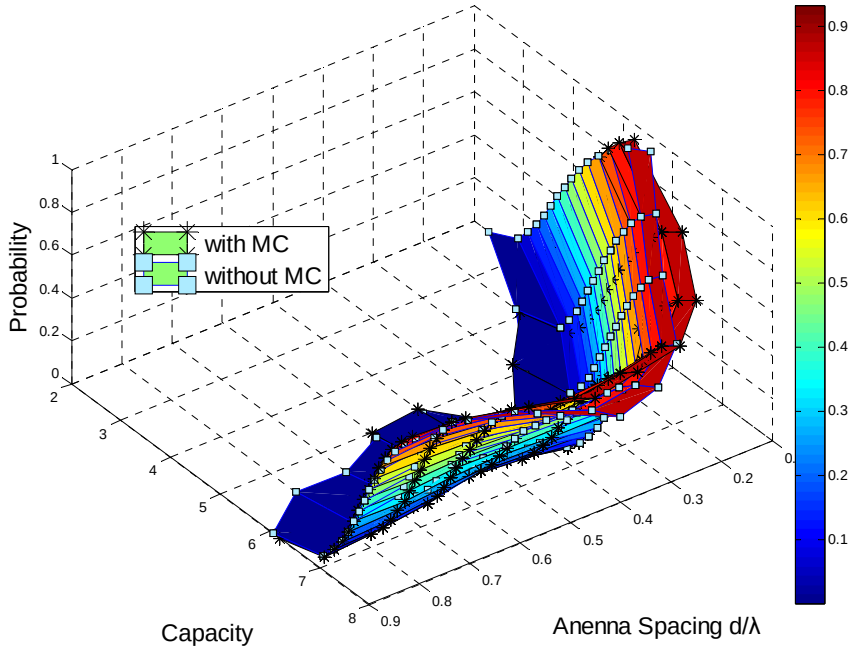



**Figure 3. The empirical distribution function (EDF) of capacity vs Antenna spacing with and without mutual coupling (MC)**

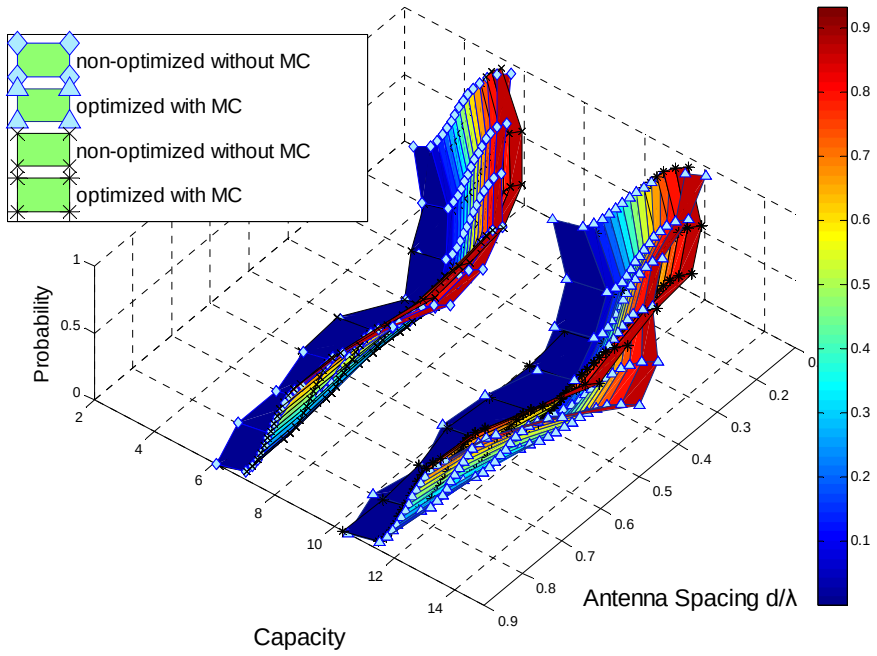
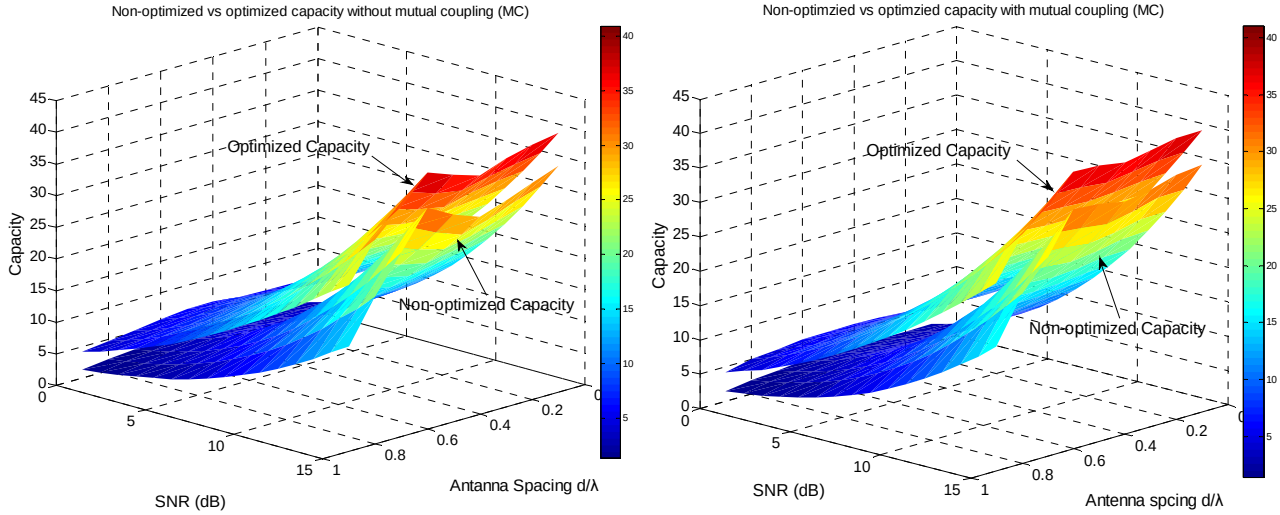



**Figure 4. Non-optimized and optimized capacity vs Antenna spacing with and without mutual coupling (MC)**




**A. Non-optimized vs optimized capacity without MC** **B. Non-optimized vs optimized capacity with MC**

**Figure 5. Non-optimized vs optimized ergodic capacity with and without mutual coupling.**

and optimized (expression (29)) signal transmission cases can also be investigated in terms of ergodic capacity.

Figure 5 presents in the 3D manner the results for ergodic capacity versus antenna spacing (from 0.1λ to 1λ) and SNR (from 0 to 15dB) for non-optimized as given by expression (29) and optimized as given by expression (24) signal transmission schemes.

Figure 5A presents the results when the mutual coupling effect is neglected, while Figure 5B shows the results when the mutual coupling effect is taken into account. One can see from the two Figures that irrespective of including or neglecting mutual coupling, the capacity for the optimized signal transmission scheme is higher than for the non-optimized case.

Figure 6 is another representation of results shown in Figure 5 when the signal transmission scheme is non-optimized.

Figure 6A presents the capacity without mutual coupling while Figure 6B presents the capacity when the mutual coupling effect is taken into account. One can see at spacing from 0.1λ to 0.4λ, that the capacity under mutual

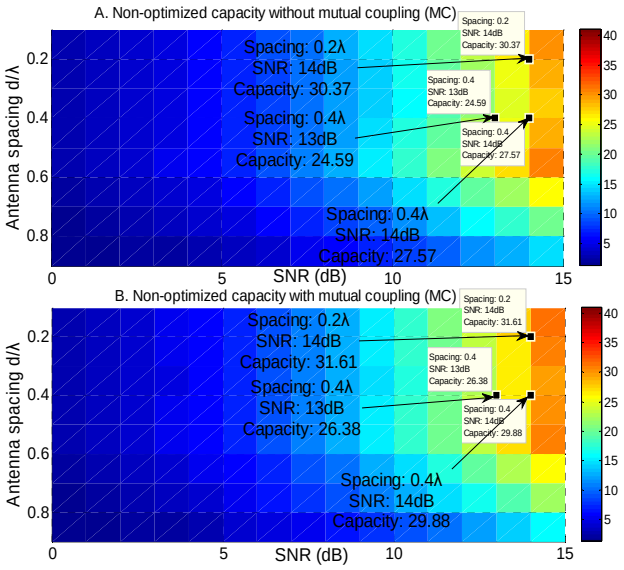



**Figure 6. 2-D view of non-optimized ergodic capacity without (A) and with (B) mutual coupling 2-D view.**

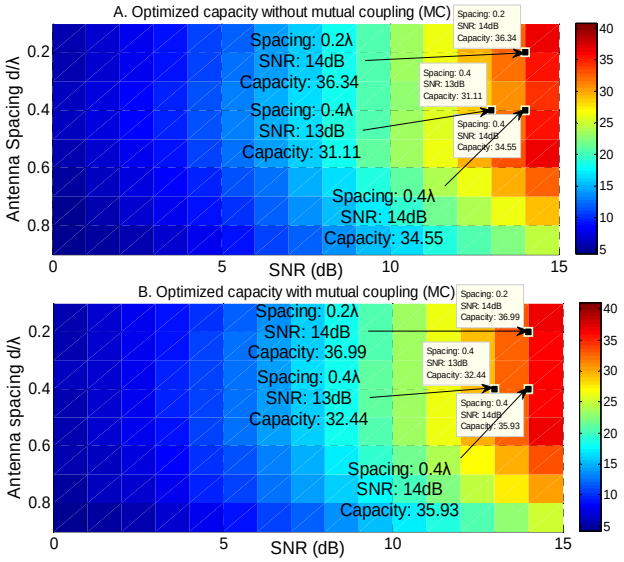



**Figure 7. Optimized ergodic capacity without (A) and with (B) mutual coupling (MC) 2-D view.**

coupling effect is higher than the one with no mutual coupling consideration. This property is more apparent at higher values of SNR.

Similarly, Figure 7 provides another representation of results shown in Figure 5 for the case when the signal transmission scheme is optimized. Figure 7A presents the capacity when mutual coupling is neglected while Figure 7B presents the capacity with the mutual coupling effect taken into account. One can see that at spacing between 0.1λ to 0.4λ the capacity under mutual coupling effect is higher than the one with no mutual coupling consideration. However, when the spacing is larger then the cross point of 0.4λ, the effect of mutual coupling is unnoticeable.

## 5. Conclusions

This paper has reported on investigations into the capacity of narrowband MIMO ad hoc network, in which nodes are equipped with multiple element antennas in the form of wire dipoles. The investigations have included the effect of mutual coupling in addition to spatial correlation that is present in the nodes array antennas. The spatial correlation has been taken into account using the Kronecker representation of the channel. Mutual coupling has been included using the closed-form expressions for impedance matrices of parallel side-to-side dipoles. Two cases of non-optimized and optimized signal transmission scheme have been considered. In the optimized signal transmission scheme the cooperative ad hoc network has been assumed, in which desired CSI and interference CSI are available at all the transmitting nodes. The computer simulations have been carried out for the case when the nodes are equipped with four- element uniform half-wave dipole arrays surrounded by circles of uniformly distributed scattering objects. The obtained simulation results for small size array antennas (of 4 elements) have shown that the spatial correlation decreases the capacity. At the antenna (dipole) spacing between 0.1λ to 0.4λ, mutual coupling decreases the spatial correlation and helps to improve the capacity of MIMO ad hoc network. When the spacing is in the range of 0.4λ to 0.6λ or exceeds 0.6λ the capacity is almost the same when the mutual coupling is taken into account or neglected. Assuming that the ad hoc network is cooperative, it has been shown that its capacity can be significantly improved by applying the optimized signal transmission scheme. The significant improvement has been demonstrated irrespectively whether the mutual coupling effects are neglected or taken into account.

## 6. References

[1] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multielement antennas,” Bell Labs Technical Journal, Vol. 1, pp. 41–59, 1985.

[2] G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” Wireless Personal Communications, Vol. 6, pp. 311–335, 1998.

[3] E. Telatar, “Capacity of multi-antenna Gaussian channels,” European Transactions on Telecommunications, Vol. 10, No. 6, pp. 585–596, November 1999.

[4] B. Chen and M. J. Gans, “MIMO communications in ad hoc networks,” IEEE Transactions on Signal Processing, Vol. 54, No. 7, Juy 2006.

[5] K. Fakih, J. Diouris, and G. Andrieux, “On the capacity optimization for MIMO ad hoc networks,” IEEE VTC

Spring, Singapore, pp. 1087–1091, May, 2008.

[6] D. Shiu, J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielemnt antenna system," IEEE Transactions on Communication, Vol. 48, No. 3, March 2000.

[7] X. Liu and M. E. Bialkowski, "Effect of channel estimation on capacity of MIMO system employing circular or linear receiving array antennas," International Journal of Electronics, Communications and Computer Engineering, Vol. 1, No. 1, pp. 61–71, 2009.

[8] X. Liu, F. Wang, K. Bialkowski, and M. E. Bialkowski, "On the capacity of MIMO system operating in multi-access scenario," to appear in Proceedings of APMC09 Singapore, pp. 7010, December, 2009.

[9] T. Svantesson and A. Ranheim, "Mutual coupling effects on the capacity of multi-element antenna systems," Proceedings of ICASSP01, pp. 2485–2488, April 2001.

[10] J. W. Wallace and M. A. Jensen, "The capacity of MIMO wireless systems with mutual coupling," IEEE VTC02, Vancouver, Canada, pp. 696–700, 2002.

[11] J. Wallace and M. Jensen, "Mutual coupling in MIMO wireless systems: A rigorous network theory analysis," IEEE Transactions on Wireless Communications, pp. 1317–1325, March 2004.

[12] P. N. Fletcher, M. Dean, and A. R. Nix, "Mutual coupling in multielement array antennas and its influence on MIMO channel capacity," IEEE Electron Letters, No. 39, pp. 342–344, 2003.

[13] E. G. Larsson and P. Stoica, "Space-time block coding for wireless communication," Cambridge University Press, 2003.

[14] C. N. Chuah, D. N. C. Tse, and J. M. Kahn, "Capacity scaling in MIMO wireless systems under correlated fading," IEEE Transactions on Information Theory, Vol. 48, pp. 637–650, March 2002.

[15] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel," IEEE Transactions on Information Theory, Vol. 46, No. 3, pp. 933–946, May 2000.

[16] S. Durrani and M. E. Bialkowski, "Effect of mutual coupling on the interference rejection capabilities of linear and circular arrays in CDMA systems," IEEE Transactions on Antennas and Propagation, Vol. 52, No. 4, pp. 1130–1134, April 2004.

[17] M. E. Bialkowski, P. Uthansakul, K. Bialkowski, and S. Durrani, "Investigating the performance of MIMO systems from an electromagnetic perspective," Microwave and Optical Technology Letters, Vol. 48, No. 7, pp. 1233–1238, July 2006.

*Wireless Sensor Network*, **2009, 1, 434-445**
doi:10.4236/wsn.2009.15052 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Localization of a Target with Three Degrees of Freedom Using a Low Cost Wireless Infrared Sensor Network

**Nikos PETRELLIS, Fotios GIOULEKAS, Michael BIRBAS, John KIKIDIS**
*Analogies S.A., Patras Science Park, Platani-Rio, Patras, Greece*
*Email*: {*nikos.petrellis, fotios.gioulekas, michael.birbas, john.kikidis*}*@analogies.eu*
*Received August* 10, 2009; *revised September* 14, 2009; *accepted September* 15, 2009

## Abstract

The estimation of the position of a mobile target on a plane as well as its orientation is an important aspect for many applications. The indoor or outdoor localization of such a target has been widely addressed in the literature but if a third degree of freedom like rotation has to be also taken into consideration the difficulty in estimating the target position and orientation is significantly increased. A network consisting of only a small number of low cost infrared transmitters/receivers is used in this paper to estimate the position of a mobile target on a plane as well as its draft orientation with an angular step of $45^o$ or less. The distance and orientation estimation is based on the success rate that infrared patterns are retrieved at the target. This success rate parameter is calculated by simple ultra low cost microcontrollers. The architectural complexity and cost of the overall localization system is significantly lower than other approaches without sacrificing speed and accuracy. An error correction scheme like Turbo decoding is applied in order to increase the reliability and stability of the results by correcting burst errors introduced by real time noise.

**Keywords:** Position Estimation, Localization, Infrared Sensor Networks, Turbo Codes

## 1. Introduction

The applications where indoor localization is important concern robotics, automation, virtual reality and pervasive computing environments. Although knowing the position of a moving target on a plane is very important it would be valuable for a large number of applications if the orientation of the target could also be estimated. For example, a system guiding a robot, a handicapped person or a person wandering in a virtual museum, should also be aware of the draft at least orientation of this robot or person. Although several approaches have been proposed for the position estimation of targets with 2 degrees of freedom, there are not many general solutions for targets with 3 degrees of freedom.

A rather complicated method used by the robots in order to familiarize with an unknown environment is based on processing the images captured by cameras that are mounted on the robots in order to recognize landmarks and their distance [1]. Conclusions extracted from image processing are combined with other localization approaches [2] in autonomous robotics applications. Stochastic processing can also help in this case for the validation of an estimated position. It is obvious that image processing requires powerful and complicated processing units leading to rather expensive localization solutions.

Measuring the time of the flight of a reflected impulse wave or the strength of a signal are ordinary methods used for the indoor or even outdoor localization. Optical or Laser beams can be used to scan the surrounding area detecting the distance of walls or other obstacles. The cost of this solution is higher since very short time intervals have to be measured with high precision [1,3]. Ultrasonic signals can offer a lower cost alternative to this approach since the sonar waves travel with much lower speed than light [4,5]. A drawback concerning the use of ultrasounds is that this type of signal is not directional enough and it is difficult to isolate the sonar transmitter from the receiver in order to ensure that only the reflected signal will be taken into consideration during the distance measurement. Localization systems based on ultrasonic waves often estimate distances by measuring phase shifts of the original and the reflected signal, which also requires processing of high precision and speed.

The signal strength of multiple transmitters surrounding the target can also provide an indication about the target position using a triangulation method [6,7]. This technique has already been adopted in cellular telephony, Wireless Local Access Network (WLAN) or Bluetooth applications.

Magnetic fields have been adopted for the accurate non-contact control of the position and orientation of

tools and medical instruments in short distances (up to a few centimeters) [8,9]; Distance estimation of up to 10m using magnetic fields has been reported in [10].

Passive infrared sensors are used by mobile targets in order to avoid obstacles while active infrared tranceivers are employed similarly with laser or ultrasonic beams to detect the distance of the target from walls, obstacles etc [11,12]. Another interesting use of infrared light is the profiling of the surface of an object by recognizing its texture (used for distinguishing metal from plastic surfaces etc) [13,14].

Low cost infrared tranceivers have been used in order to estimate the position of a moving target with 2 degrees of freedom with an absolute error of less than 20cm in [15–18]. The reception quality of the digital patterns that are sent by at least two transmitters that are placed around the covered area is exploited for the estimation of the position of a receiver that can move on a plane without rotating. A calibration procedure before real time operation is required in order to familiarize the target with the area. During this calibration stage, the target visits predetermined positions and enumerates the recognized patterns in a period of time in order to estimate the success rate of the reception. A position identity can be formed by the success rates of the various pattern types that are employed. During real time operation, a position identity is constructed in the same way for the current target position and is compared to the identities that were estimated during the calibration stage. The closer position that was visited during the calibration is selected. A regression technique can help reaching a more accurate estimation of the real target position.

Instant noise that has not been taken into consideration during the calibration stage can reduce the speed and accuracy of the position estimation. Several rules can be applied to validate the results of the position estimation procedure [16]. If a positive estimation is characterized by these rules as unacceptable, it is discarded and a new estimation is initiated. Error Correction techniques have also been employed by the authors [17] in order to reduce the effect of instant noise and speed up the estimation procedure. The interleaving process employed in Turbo decoding can minimize the effect of the burst errors caused by the instant noise [19,20]. When the attenuated signal is received at the target, it is corrected by a decoder that can be implemented either in software or by dedicated hardware [21,22]. Since our localization approach is based on the quality of the received signal, the intension is to minimize the effect of the burst errors caused by instant noise through the selected Error Correcting method rather than fully recover the initial patterns. A description of how an additional sensor at the side of the target can be used to provide an orientation indication has been given in [18].

In the present work we modify some architectural features of the infrared transmitters and receivers in order to obtain a draft indication of the target orientation in $45^o$ angular steps i.e, 8 potential directions. It will also be explained how narrower angular steps and consequently more accurate orientation estimation can been supported. The position and orientation estimation in the architecture that will be presented here, takes place in two phases: the draft orientation is initially estimated by classifying the retrieved success rates as strong or weak and then the target position coordinates related to an infrared transmitter are estimated using the exact success rate values.

The designed localization system covers a large area with a small number of low cost transmitters/receivers since it is based on a simple digital processing method that merely counts patterns. A very good trade off between cost, area covered, accuracy and speed is achieved.

The architecture of the infrared transmitting and receiving devices along with the estimation methods used in our previous approaches are described in Section 2. The topology used in the present work and the position/orientation estimation method is described in Section 3. Finally, the experimental results will be discussed in Section 4.

## 2. System Architecture

### 2.1. Infrared Transmitter (Irtx)

The architecture of the infrared transmitters (IRTX) that are used in the proposed localization system is presented in Figure 1. A processing unit generates both the patterns and the carrier that are mixed and amplified before they are transmitted by one or more infrared emitting diodes. In [15,16,18] the infrared patterns transmitted by an IRTX device had the form that appears in Figure 2a. Each pattern type consists of a number ($i$) of pulses that have a duration inversely proportional to the number $i$.

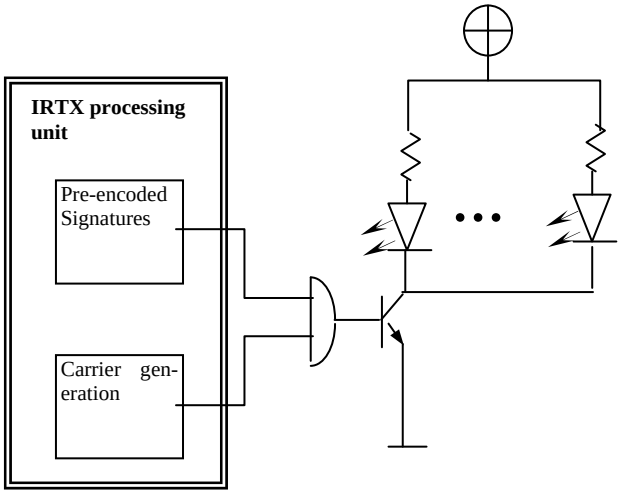



**Figure 1. Architecture of an IRTX device.**

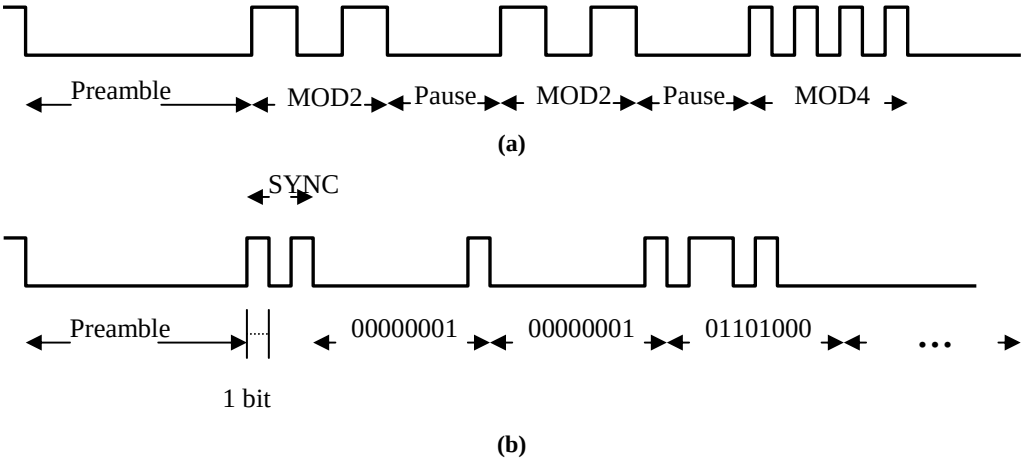



**Figure 2. Infrared pattern formats consisting of a preamble followed by several MODi patterns (a) or signatures (b).**

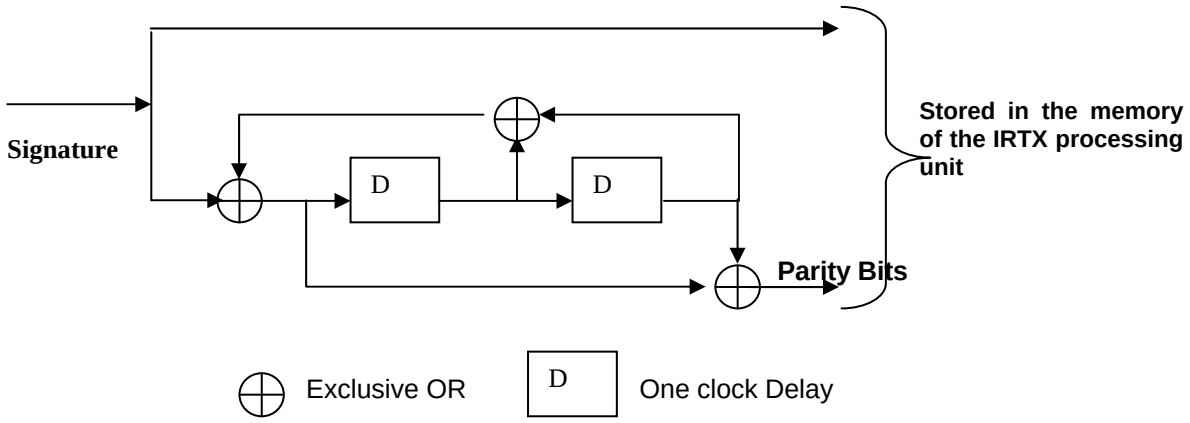



**Figure 3. Recursive systematic convolutional (RSC) encoding used by the IRTX devices.**

Such a pattern is designated as MOD*i*. This type of format was chosen because patterns with higher number of short pulses are recognized with lower success rate than patterns consisting of lower number of long pulses. Moreover, the receiver may simply count rising or falling edges between long pause intervals in order to recognize a pattern type. Each IRTX device initially transmits a preamble and then, a specific number of codes from each pattern type. After the transmission of all the supported pattern types, this procedure is repeated by transmitting a new preamble.

The digital pattern signal is mixed with a carrier before driving the infrared emitting diodes. The carrier is used in order to avoid interference from other infrared sources like sunlight. More than one infrared emitting diodes can be connected in parallel and placed in a circular arrangement in order to cover a wider area.

Using a pattern format that consists of a constant number of bits that have equal duration as shown in Figure 2b in order to employ an error correcting method at the receiver can lead to faster implementations with higher precision results as discussed in [17]. Using this type of patterns, the receiver sensors have to take samples at regular time intervals instead of merely counting rising or falling edges. A preamble is initially transmitted followed by a synchronization sequence (SYNC). The preamble is a long pause period while the SYNC is a small sequence of identical pulses used by the receiver for synchronization. Then, all the supported pattern codes are transmitted. Patterns of this type will be called henceforth "signatures" since they identify their transmitter. Although multiple signatures with smaller length can be used, a single signature with 160-bit length can lead to a better trade off between speed and accuracy as discussed in [17].

The signatures are transmitted along with their parity bits that have been generated through a proper encoding. The receiver is aware of the supported patterns or signatures and simply enumerates the recognized ones i.e., those that have not been distorted. For this reason, pre-

encoded signatures and parity bits are stored in the memory of the IRTX processing unit. The Recursive Systematic Convolutional (RSC) encoding described by Figure 3 is used off line to generate the parity bits that correspond to the supported signatures. The polynomial describing this encoding is:

$$(1+ D^2)/(1+D+D^2) \quad (1)$$

## 2.2. Infrared Receiver (IRRX)

The architecture of the infrared receiver devices (IRRX) used, is shown in Figure 4. The infrared sensors are connected to a bandpass filter to select only the infrared signals modulated at the specific carrier frequency used. Then, an integrator rejects the carrier and the pure pattern signal is recognized by the processing unit of the receiver. If the target moves with 2 degrees of freedom, two infrared sensors ($IRRX_A$ and $IRRX_B$) are adequate [15–17]. If the target is also allowed to rotate a third IRRX sensor may be helpful as discussed in [18]. Nevertheless, in the next section we will describe how only 2 IRRX sensors can also be used in order to get an indication of the orientation of the target if they are placed at a proper angle.

The IRRX processing unit marks the start and the end of the signature and parity bit parts. The next stage is the application of a Turbo Decoding algorithm in order to minimize the effect of the burst errors caused by instant noise that has not been taken into consideration during the calibration stage [17]. It is noticed that the aim of the applied Turbo decoding method is to smooth the error occurrence instead of correcting all the errors. More specifically, the interleaving used by the Turbo decoders, scatters the burst errors throughout the length of a signature making easier their correction. The turbo decoder can be implemented at the host computer where the application that exploits the position estimation results is running. If the IRRX processing unit is a more complicated one supporting all the applications required by the target, the Turbo decoding can be implemented in a low level language like assembly. Another option is to embed a special Turbo decoding peripheral like the one described in [21]. This option has a higher cost but would significantly speed up the decoding procedure.

## 2.3. Turbo Decoding Implementation Alternatives

The state machine and the Trellis diagram that correspond to the encoding scheme of Figure 3 are presented in Figures 5a and 5b respectively. The interleaver used is of the random type and it is shown in Figure 6 since a lower Bit Error Rate (BER) can be achieved in this way [17].

The Turbo Decoder used consists of two Soft In Soft Out (SISO) decoders each one of them operating alternatively on the un-interleaved and interleaved input signatures respectively (see Figure 7). Each SISO decoder consists of identical blocks that implement a stage of the Trellis diagram (Figure 8). A SISO decoder generates extrinsic information that is used as intrinsic input by the other SISO decoder at the following iteration. The decoding stops and the outputs are settled after a specific number of iterations.

The parameters *a_S*(*t*) and *b_S*(*t*) of Fig. 8 are branch metrics of the Trellis diagram qualifying how possible is a path leading to state *S* of the stage *t*. *Int*(*t*), *Ext*(*t*) and *U*(*t*) are the intrinsic, the extrinsic info and the output of this stage. The inputs P(Yu|Xu) and P(Yp|Xp) are the

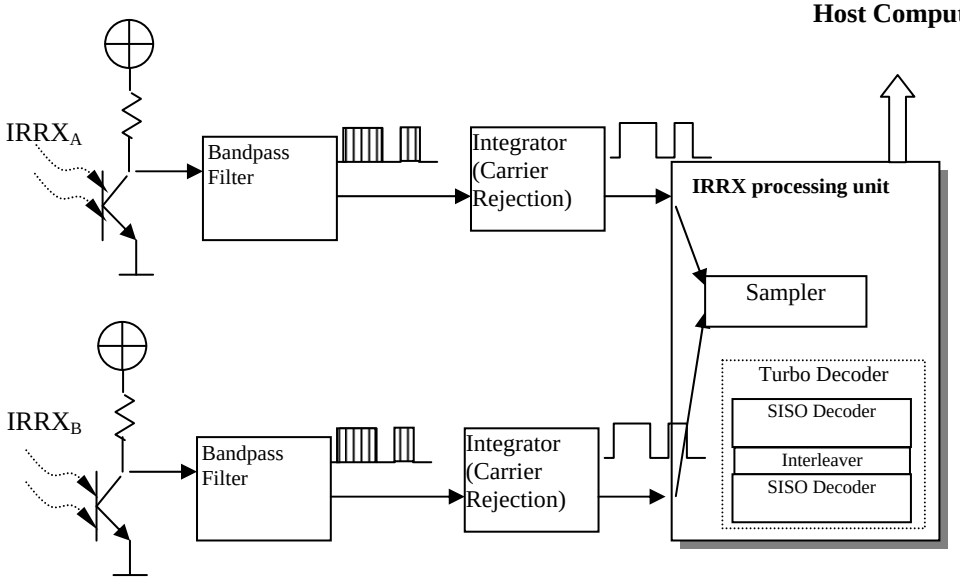
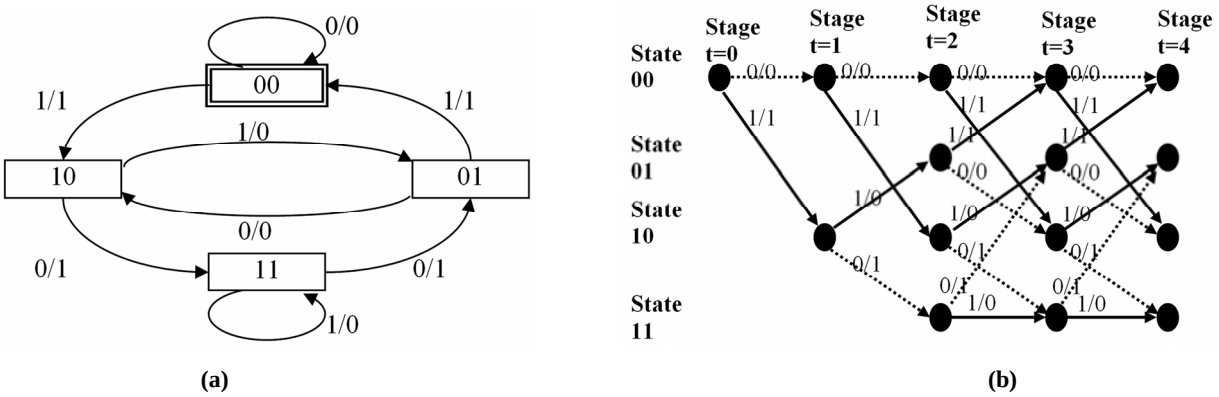



**Figure 4. Architecture of the IRRX device consisting of two sensors and a processing unit that communicates with a Host Computer.**




**(a)** **(b)**

**Figure 5. State machine (a) and Trellis diagram (b) for the RSC encoder of Figure 3.**

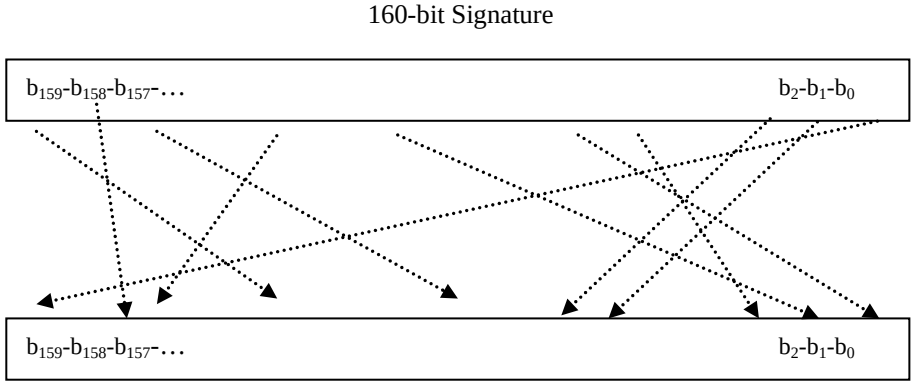



**Figure 6. The bit positions of the original signature are mapped to random positions in the interleaved one (random interleaver).**

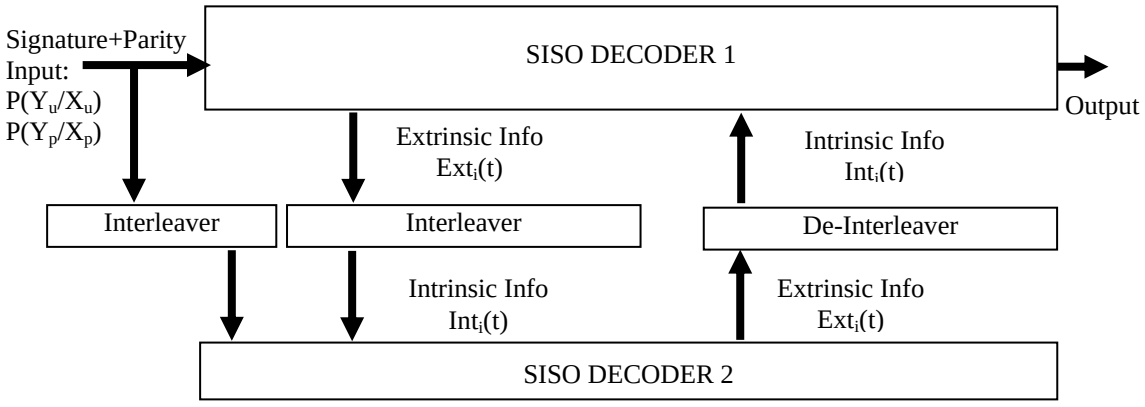



**Figure 7. Turbo decoding scheme.**

probabilities to receive symbols Yu and Yp given that the input Xu and the parity Xp has been transmitted. The lower BER is achieved if the decoding is performed by the Sum-Product Algorithm (SPA) that is described by the relations of Table 1.

The analog implementation of a Turbo Decoder presented in [21] is an example of how the decoding needed by the localization system can be implemented as a processing unit peripheral. If simpler operations than multiplication are needed in order to have a less complicated hardware implementation of the Turbo decoding, the Max-Log MAP or the Min-Sum algorithm can be employed [22]. Their relations are derived by the equations of Table 1 if a logarithmic operation is applied to

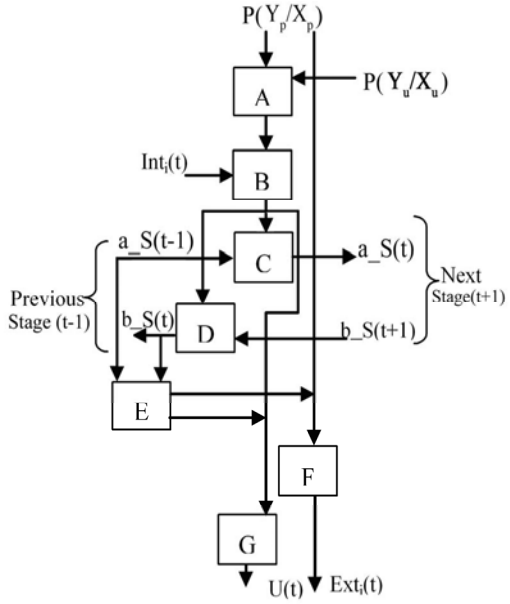



**Figure 8. Implementation of a Trellis stage.**

**Table 1. The relations of the SPA algorithm.**

| Block | Calculation |
|---|---|
| A | $\gamma'_{ij} = \Pr(y_u(t) \mid x_u(t) = i)\Pr(y_p(t) \mid x_p(t) = j)$, $i, j \in \{0,1\}$ |
| B | $\gamma_{ij} = \gamma'_{ij} Int_i(t)$, $i, j \in \{0,1\}$ |
| C | $a\_S(t) = \sum_{S'} a\_S'(t-1)\gamma_{S' \to S}$ |
| D | $b\_S'(t) = \sum_{S} b\_S(t+1)\gamma_{S' \to S}$ |
| E | $I_{ij} = \sum_{i/j, S' \to S} a\_S'(t-1) b\_S(t)$ |
| F | $Ext_i(t) = \sum_j I_{ij} \Pr(y_p(t) \mid x_p(t) = j)$, $i \in \{0,1\}$ |
| G | $U_i(t) = \sum_j I_{ij}\gamma_{ij}$, $i \in \{0,1\}$ |

both sides and the following identity is taken into consideration.

$$-\ln(e^{-x} + e^{-y}) = \min(x, y) - \ln(1 + e^{-|y-x|}) \quad (2)$$

The second term of the right side of Equation (2) is often neglected with a penalty of about 0.5dB in the achieved BER. In the present work the Turbo decoding scheme is implemented by software at the Host Computer. For this reason, SPA algorithm was chosen in order to achieve the best possible error correction.

## 2.4. Topologies for Position Estimation

In [15] the success rates of various pattern types at regular distance and angular steps was measured as shown in Figure 9. The position of the target was represented with polar coordinates related to an IRTX device. For example, Figure 10 shows the success rate curves of pattern types MOD2, MOD5, MOD6 and MOD9 at 2m distance and angular displacement ranging from –90° to +90°. If the target resides at a distance different than 2m, the success rates that are measured for MOD2, MOD5, MOD6 and MOD9 may be the $r_2$, $r_5$, $r_6$ and $r_9$ of Fig. 10, respectively that do not converge to the same angle. On the contrary, if it resides at a 2m distance and the measured success rates are $r'_2$, $r'_5$, $r'_6$ and $r'_9$, then all these success rates converge to the same angle (35° or +35°).

A second IRTX device is used to break the symmetry at the right and the left side of the reference infrared transmitter. Such curves can also be used to give an indication about the orientation of the target since the position estimation is expressed in polar coordinates (distance, angle) as described in [18].

The smooth success rate behavior designated by the curves of Figure 10 is valid only if the multiple IRTX devices do not transmit concurrently in order to avoid scrambling. This restriction slows down the localization procedure since the estimation time needed is longer in order to allow the target to receive the necessary pattern codes from all the neighboring IRTX devices.

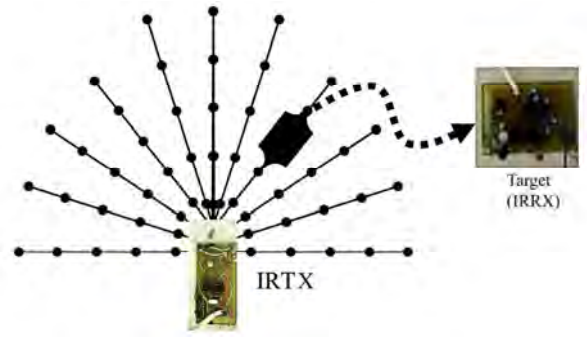



**Figure 9. Topology for polar coordinate estimation**

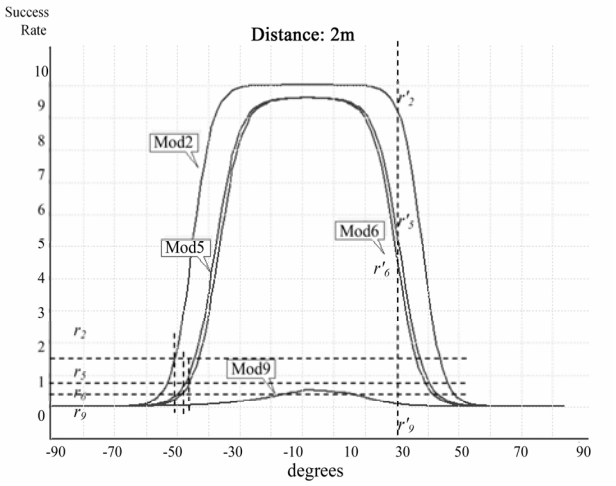



**Figure 10. Success rates at 2m distance from IRTX.**

The topology proposed at [16,17] overcomes this restriction since the target moves with 2 degrees of freedom on a virtual grid plane as shown in Figure 11. The transmitters are positioned at the borders of the covered area. During the calibration stage, the target visits the grid nodes and stores the retrieved success rates. At real time operation, the closer grid node is selected after comparing the current success rates with the stored ones. An interpolation method can give a more accurate approximation of the real target position expressed in Cartesian coordinates. A larger area can be covered if more than two IRTX devices are properly positioned. An example of how the success rate of a specific pattern type changes within a $1m^2$ area covered by 2 IRTX devices is shown in Figure 12. Comparing Figure 10 with Figure 12 we conclude that the success rates change in a more random way when the IRTX devices transmit concurrently.

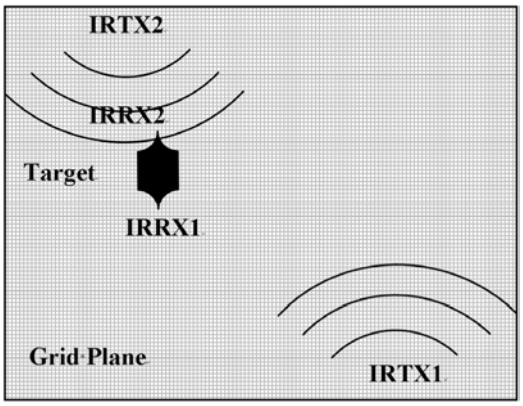



**Figure 11. Grid topology.**

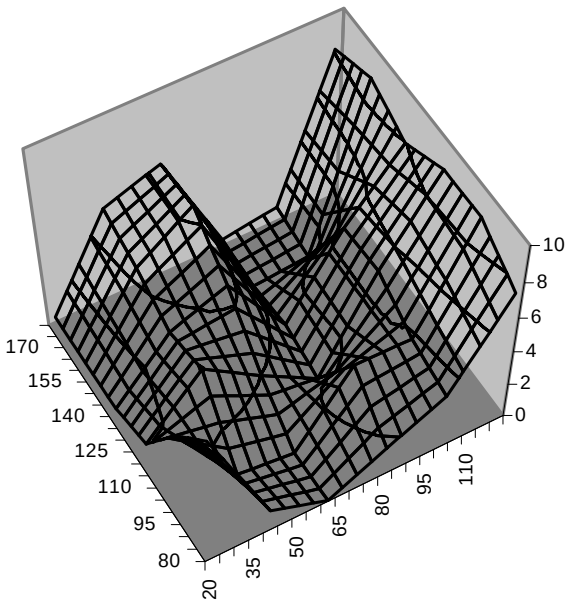



**Figure 12. Example success rate in a $1m^2$ area when the IRTX devices transmit concurrently.**

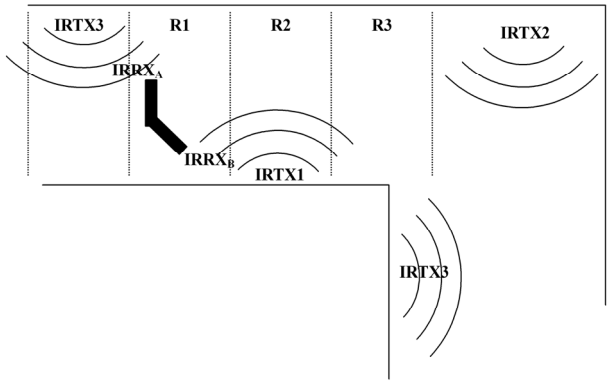



**Figure 13. Example positioning of the IRTX devices in order to estimate both distance and orientation.**

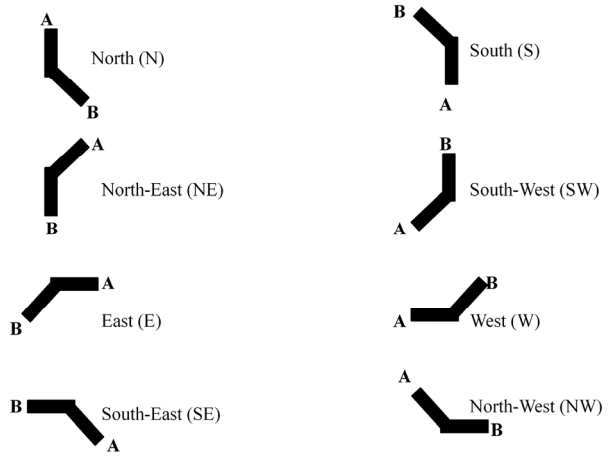



**Figure 14. Distinct orientations**

## 3. Proposed Topology and Estimation Method of the Target Position and Orientation

The detailed topology and position estimation method proposed in this paper for obtaining an orientation indication except from the Cartesian coordinates is presented in this section. The IRTX devices have the architecture shown in Figure 1 and transmit the stored, pre-encoded 160-bit signatures concurrently. They can be positioned as shown in Figure 13, in order to cover a corridor or a hall. Each IRTX device transmits a different signature. An IRTX device should transmit different signatures from its neighboring IRTX devices. Multiple IRTX devices can be used to cover the desired area transmitting alternatively only 3 types of signatures. For example, the IRTX3 device at the bottom of Figure 13 should be followed by IRTX1, IRTX2 etc in order to extend the area covered. A virtual grid is assumed to cover the area of this corridor or hall. The nodes of this grid are visited during the calibration phase by the target in order to measure the success rates at various orientations and familiarize with the area.

The two IRRX sensors that reside on the target, form a 135° angle in order to accept from a different angle the signatures sent by neighboring transmitters. This is useful in order to define unique orientation identities. The received signatures along with their parity bits are corrected by a Turbo decoder that is implemented in software at the Host computer at this prototype level. The algorithm employed by the Turbo decoder is the Sum-Product. The system can distinguish the 8 draft orientations of the target that are shown in Figure 14.

An indication about the orientation of the target can be extracted if we assume that each IRTX device covers three adjunct regions in front of it and at its left and right side. A target residing at the region in front of this specific IRTX device may receive strong or weak signal from it and a weak signal from either of its neighboring IRTX devices. If the target resides at the left of this specific IRTX device it can receive strong or weak signals from it and its neighboring IRTX device at the left. If the target resides at the right of this specific IRTX device it can receive strong or weak signals from him and his neighboring IRTX device at the right.

In order to define when a signal will be characterized as strong or weak we should define the success rate in the present architecture. As already mentioned a 160-bit signature is transmitted by each IRTX device. This signature is accompanied by a 160-bit parity that has been generated after encoding the initial signature using the RSC encoder of Figure 3. The signature is interleaved by a random interleaver and encoded by the same RSC encoder. An IRTX device transmits the signature in its original and interleaved form as well as all the parity bits i.e., a total of 640-bits are sent (Rate=1/4). The 480 redundant bits are used by the receiver in order to correct the burst errors that occurred during the transmission due to instant noise and recover the original 160-bit signature.

The original 160-bit signature is split in ten 16-bit parts consisting of different patterns of 1's and 0's in order to differentiate the difficulty in correcting each one of these parts. For example, the correction of the signature parts 0x5555 and 0xEEEE may not have the same success. The number of signature part bits that match the expected ones after the Turbo Decoding, define the success rate (SR*i*) for this signature part *i*. Its value may range from 0 to 16. The success rate of the whole signature is a complex identity consisting of the 10 individual success rates SR*i*. The Averaged Success Rate (ASR) of a signature is defined as:

$$ASR = (SR_0 + SR_1 + ... + SR_9)/10 \quad (3)$$

The success rate of more than one sample can be averaged to get a more reliable estimation. An IRTX signal is characterized weak if the ASR of its signature is less or equal to 5 and strong if it is higher than 5.

Focusing on the regions R1, R2 and R3 of Figure 13 and assuming that the target is at the specific position and has a North orientation we can state and experimentally prove that the $IRRX_A$ receives a weak signal from IRTX3 and $IRRX_B$ receives a strong signal from IRTX1. In the same way the rules listed in Table 2 can be extracted for the topology of Figure 13. The columns A and B of this table designate the strength of the signals received from each IRTX device on the sensors $IRRX_A$ and $IRRX_B$ respectively. The letters W and S mean Weak and Strong respectively, while the number following this letter corresponds to the IRTX device transmitting this signal. For example, W3 means that a weak signal is received from IRTX3.

The draft signal characterizations used in Table 2, form an orientation identity of the target. Hence, the target determines its orientation in a specific region as a first stage of the localization procedure. Unfortunately, the East orientation of R1 can be confused with the South East one of R2, as we can see from this table. Moreover, two target orientations of R2 and R3 (the ones with the shaded background) can also be confused. The higher number of orientations that can be potentially confused in this case is owed to the corner of the corridor and the IRTX2 and IRTX3 positioning. Such a problem can be resolved at a certain extent if we know some a priori restrictions (maximum/minimum speed, steering rules etc) in the move of the target as discussed in [16].

After determining the orientation of the target, a specific set of signature success rate maps like the one presented in Figure 12 are used to select the closer grid node to the target. More specifically, the success rates SR*i*

**Table 2. Orientation rules.**

| R1 | | | R2 | | | R3 | | |
|---|---|---|---|---|---|---|---|---|
| **Orientation** | **A** | **B** | **Orientation** | **A** | **B** | **Orientation** | **A** | **B** |
| N | W3 | S1 | N | W2+W3 | W1 | N | S2 | W2 |
| NE | W1+W3 | W1 | NE | W2 | S1 | NE | W2 | W3 |
| E | W1 | W1+W3 | E | W2 | W1 | E | W2 | W1+W3 |
| SE | S1 | W3 | SE | W1 | W1+W3 | SE | W2 | W1 |
| S | W1 | S3 | S | S1 | W3 | S | W3 | W2 |
| SW | W1+W3 | W3 | SW | W1 | W2+W3 | SW | W1+W3 | S2 |
| W | W3 | W1+W3 | W | W1 | W2 | W | W2+W3 | W1 |
| NW | S3 | W1 | NW | W3 | W1 | NW | W1+W2 | W2 |

retrieved at the current position are compared to the corresponding stored ones from the calibration phase in order to select the closest grid node. The comparison is based on the following equation:

$$D_j = \sum_{i=0}^{9} abs(SR_i - SR_{ij}^{'}) \quad (4)$$

The parameter $SR_i$ is the success rate of the signature part $i$ at the current target position. $SR_{ij}'$ is the success rate of the corresponding part $i$, retrieved during calibration at the node $j$ when the target had the orientation recognized at the first stage of the localization procedure. A different $D_j$ parameter is estimated for each grid node and the node that has the smallest $D_j$ value is selected as the closest one. This is the second stage of the localization procedure.

It should be emphasized that during the calibration phase, a different set of success rate maps should be stored for each orientation. For example, if a 1.8x1.8 meter region is assumed to be covered by a 30x30cm grid, then 25 grid nodes should be visited by the target without changing his orientation. A total of 8x25=200 success rate measurements should be carried out during the calibration phase to familiarize with a specific region, since 8 distinct orientations are considered.

A third localization procedure stage may follow, in which the actual target position can be further approximated by an interpolation method as described in [16]. This interpolation method assumes that the success rate parameters change linearly between neighboring grid nodes. This assumption is valid if the grid node distance is short enough (less than 30cm).

## 4. Case Study-Discussion

Considering the first stage of the localization procedure where the draft target orientation is decided, the successful estimations approach a level of 100%. Some failures may occur at positions or orientations where the measured success rates are close to the limit between the Weak and Strong signal characterization. Of course, if the orientation estimation fails, the second and third stage of the localization procedure will possibly fail too, since a wrong success rate set of maps derived from the calibration stage will be used.

If the 1st stage of the localization procedure is successful, then the expected absolute error at the position estimation that is carried out at the 2nd and 3rd localization stage should be comparable to the one achieved at [16]. The worst error in this case was equal to the grid node distance (20cm). Having in mind that a longer grid node distance (30cm) has been selected in the present work, the worst position estimation error is 30cm.

The worst error may be reduced if shorter grid node distance is selected. The 3rd localization procedure stage (interpolation) would also be more efficient in this case since the success rate changes more linearly between neighboring nodes with 20cm distance than those that have a 30cm distance. Nevertheless, a time consuming calibration stage would be necessary since the number of nodes that have to be visited is extremely higher in this case. Moreover, if a large number of grid nodes are considered, there is a higher possibility that several nodes with similar success rate identities that can be confused at real time operation leading to false position estimations will be existed.

Similar problems need to be faced if more than 8 distinct target directions are considered. For example, if we try to distinguish between 16 directions the number of grid nodes that would have to be visited during calibration would be doubled. More than two signal characterizations would be required in this case to form different orientation identities (e.g., weak, medium, strong) but if a higher number of signal characterizations are used, then it is more difficult to distinguish the limits between them and more errors will appear when determining the target orientation.

The carrier frequency used at the prototype level is a low standard frequency of 38KHz that is adopted by commercial IR remote control devices. The estimation of the target orientation and position requires more than 1 sec in this case. This time can be reduced to less than 50ms if a higher carrier frequency is used (e.g., 1MHz) [16,17]. In this case, a custom bandpass filter and carrier rejecting circuit is required at the side of the receiver while the receiver processing unit should operate at higher clock frequencies (at least 20MHz) to sample efficiently the IRRX sensors.

In Figure 15 an experimental target track in three regions (R1, R2, R3) neighboring to the IRTX1 transmitter

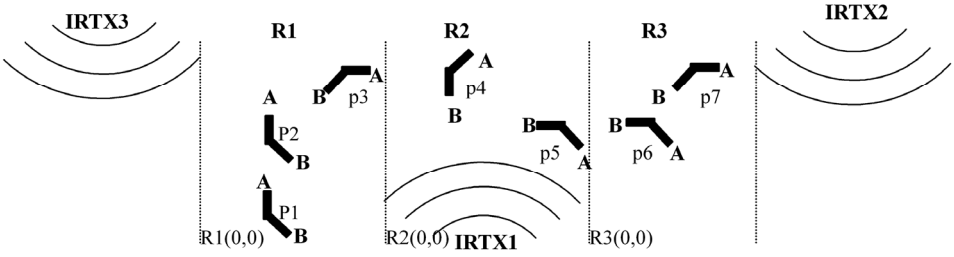



**Figure 15. Experimental target track.**

**Table 3. Experimental target localization results.**

| Posit. Ident. | Real Position | | Experimentally Estimated Position | | | | $IRRX_A$ ASR | $IRRX_B$ ASR | FEC Used/ Inter pol. |
|---|---|---|---|---|---|---|---|---|---|
| | Coordinates | Orient | Coordinates | | Orientation | | | | |
| | | | Coord. | Stab | Orient. | Stab | | | |
| p1 | R1(80cm, 30cm) | N | R1(60cm, 0) | 1/3 | N | 3/3 | 2(3) | 6(1) | Y /N |
| p2 | R1(80cm, 90cm) | N | R1(90cm, 90cm) | 2/3 | N | 3/3 | 3(3) | 10(1) | N /N |
| p3 | R1(140cm 160cm) | E | R1(150cm, 180cm) | 2/3 | E | 3/3 | 1(1) | 1(1) 1(3) | Y /N |
| p4 | R2(70cm, 160cm) | NE | R2(50cm, 150cm) | 3/3 | NE | 3/3 | 1(2) | 7(1) | N /Y |
| *p5* | *R2(160cm, 80cm)* | *SE* | *R2(120cm, 90cm)* | *1/3* | *E* | *2/3* | *2(1) 1(2)* | *2(1)* | *Y /N* |
| p6 | R3(50cm, 80cm) | SE | R3(30cm, 60cm) | 2/3 | SE | 2/3 | 1(1) 2(2) | 3(1) | Y /N |
| p7 | R3(100cm 170cm) | E | R3(120cm, 150cm) | 1/3 | E | 3/3 | 4(2) | 5(1) | Y /N |

**Table 4. Localization method comparison.**

| Method | Complexity/Cost | Range | Accuracy/ Orientation Indication |
|---|---|---|---|
| This work | Very Low | $16m^2$/transmitter | 20-30cm/ Yes (<45 degrees) |
| Acoustic sound (time of flight/ reception angle) | Medium | Tens of meters | 10cm/ Yes (5 degrees) |
| Ultrasonic (time of flight) | Medium | Tens of meters | 2-3cm/ No |
| Ultrasonic (phase difference) | High | $10\text{-}20m^2$ | Few centimeters/ Yes (<20degrees) |
| Laser (time of flight) | High | One hundred meters or longer | 2-3cm/ Indirectly |
| Image Processing | Very High | Tens of meters | Few centimeters / Indirectly |
| RF(Received Signal Strength Indicator) | Medium | 2-3m | Tens of centimeters/ No |

is presented. Each one of these regions has an area of 1.8mX1.8m and is covered by a virtual grid with 30cm node distance. It should be noted that a single IRTX device can cover an area of up to $16m^2$ but the areas covered by two neighboring IRTX devices should overlap in order to define rules like the ones presented in Table 2. It will also be assumed that the lower left corner of a region R*i* has coordinates $R_i(0,0)$ as shown in Figure 15.

Table 3 lists the real target coordinates and orientations and the experimentally measured ones. Three samples have been examined for every position and the experimental coordinates and orientation listed in Table 3 are the best ones achieved from these 3 samples. The 5th and 7th columns are an indication of the results' stability denoted as x/3 where x shows how many samples among the 3 considered leaded to the selected best results. In the ASR columns the Average Success Rates measured by $IRTX_A$ and $IRTX_B$ are listed and these values were used for the estimation of the best position/ orientation mentioned in this table. The number in the parentheses in these two columns refers to the IRTX device that corresponds to the listed ASR. The last column shows the positions where the Forward Error Correcting method (FEC) and the 3rd stage of the localization procedure improved the results.

It is worth mentioning the case of p5 where the system was unable to reach the right decision about the orientation of the target (decided E instead of SE). Consequently, the estimated coordinates could not be close enough to the real target position. The signal characterization extracted at p5 did not match any orientation and the conclusion for the E orientation was reached by ignoring the Weak indication from IRTX1.

The stability of the orientation results is also very good since in most positions the right orientation was constantly selected. The stability of the position coordinates is slightly worse, but rules like the ones described in [16] that were not used in the present case study can assist the system to improve its results. In many cases, the Turbo Decoding method led to better results by discarding instant noise effects as indicated by the last column of Table 3. Finally, the 3rd localization stage was helpful only once. This is due to the fact that the grid node distance used was relatively long.

Table 4 provides a general comparison between various localization methods. As already mentioned the cost of the proposed solution is extremely low since it is sufficient to count the received patterns by incorporating slow-speed microcontrollers without the necessity to employ high-precision sensors. Moreover, a large area can be covered by a small number of infrared transmitters since a single transmitter can cover a $16m^2$ area. The accuracy in the measured distance or the orientation is quite good for several applications. Acoustic sound and ultrasonic waves can achieve a better accuracy and they can also provide some orientation indication but they require significantly higher cost sensors and processing units in order to discriminate small time intervals or phase shifts. Laser beams can achieve a very good accuracy and can cover a wide range but they have significantly higher cost since they require much faster processing units. They also provide orientation indication in an indirect manner since they are used to measure a distance at a specific direction. Image processing also requires very high cost complicated sensors and processing units in order to analyze a captured image. Orientation can be concluded indirectly if the target is aware of his environment by detecting specific landmarks in his direction. Finally, Received Signal Strength Indicator (RSSI) can also provide a draft distance estimation of the RF receiver.

## 5. Conclusions

A localization system capable of determining the orientation of a target in one of 8 possible directions with high estimation stability was described in this paper. Beside this draft orientation, the system is also capable to determine the position of the target with a maximum absolute error less than 30cm. The estimation method is based on a simple enumeration of infrared patterns in order to estimate a success rate parameter that qualifies the signal of a specific transmitter. Each infrared transmitter covers an area of up to $16m^2$ and multiple transmitters with partially overlapping range can be used to cover the desired area. The infrared patterns are received by two sensors that are positioned at an angle of $135^o$ on the target in order to differentiate the quality of reception from different transmitters. Simple commercial components have been used at the side of the transmitter and the receiver, since no high precision analog measurements are required, leading to ultra low cost implementations.

Future work will focus on increasing the distinct directions that can be distinguished at the various target positions without sacrificing the accuracy, stability and speed of the estimation method. The localization of a target in 3 dimensions with 4 degrees of freedom (including rotation) using a similar method than the one presented here will also be investigated.

## 6. References

[1] A. Clerentin, A. Delahoche, E. Brassart and C. Drocourt, "Self localisation: A new uncertainty propagation architectur," Elsevier Robotics and Autonomous Systems, Vol. 51, pp. 151–166, 2005.

[2] J. Borenstein, B. Everett and L. Feng, "Navigating mobile robots: Systems and techniques," A.K. Peters Ltd Wellesley, 1996.

[3] J. Miura, Y. Negishi, and Y. Shirai, "Adaptive robot speed control by considering map and motion uncertainty," Elsevier Robotics and Autonomous Systems, Vol. 54, pp. 110–117, 2006.

[4] P. Smith and K. Zografos, "Sonar for recognizing the texture of pathways," Robotics and Autonomous Systems, Vol. 51, pp. 17–28, 2005.

[5] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama, "Dolphin: A practical approach for implementing a fully distributed indoor ultrasonic positioning system," Lecture Notes for Computer Science (LNCS), Vol. 3205, pp. 347–365, 2004.

[6] A. Ladd, K. Bekiris, A. Rudys, L. Kavraki, and D. Wallach, "Robotics based location sensing using wireless ethernet," Wireless Networks, Vol. 11, No. 1–2, pp. 189–204, January 2005.

[7] C. Flora, M. Ficco, S. Russo, and V. Vecchio, "Indoor and outdoor location based services for portable wireless devices," Proceedings of 1st IEEE International Workshop on Services and Infrastructure for Ubiquitous and Mobile Internet, Columbus OH, pp. 244–250, June 2005.

[8] J. Kosel, H. Pfutzner, L. Mehnen, E. Kaniusas, T. Meydan, N. Vazquez, M. Rohn, A. Merlo, and B. Marquardt, "Non contact detection of magnetoelastic position sensors," Elsevier Sensors and Actuators A, No. 123–124, pp. 349–353, 2005.

[9] V. Schlageter, P. Besse, R. Popovic, and P. Kucera, "Tracking system with 5deg of freedom using a 2d array of hall sensors and a permanent magnet," Elsevier Sensors and Actuators A, No. 92, pp. 37–42, 2001.

[10] E. Prigge and J. How, "Signal architecture for distributed magnetic local positioning system," IEEE Sensors Journal, Vol. 4, No. 6, pp. 864–873, 2004.

[11] T. Jin, J. Lee, and S. Tso, "A new space and time sensor fusion method for mobile robot navigation," Wiley Journal of Robotics Systems, Vol. 21, No. 7, pp. 389–400, 2004.

[12] E. Bicho, P. Mallet, and G. Schoner, "Target representation on an autonomous vehicle with low level sensors," International Journal of Robotics Research, Vol. 19, No. 5, pp. 424–447, May 2000.

[13] T. Aytac and B. Barshan, "Simultaneous extraction of geometry and surface properties of targets using simple infrared sensors," SPIE Optical Engineering Journal, Vol. 43, No. 10, pp. 2437–2447, October 2004.

[14] G. Benet, F. Blanes, J. Simo, and P. Perez, "Using infrared sensors for distance measurement in mobile robots,"

Robot Autonomy Systems, Vol. 30, pp. 255–266, 2002.

[15] N. Petrellis, N. Konofaos, and G. Ph. Alexiou, "Target localisation utilising the success rate in infrared pattern recognition," IEEE Sensors Journal, Vol. 6, pp. 1355–1364, 2006.

[16] N. Petrellis, N. Konofaos, and G. Ph. Alexiou, "Using future position restriction rules for stabilizing the results of a noise sensitive indoor localization system," SPIE Optical Engineering Journal, Vol. 6, No. 46, Article No 067202, pp. 067202-1–067202–11, 2007.

[17] N. Petrellis, F. Gioulekas, M. Birbas, J. Kikidis, and A. Birbas, "Use of interleaving and error correction to infrared patterns for the improvement of position estimation systems," in Proceedings of IEEE Conference Emerging Technologies-Factory Automation, Hamburg, Germany, pp. 888–891, September 2008.

[18] N. Petrellis, N. Konofaos, and G. Ph. Alexiou, "A wireless infrared sensor network for the estimation of the position and orientation of a moving target," in Proceedings ICST-ACM Conference MobiMedia'07, Nafpaktos, Greece, August 2007.

[19] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedig, and M. Jezequel, "Semi-iterative analog turbo decoding," IEEE Transactions on Circuits and Systems, Vol. 54, No. 6, pp. 1305–1315, 2007.

[20] C. Schlegel and L. Perez, "Trellis and turbo coding," IEEE Series on Digital & Mobile Communication, Wiley Interscience, 2004.

[21] F. Gioulekas, M. Birbas, A. Birbas, and G. Bilionis, "Analog error-correcting decoders using SiGe BiCMOS technology," International Journal of Analog Integrated Circuits and Signal Processing, Springer, Vol. 52, No. 3 pp. 117–132, October 2007.

[22] M. A. Bickerstaff, D. Garret, T. Prokop, C. Thomas, and C. Nicol, "A 24 Mb/s radix-4 logMAP turbo decoder for 3GPP-HSDPA mobile wireless," in Proceedings IEEE Conference International Solid-State Circuits, San Francisco, CA, pp. 150–151, February 2003.

*Wireless Sensor Network*, **2009, 1, 446-452**
doi:10.4236/wsn.2009.15053 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Group Target Tracking in WSN Based on Convex Hulls Merging

**Quanlong LI, Zhijia ZHAO, Xiaofei XU, Tingting ZHOU**
*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China*
*Email*: *liquanlong@hit.edu.cn*
*Received July* 22, 2009; *revised September* 21, 2009; *accepted September* 22, 2009

## Abstract

When a mass of individual targets move closely, it is unpractical or unnecessary to localize and track every specific target in wireless sensor networks (WSN). However, they can be tracked as a whole by view of group target. In order to decrease the amount of energy spent on active sensing and communications, a flexible boundary detecting model for group target tracking in WSN is proposed, in which, the number of sensors involved in target tracking is adjustable. Unlike traditional one or multiple individual targets, the group target usually occupies a large area. To obtain global estimated position of group target, a divide-merge algorithm using convex hull is designed. In this algorithm, group target's boundary is divided into several small pieces, and each one is enclosed by a convex hull which is constructed by a cluster of boundary sensors. Then, the information of these small convex hulls is sent back to a sink. Finally, big convex hull merged from these small ones is considered as the group target's contour. According to our metric of precision evaluation, the simulation experiments confirm the efficiency and accuracy of this algorithm.

**Keywords:** Sensor Networks, Target Tracking, Group Target, Convex Hull

## 1. Introduction

Target tracking is a basic application of WSN, and hence has received a considerable amount of attentions in research community [1–5]. However, most of the previous works are limited to investigating point target tracking, where a target is regarded as a point and the localization result is also a point. There are also several methods which have discussed to solve multi-target tracking in WSN, e.g. [4]. However, in order to distinguish different targets, more complicated computations and communications are required, especially when the number of targets is relatively large. In fact, when a group of targets move closely to each others (e.g., motorcade, tank column, or a herd of buffalo) in a sensor network, it is unpractical or pointless to localize every specific target. Therefore, the schemes for tracking point targets in WSN are not suitable for the cases where there are large quantities of targets moving within a huge area. Instead, it's more valuable and feasible to track these targets as a whole called group target.

Recently, there are several works on group target tracking in WSN [6–9]. In order to detect chemical pollution-like events, [6] proposed a scheme for contour tracking by constructing and maintaining a contour network which tightly surrounds the contours and captures the important topological features. A fully distributed protocol, named CollECT, is proposed in [7] to detect and track events in a wireless heterogeneous sensor network. A dynamic cluster-based structure is introduced in [8] to detect and track the movements of boundaries of continuous objects in a sensor network. The authors in [9] estimate a group target's boundary as a circle covering all border sensors.

Unlike all the above mentioned revelatory works, we proposed a group target tracking protocol on the boundary detecting model, clustering model and group target's boundary construction algorithm. As a first attempt, the idea of convex hulls merging is used to track group target.

In this method, the target tracking process is separated into two steps: boundary dividing and boundary merging. In the first step, the sensors chosen to detect the boundary of a group target are divided into multiple clusters and each cluster is responsible for tracking a partial boundary of the group target. In each cluster, there is a cluster head (CH) which gathers information from its cluster members (CM) and aggregates these data to form a local convex hull. Then, the aggregated data is sent back to the sink which is usually monitored by humans. In the second step, when sufficient information of local convex hulls is collected at the sink, it will execute the merging algorithm to combine those convex hulls into a global convex hull which is considered as the whole

contour of the group target.

The rest of the paper is organized as follows. In Section 2, two analytical models for group target tracking in WSN are proposed. In Section 3, the group target tracking algorithm is designed. The details of simulations are discussed in Section 4. Finally, Section 5 concludes this paper.

## 2. Analytical Model

Unlike the point target tracking, the problem of group target tracking is much more challenging because of several reasons. Firstly, if considering target as a point, it's easier to measure the location result and obtain the localization accuracy. However, in the group target tracking case, the location result is an area. Then, the question is how to measure the accuracy of localizing group target? Secondly, the group target always occupies a region which is covered by numerous sensors, but some of them may be too far to communicate. So how could they cooperate with each other to efficiently track the same group target? Thirdly, if we want to divide the tracking nodes into several groups, what are the rules to categorize them and how could these segments be merged to form a global contour? All these questions will be clarified by the novel methods which will be stated in this and following sections.

### 2.1. Boundary Detecting Model

In [6], there is a simple method to detect object's boundary, but the width of boundary is fixed. In the following, a more flexible detecting model called Boundary Detecting Model (BDM) will be proposed, where the width of boundary is adjustable. The model is built upon the binary sensing model [3,5], which is famous for its minimal requirements of sensing and processing capabilities. Moreover, it can be easily extended to be applied on other types of sensors.

**Definition 2.1 Discovering Neighbors Ratio (DNR)**: The Discovering Neighbors Ratio (DNR) is defined as a function $\eta$ of node $i$:

$$\eta : I \to [0,1]$$
$$\eta(i) = \frac{\varphi(i)}{\theta(i)}, \text{ for all } i \in I \quad (1)$$

where $I$ is set of sensors, $\varphi(i)$ is the number of node $i$'s neighbors that are discovering targets (named *discovering neighbors*) and $\theta(i)$ is the number of node $i$'s neighbors.

DNR denotes the ratio between the number of discovering neighbors and that of all neighbors. Further, when $0 < \eta(i) < 1$, some of node $i$'s neighbors are discovering neighbors, while others not. Then it is reasonable to conclude that $i$ is closed to group target's "boundary". There is a *neighbors table* in each node, where the neighbors' discovery information is stored. According to the Formula (1), the *boundary status* can be defined as:

**Definition 2.2 Boundary Status (BS)**: The Boundary Status is a function $\chi$ of node $i$:

$$\chi : I \to \{INNER,\ BOUNDARY,\ OUTER\}$$
$$\chi(i) = \begin{cases} OUTER, & \eta(\mathrm{i}) \le H_0 \\ BOUNDARY, & H_0 < \eta(\mathrm{i}) < H_1, \text{ for all } i \in I \\ INNER, & \eta(\mathrm{i}) \ge H_1 \end{cases} \quad (2)$$

where $H_0$ and $H_1$ are parameters to adjust the width of boundary. The BS describes whether one node is on the boundary of a region which contains a group target.

**Definition 2.3 Boundary Detecting Model (BDM)**: The Boundary Detecting Model (BDM) is a sensing model, in which every sensor can calculate its BS by communicating with its neighbors.

Figure 1 illustrates a snapshot of group target appearing in a region deployed with numerous BDM sensor nodes. The white circle represents OUTER sensor node, gray circle represents BOUNDARY sensor node, black circle represents INNER sensor node and red triangle represents target.

Every sensor is in the OUTER status initially. Once the signal strength of events it captured exceeds a certain threshold, the node turns into discovering status. Meanwhile, it also has the responsibility of notifying its neighbors to update their neighbors' tables. Based on the updated neighbors table, a node could calculate its DNR and decide which BS it will become. There are three different situations: i) if DNR$\leqslant H_0$, it gets into the OUTER status; ii) if DNR$\geqslant H_1$, it gets into the INNER

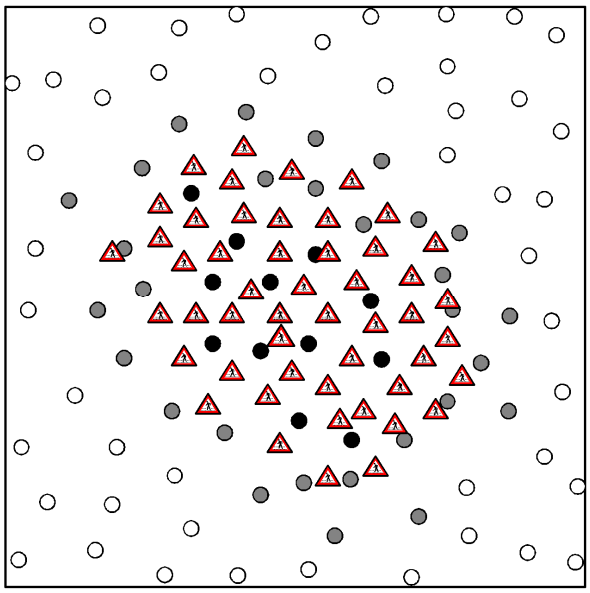


**Figure 1. A scene of group target in WSN.**

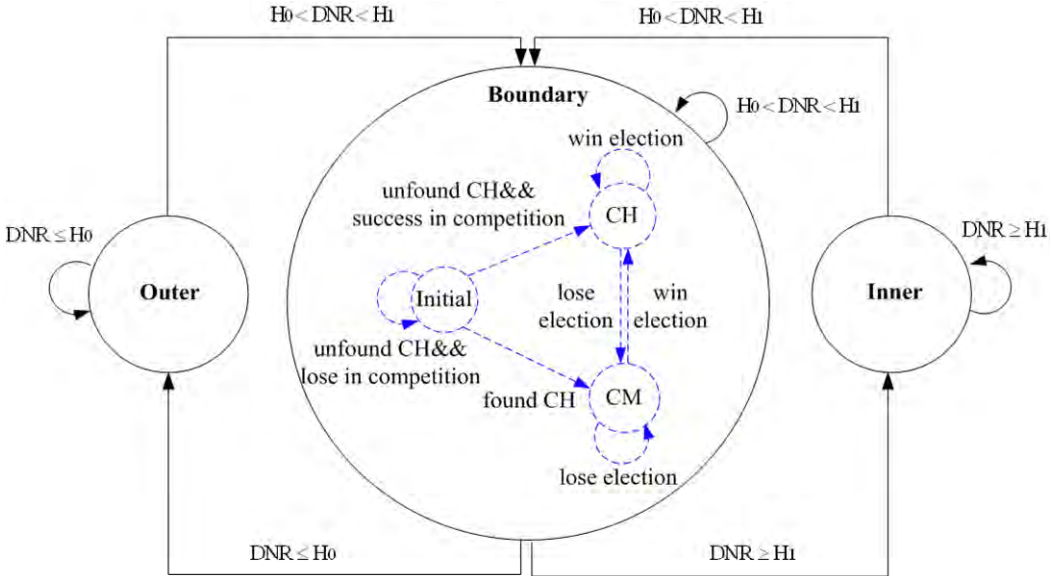



**Figure 2. Node's complete status transition diagram.**

status; iii) otherwise, it gets into the BOUNDARY status. The whole process is completely distributed and only needs local communications among sensors.

### 2.2. Boundary Clustering Model

Unlike other clustering methods [1] used in WSN, we only needs the BOUNDARY nodes in the clustering process in our clustering model. We divide BOUNDARY nodes into several clusters without considering the INNER and OUTER nodes. Because the ratio between the number of BOUNDARY nodes and the number of discovering nodes decreases with the group target's scale increasing, the energy consumption of the whole WSN can be reduced accordingly.

At the beginning, all BOUNDARY Nodes stay in the initial status. Then, they get into an election phase. After that, the winner changes into the CH status, while loser gets into the CM status after it chooses one CH as its head.

In this section, we introduced two useful models which are essential to support our algorithm for tracking group target. In Figure 2, complete status transition diagram of a node is shown. The dotted lines represent the *cluster status* transition that runs on each BOUNDARY node, while the solid lines represent the BS transition that runs on every node.

## 3. Tracking Algorithm Based on Convex Hulls Merging

### 3.1. Group Target Localization Algorithm

In the previous work [9], a group target is localized as a rough circle. In this paper, our localization result is a convex hull which is more accurate compared with the circle in [9]. However, considering the limitation of the capability of sensors, we cannot get the accurate convex hull contains all the targets in a group. Instead, we can calculate the convex hull enveloped by all BOUNDARY nodes and it approximates the accurate convex hull.

There are some methods [10–12] to calculate the convex hull of a set of vertexes. To reduce the computation complexity and energy consumption, two algorithms are selected carefully and some variations and integrations are made necessarily. One of the convex hull algorithms is known as *Graham-Scan* [10], and the other is an on-line approximate convex hull algorithm [12]. These two algorithms run on the sink and deployed nodes respectively. The main idea about the group target localization algorithm can be illustrated as follows. First, the convex hull algorithm runs on each CH, which collects the information from all its members and is responsible for sending the result (local convex hull) back to the sink. When the sink gathers enough data of local convex hulls, the merging algorithm will be executed to generate the entire convex hull. The complete algorithm can be described as follows, named *Convex Hulls Merging Localization* (*CHML*).

***Algorithm 1*: Convex Hulls Merging Localization**
*Divide*:1:CH collects position information from its members;
2:executes on-line algorithm to compute local convex hull;
3:sends local convex hull back to a sink;
*Merge*:4:**IF** (the sink receives sufficient local convex hulls)
Execute merging algorithm to compute global convex hull;
**ELSE**
Receive local convex hulls.

Considering the dynamic case, when the group target moves the clusters around it will keep changing. If the convex algorithm is re-executed for each change, more computation and communication resources will be consumed. Therefore, it is reasonable to use an on-line algorithm on CHs. In the sink, we use the convex hulls merging algorithm which is a divide-and-conquer algorithm [11] for convex hull. In this paper, we choose an approximation of the on-line algorithm for its simplicity. The error of this algorithm is guaranteed to be very low. The time-complexities of these two algorithms are $O(n)$ and $O(n\log n)$ respectively, where $n$ is the number of BOUNDARY nodes in total. So the total time-complexity of our algorithm is $O(n\log n)$. Refer to appendix for details of the algorithms.

### 3.2. Cluster Maintenance Algorithm

When a group target is moving, some previously constructed clusters are destroyed and some new ones will be formed. In order to track the group target continuously, clusters must be maintained so that out-of-date clusters are eliminated and new clusters are dynamically created along the trajectory of the group target.

Consider a newly formed cluster. When the group target is moving, some new nodes will join the cluster. Meanwhile, some old ones quit. So the topology of the cluster is changing and the original CH may not be able to continue playing as a cluster head. For instance, some new nodes may be too far away to achieve the communication with the CH. In this case, a new CH will be selected.

Following criterions need to be considered when a new CH is elected. First, the nearer a node is to the center of the cluster, the more suitable it is to become a CH [8]. Therefore, the sum of the distances from the CH to its members will be the smallest one and it will help to reduce the communication consumption and delay. Furthermore, it will ensure the cluster moves in a direction which the group target towards. Second, the convex hull information owned by the original CH is useful to the new CH. For this reason, a node should better be chosen if it could inherit this information easily.

Based on the two criterions, a weighted election method is proposed. Let $C_j$ be a cluster, $C_j \subseteq I$, then for any $i$, $i \in C_j$, the distance from $i$ to its cluster's topology center is $d_i$,

$$d_i = \sqrt{(a_i.x - (\frac{1}{|C_j|}\sum_{i \in C_j} a_i.x))^2 + (a_i.y - (\frac{1}{|C_j|}\sum_{i \in C_j} a_i.y))^2} \quad \text{d10} \quad (3)$$

where $a_i$ is the position of node $i$. Then the weight of node $i$ is $W_i$,

$$W_i = w_1 \times d_i + w_2 \times \frac{1}{n+1} \quad (4)$$

where $w_1$ and $w_2$ are weights of factors, $n$ is the number of lost nodes if node $i$ is elected as a new CH. So when a new BOUNDARY node participates, or an old one quits, the CH will compute the weight of each node in its cluster. If there is a node whose weight is bigger than that of the previous CH, this node will take the old CH's place and become the new CH.

## 4. Performance Evaluation

### 4.1. Simulation Setup

In the simulation, binary sensors are randomly scattered with a uniform distribution in the monitoring region which is a rectangle area with the size of $1000m \times 700m$. The communication radius and sensing radius are changed according to the deployment density of sensors to guarantee enough coverage of the monitoring region.

We simulate a group target moving at the speed of $5m/s$. If there is any target gets into one sensor's sensing area whose sensing radius is $R_{\text{sense}}$, the sensor will discover it without knowing the number of targets or their accurate positions.

### 4.2. Evaluation Criteria

In a point target tracking evaluation, the localization result is a point and the error can be defined as the distance between the actual position and estimated position. However, the group target localization result is a polygon area. To measure the localization accuracy, we propose the following evaluation criteria. Let polygon $P_A$ be a group target's convex hull and polygon $P_E$ be its estimated convex hull. Then, the error is defined as:

$$e = \frac{S_{P_E \cup P_A} - S_{P_E \cap P_A}}{S_{P_E \cup P_A}} \times 100\% \quad (5)$$

where $S$ represents the area of a polygon. Apparently, $0 \le e \le 1$ is satisfied. This error definition not only reflects the coverage extent of $P_E$ over $P_A$, but also avoids $P_E$ to be too large. So the lower the $e$ is, the better the performance can be achieved. By varying the number of deployed sensors and their sensing radiuses, the impact of sensor density and sensing radius on the accuracy of estimated convex hull is investigated.

### 4.3. Simulation Results

Figure 3(a) shows the tracking accuracy during a complete tracking process. It can be seen that, from time 0s to 10s, the accuracy increases sharply. But at the end of the process, the different situations happen. In fact, these variations reflect the process of a group target getting into the monitoring region and that of leaving the area.

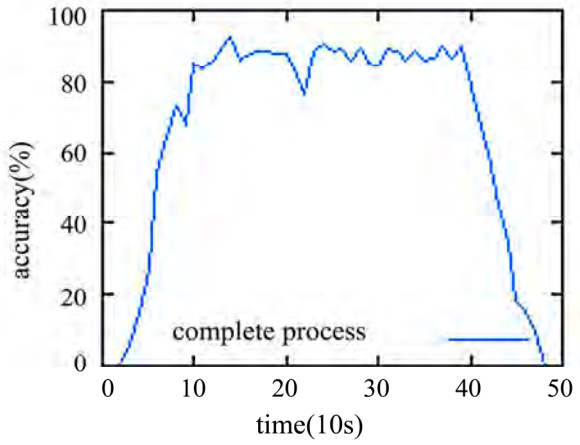



**Figure 3(a). Tracking accuracy during a complete tracking process.**

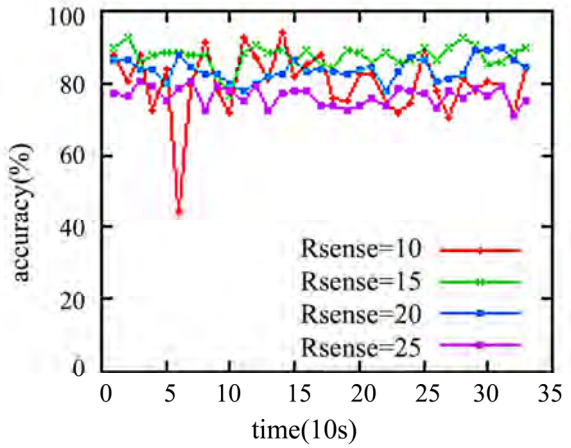



**Figure 3(b). Accuracies of group target tracking under different sensor deployment densities.**

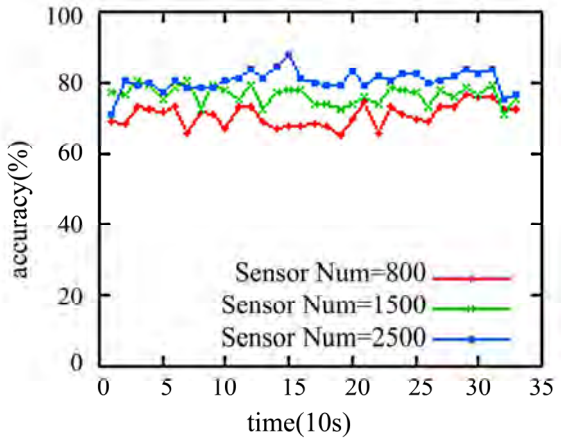



**Figure 3(c). Impact of sensors' density on accuracy.**

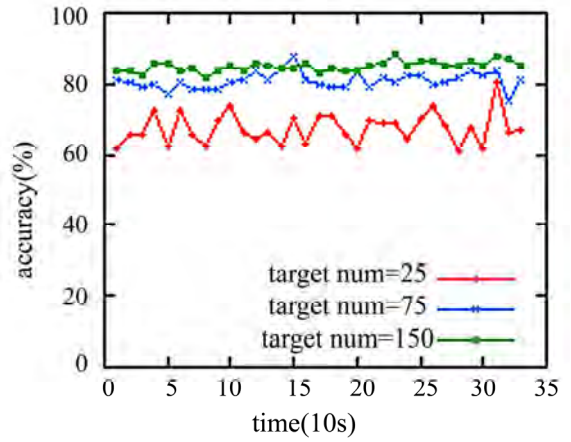



**Figure 3(d). Number of targets and tracking accuracy.**

In Figure 3(b), accuracies of group target tracking algorithm under different sensor deployment densities are presented. This figure indicates that, when the sensors' density is fixed, the performance gets better as the sensing radius decreases. But this situation changes when $R_{sense}$ =10. At this time, the tracking results are unstable. The reason is that, as sensing radius is decreasing, the degree of network coverage also reduces, which affect the tracking accuracy of our algorithm.

The impact of sensors' density on accuracy is showed in Figure 3(c). To be reasonable, no matter how many sensors are there, the coverage degree should be guaranteed. The simulation results show that the more sensors are deployed, the better performance is. By varying the number of targets in the group target, its impact on accuracy is examined. Figure 3(d) shows that, when the number of targets increases, the accuracy of tracking algorithm improves accordingly.

Figure 4 illustrates a segment of trajectory in a complete tracking process. From this figure, we can find that

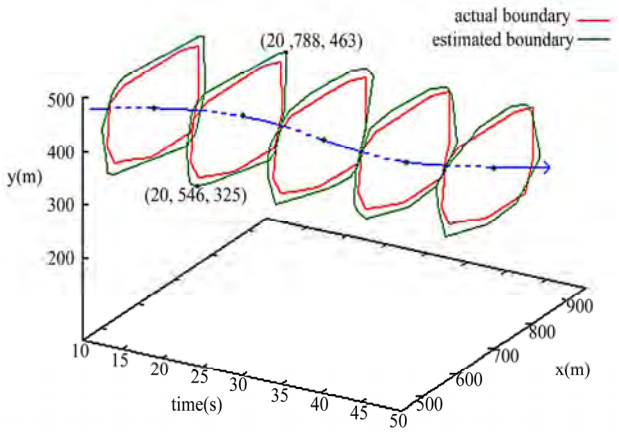



**Figure 4. Comparison between actual boundary and estimated one.**

the estimated boundary is closed to actual boundary as the group target moving.

## 5. Conclusions

The problem of group target tracking in wireless sensor networks is fully investigated in this paper. To achieve the group target tracking goal, a flexible group target boundary detecting model is proposed. Moreover, utilizing a computing geometry conception—convex hull, a novel divide-merge group target tracking algorithm is proposed to solve this challenging problem. By analyzing the experimental results from simulations, the group target tracking performance is evaluated. The results show that the proposed algorithm is effective and efficient.

## 6. References

[1] W. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," IEEE Transactions on Mobile Computing, Vol. 3, No. 3, pp. 258–271, 2004.

[2] W. Zhang and G. Cao. "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," IEEE Transactions on Wireless Communications, Vol. 3, No. 5, pp. 1689–1701, 2004.

[3] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," ACM Sensys'03, Los Angeles, California, USA, pp. 150–161, 2003.

[4] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," In Proceedings of the 6th International Conference on Information Processing In Sensor Networks, April 2007.

[5] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri. "Target tracking with binary proximity sensors: Fundamental limits, minimal descirptions, and algorithms," In Proceedings of ACM SenSys, 2006.

[6] X. Zhu, R. Sarkar, J. Gao, and J. S. B. Mitchell, "Light-weight contour tracking in wireless sensor networks," In Proceedings of IEEE INFOCOM, pp. 1849–1857, 2008.

[7] K. P. Shih, S. S. Wang, P. H. Yang, and C. C. Chang, "CollECT: Collaborative event detection and tracking in wireless heterogeneous sensor networks," In Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06).

[8] X. Ji, H. Zha, John J. Metzner, and G. Kesidis, "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks," IEEE International Conference on Communications, Paris, FRANCE, June 20–24, 2004.

[9] D. Cao, B. Jin, and J. Cao, "On group target tracking with binary sensor networks," In Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor System (MASS), pp. 334–339, 2008.

[10] O' Rourke and Joseph, "Computational geometry in C," China Machine Press, 2005.

[11] T. H. Cormen, C. E. Leiseron, and R. L. Rivest. "Introduction to algorithms," The MIT Press, pp. 947–956, 2002.

[12] R. C. T. Lee, S. S. Tseng, and R. C. Chang, "Introduction to the design and analysis of algorithms," Addison-Wesley, Chapter 12.6, 2002.

# Appendix

## I. Dynamic approximation convex hull algorithm:

**Input:** A sequence of points with operators $(p_1, operator_1), (p_2, operator_2), \cdots$, and the number m of pairs of parallel lines being used.

**Output:** A sequence of approximate convex hulls $a_1, a_2, \cdots$, where $a_i$ is covering the reserved points $p_1, p_2, \cdots, p_l$, where $l \le i$.

**Initialization:** Construct m pairs of parallel lines with slopes $0, \tan(\pi/m), \tan(2\pi/m), \cdots, \tan((m-1)\pi/m)$ respectively and locate all lines so that they all intersect at the first input point $p_1$. Set i=1, i.e. the current input point with operator is $(p_1, insert)$.

Step 1. **IF** ( $p_i$ is between each of the m pairs of parallel lines)

**IF (**operator=delete**)** delete the point $P_i$, $l = l - 1$, Stop;

**ELSE** insert the point $P_i$, $l = l + 1$, Stop;

**ELSE**

**IF** (operator=delete**)** delete the point $P_i$, $l = l - 1$;

**ELSE** insert the point $P_i$, $l = l + 1$, $p' = p_i$;

Step 2. Move the line nearest to point $p'$, without changing its slope, to touch $p'$ and associate $p'$ with this line.

Step 3. Construct an approximate convex hull by connecting the 2m points with respect to each line in the clockwise fashion and denote this approximate convex hull as $a_j$.

Step 4. If no other point will be input, then stop; otherwise, set i=i+1 and receive the next input point as $P_i$. Go to Step 1.

## II. Graham-Scan(Q) // Algorithm which will be run on the sink

Step 1.let $p_0$ be the point in Q with the minimum y-coordinate, or the leftmost such point in case of a tie;

Step 2. let $\langle p_1, p_2, \cdots, p_m \rangle$ be the remaining points in Q, sorted by polar angle in counterclockwise order around $p_0$ (if more than one point has the same angle, remove all but the one that is farthest from $p_0$);

Step 3. PUSH( $p_0$ ,S);

Step 4. PUSH( $p_1$ ,S);

Step 5. PUSH( $p_2$ ,S**)**;

Step 6. for(i=3, i<=m, i++)

Step 7. {while(the angle formed by points NEXT-TO-TOP(S), TOP(S), and $P_i$ makes a nonleft turn)

Step 8. POS(S);

Step 9. PUSH($P_i$, S); }

Step 10. return S;

*Wireless Sensor Network*, **2009, 1, 453-457**
doi:10.4236/wsn.2009.15054 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Research on Blind Source Separation for Machine Vibrations

**Weiguo HUANG, Shuyou WU, Fangrang KONG, Qiang WU**
*Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei, China*
*Email*: *wghuang@mail.ustc.edu.cn*
*Received August* 3, 2009; *revised August* 18, 2009; *accepted August* 20, 2009

## Abstract

Blind source separation is a signal processing method based on independent component analysis, its aim is to separate the source signals from a set of observations (output of sensors) by assuming the source signals independently. This paper reviews the general concept of BSS firstly; especially the theory for convolutive mixtures, the model of convolutive mixture and two deconvolution structures, then adopts a BSS algorithm for convolutive mixtures based on residual cross-talking error threshold control criteria, the simulation testing points out good performance for simulated mixtures.

**Keywords:** Blind Source Separation, Independent Component Analysis, Convolutive Mixtures, Machine Vibration, Residual Cross-Talking Error

## 1. Introduction

Blind source separation (BSS) is a typical tool to recover source signals from several observations usually provided by a set of sensors. Due to the lack of prior knowledge of the source signals, generally, this method considers an assumption of independence between the sources. It has been successfully used in many fields, such as biomedicine, telecommunications, speech processing, underwater acoustics [1–3]. But BSS methods have seldom been used for monitoring or diagnosing the mechanical devices, in many cases the signals obtained by sensors consist of some useless signals, such as environment noise, other mechanical devices [4–6]. In these condition, using BSS as a preliminary step will reduce, even remove the useless signals, which can significantly improve the efficiency and accuracy of the condition monitor and fault diagnosis.

At present, most of BSS works are related to the separation of linear mixtures of sources. However, the vibration of mechanical devices is complex, and so is the propagation medium. The mechanical vibration signals are often convolutive mixtures [7].

Consequently, this paper mainly describes BSS for convolutive mixtures and its application to mechanical vibrations.

## 2. BSS

### 2.1. BSS Introduction

Blind source separation is a signal processing technique by which unobserved signals, also called sources, are recovered from the observation of several mixtures. The term “Blind” includes two facts: both the source signals and the mixing structure are unknown. In the present research work, most are assuming the mutual independence of the sources. This is the fundamental basis of BSS.

For condition monitoring and fault diagnosis, the observed signals are usually the output of a set of sensors and linear combinations of the sources. Just as Figure 1 shows.

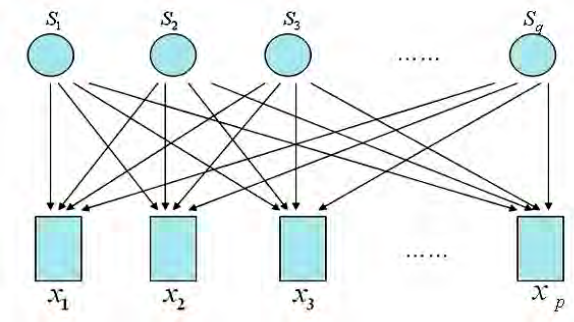



**Figure 1. Observed signals and sources.**

In Figure 1, the $S_1$, $S_2$, $S_3$,…$S_q$ denote the sources and the $x_1$, $x_2$, $x_3$,…$x_p$ denote the observations (output of sensors). When condition monitoring and fault diagnosing of mechanical devices, the $P$ observations (output of $P$ sensors) are the linear combinations of the $q$ sources. In this way, the i-th observation (output of the i-th sensors) is:

$$x_i = \sum_{j=1}^{q} a_{ij} s_j(t) + n_i(t) \quad i = 1, 2, 3, \cdots, p \tag{1}$$

where $a_{ij}$ is the linear combination coefficients, $n_i(t)$ denotes the environment noise received by the i-th sensor.

The noise may be considered as a source signal, in this way, the mathematical model of BSS could be shown as

$$X(t) = AS(t) \tag{2}$$

where $X(t) = [x_1(t), x_2(t), \cdots, x_p(t)]^T$ denotes the $p$ observed signals, which are the available data. A is the unknown $p \times q$ mixing matrix, which denotes the unknown propagation. $S(t) = [s_1(t), s_2(t), \cdots, s_q(t)]^T$ denotes the $q$ source signals, which include the noises. Here, generally, $p \geq q$ and $[]^T$ denotes the transpose operator.

Here, assuming the mixing matrix A is invertible and the sources $S_i(t)$ ($i$=1,2,…$q$) are statistically independent. The assumption of independence between the sources is physically plausible because they have different origins [8].

The kernel of BSS is to find a $q*p$ separation matrix B and the recovered signals are

$$\tilde{S}(t) = BX(t) = BAS(t) = CS(t) \tag{3}$$

If matrix B could make the matrix C be an identity matrix, it could be concluded that the source signals have been separated perfectly.

The general model of BSS could be shown as Figure 2.

## 2.2. Model of Convolutive Mixture [9]

For condition monitoring and fault diagnosis, vibration analysis involves a convolutive mixture because of the propagation medium (structure of the system). The environment noise may be considered as a source signal. The general model of a convolutive mixture can be represented as in the Figure 3 for two source signals and two observation signals (p=q=2) to be simplified.

Moreover, if the sensors are located near the source signals, respectively, we could consider that the filters

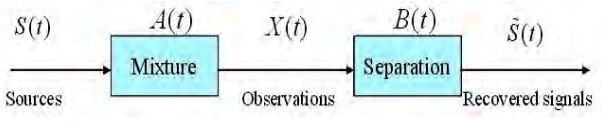



**Figure 2. BSS general model.**

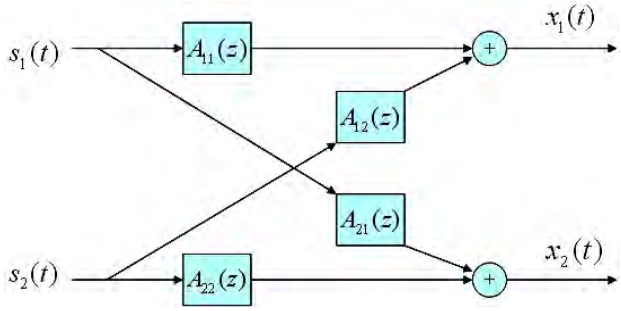



**Figure 3. Two source signals and two observation signals model.**

$A_{11}$ and $A_{22}$ are equal to 1. In fact, it is significantly for condition monitoring and fault diagnosis that the sensors are as close as possible to the origins. In this way, we can get a simplified model:

$$\begin{aligned} x_1(n) &= s_1(n) + A_{12} * s_2(n) \\ x_2(n) &= A_{21} * s_1(n) + s_2(n) \end{aligned} \tag{4}$$

Using the Z transform

$$\begin{aligned} X_1(z) &= S_1(z) + A_{12}(z) S_2(z) \\ X_2(z) &= A_{21}(z) S_1(z) + S_2(z) \end{aligned} \tag{5}$$

and the matrix formulation

$$X(z) = A(z) S(z) \tag{6}$$

where

$$A(z) = \begin{bmatrix} 1 & A_{12} \\ A_{21} & 1 \end{bmatrix} \tag{7}$$

assuming that the filters of $A(z)$ can be modeled by Lth-order causal transverse filters, so that the matrix $A(z)$ are given by

$$A_{ij}(z) = \sum_{k=0}^{L-1} a_{ij}(k) z^{-k} \tag{8}$$

Formulation (5) may be expanded in the time domain

$$\begin{aligned} x_1(n) &= s_1(n) + \sum_{k=0}^{L-1} a_{12}(k) s_2(n-k) \\ x_2(n) &= \sum_{k=0}^{L-1} a_{21}(k) s_1(n-k) + s_2(n) \end{aligned} \tag{9}$$

## 2.3. Separation Principle and Criteria

Theoretically, if the mixing matrix A is invertible, and the separation matrix B is the inverse matrix of the mixing matrix A, it could be reconstituted the source signals perfectly. But, in fact, the mixing matrix A is unknown; we could not obtain the separation matrix B by A directly.

In the case of the Figure 3, Herault [2] proposed a solution

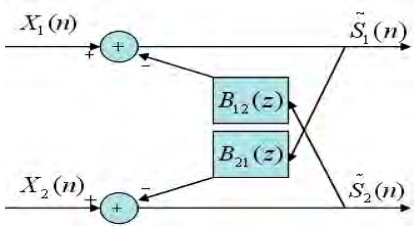



**Figure 4. Recursive structure for the BSS.**

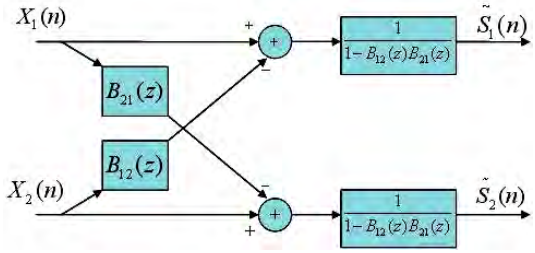



**Figure 5. Direct structure for the BSS.**

based on a recursive architecture, which can be generalized in the case of convolutive mixtures modeled by FIR filters as Figure 4: Recursive structure.

Also, it can be modeled by Figure 5: Direct structure.

In the Z-domain, the relationship between recovered signals and observation signals is

$$\begin{bmatrix} \tilde{S}_1(z) \\ \tilde{S}_2(z) \end{bmatrix} = \frac{1}{1-B_{12}(z)B_{21}(z)} \begin{bmatrix} 1 & -B_{12}(z) \\ -B_{21}(z) & 1 \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} \quad (10)$$

Therefore, the relationship between source signals and recovered signals is

$$\begin{bmatrix} \tilde{S}_1(z) \\ \tilde{S}_2(z) \end{bmatrix} = \frac{1}{1-B_{12}(z)B_{21}(z)} \begin{bmatrix} 1-B_{12}(z)A_{21}(z) & A_{12}(z)-B_{12}(z) \\ A_{21}(z)-B_{21}(z) & 1-B_{21}(z)A_{12}(z) \end{bmatrix} \begin{bmatrix} S_1(z) \\ S_2(z) \end{bmatrix} \quad (11)$$

For the Formulation (10),

If $B_{12}(z) = A_{12}(z)$ and $B_{21}(z) = A_{21}(z)$,

Then

$$\tilde{S}_i(z) = S_i(z), i \in \{1,2\} \quad (12)$$

If $B_{12}(z) = \frac{1}{A_{21}(z)}$ and $B_{21}(z) = \frac{1}{A_{12}(z)}$,

then

$$\tilde{S}_i(z) = A_{ij}(z)S_j(z), i \neq j \in \{1,2\} \quad (13)$$

In the case of the Formulation (13), the filters $B_{ij}(z)$ are infinite impulse response, so only the case of the Formulation (12) is valid. In practice, the mixing filters $A_{ij}(z)$ are unknown, therefore, we must estimate them by a method, which could be a stochastic iteration by maximizing the independence.

$$b_{ij}(n+1,k) = b_{ij}(n,k) + u_n E[f(s_i(n))g(s_j(n-k))], i \neq j \in \{1,2\}, k \in [0,L] \quad (14)$$

where $b_{ij}(n,k)$ is the k-th coefficient of filter $B_{ij}$ at the n-th iteration, $u_n$ is a positive adaptation gain, $f(\cdot)$ and $g(\cdot)$ are non-linear functions, such as $f(x) = x^3$ and $g(x) = x$.

## 2.4. Performance Criteria

The separation performances two aspects:

Residual cross-talking error (RCTE), defined as

$$RCTE(\tilde{s}_i, s_i) = 10 \lg \left\{ \frac{E\left[ (\tilde{s}_i - s_i)^2 \right]}{E(s_i^2)} \right\} \quad (15)$$

This could verify the quality of recovered signals, generally, when its value is less than –20db, we can consider that the recovered signals are correct.

For the computing speed, this paper adopts the rule based on RCTE threshold control criteria [10]

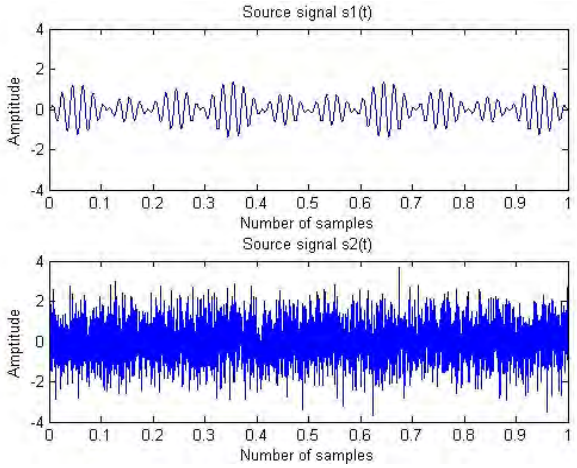



**Figure 6. Source signals.**

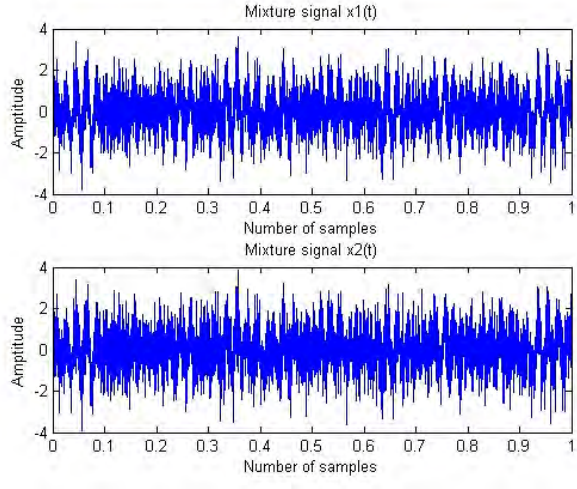



**Figure 7. Observed signals.**

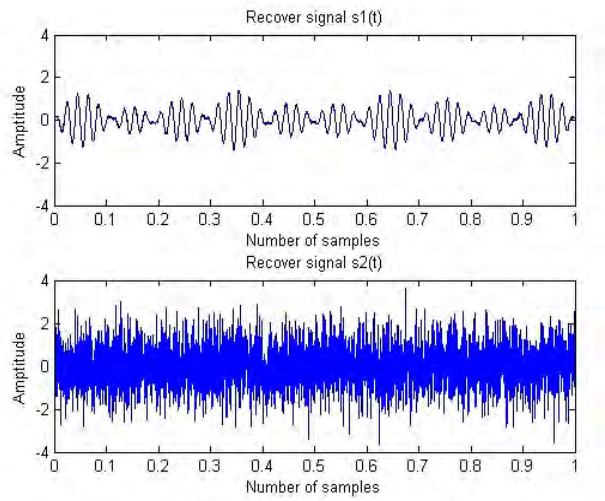



**Figure 8. Recovered signals.**

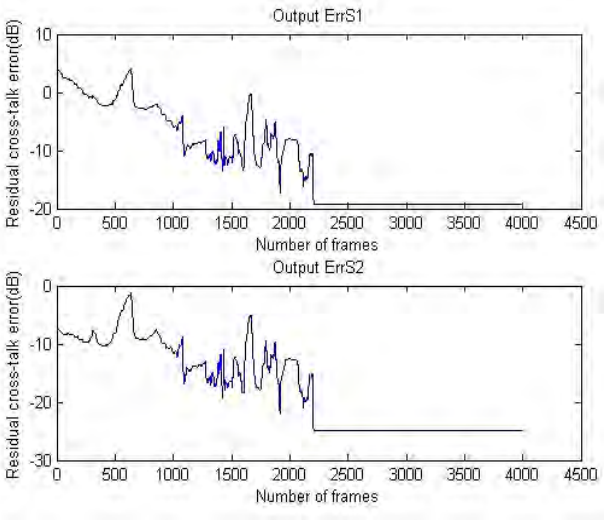



**Figure 9. RCTE between the recovered signals and the source signals.**

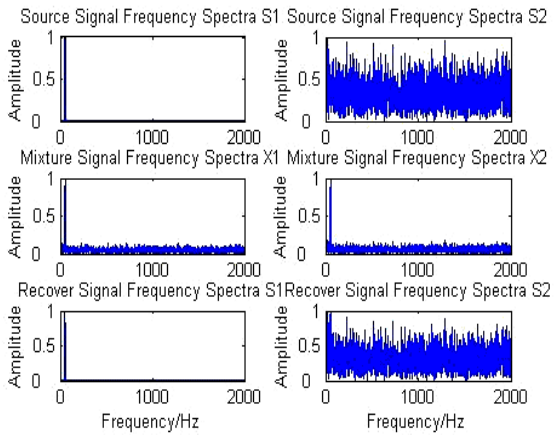



**Figure 10. Frequency spectra of source signals, observed signals and recovered signals.**

$$RCTE_1 + RCTE_2 \le \alpha \tag{16}$$

where $\alpha$ is the goal constant, its value is decided by the separation efficiency and separation purpose.

## 3. Simulation and Results

As previously specified, the aim of BSS is to recover unknown sources with the observations. The purpose of this part is to illustrate the capability of BSS algorithms to separate signals from rotating machine vibration. The rotating machine vibration contains some character, including transient impact, ambient noise [11]. From this point of view, two simulation signals are generated as below.

$$s_1=(\sin(12\pi t)+0.4*\sin(2\pi t))*(\sin(100\pi t+0.5*\sin(0.02\pi t)))$$

$$s_2 = randn(t)$$

The mixture coefficients are acquired using a transfer matrix defined in Equation (6), the mixing filter is

$$A_{12}=[0.8459 \;\; 0.3561 \;\; -0.4625 \;\; 0.0251 \;\; 0.2413 \;\; -0.0866]$$

$$A_{21}=[-0.0599 \;\; 0.0326 \;\; 0.4426 \;\; -0.1977 \;\; -0.0748 \;\; 0.7442]$$

The value of $u_n$ is 0.0016.

The value of goal constant $\alpha$ is –42db.

In this way, we can obtain observed signals as Figure 7, x1 and x2 are the mixture signals. The recovered signals by the BSS algorithm are as Figure 8 shows. The algorithm produces satisfactory separation results in Figure 9, for recovered signals $\tilde{s}_1$ and $\tilde{s}_2$, the RCTE can reach –18.58db and –23.59db respectively. It also shows satisfactory results in Figure 10, the frequency spectrum of recovered signals are almost the same as the sources. It demonstrates that the transient can be extract from the observed signals.

## 4. Conclusions

In this paper, we reviewed the basic theory of BSS for convolutive mixtures, and then analyzed its application to machine vibration, presented a BSS algorithm for convolutive mixtures based on RCTE threshold control criteria. The results of the simulation are favorable. However, for real signals, improvements are necessary for the mixture model and the algorithm. Presently, we are trying to seek some other control criteria for BSS, do research on its application to gearbox condition monitoring and fault diagnosis.

## 5. References

[1] N. Charkani and Y. Deville. “Self-adaptive separation of convolutively mixed signals with a recursive structure, Part I: Stability analysis and optimization of asymptotic behaviour,” Signal Processing, No. 73, pp. 225–254, 1999.

[2] C. Jutten and J. Herault, "Blind separation of sources, Part 1: An adaptive algorithm based on neuromimetic structure," Signal Processing, No. 24, pp. 1–10, 1991.

[3] M. Gaeta, F. Briolle, and P. Esparcieux, "Blind separation of sources applied to convolutive mixtures in shallow water," IEEE Signal Process, Workshop Higher Order Statistics Banff, Canada, pp. 340–343, July 1997.

[4] N. Bouguerriou, M. Haritopoulos, C. Capdessus, and L. Allam, "Novel cycostationarity based blind source separation algorithm using second order statistical properties: Theory and application to the bearing defect diagnosis," Mechanical Systems and Signal Processing, No. 19, pp. 1260–1281, 2005.

[5] G. Gelle, M. Colas, and G. Delaunay, "Blind sources separation applied to rotating machines monitoring by acoustical and vibrations analysis," Mechanical Systems and Signal Processing, Vol. 14, No. 3, pp. 427–442, 2000.

[6] G. Gelle and M. Colas, "Blind sources separation: A tool for rotating machine monitoring by vibrations analysis," Journal of Sound and Vibration, Vol. 248, No. 5, pp. 865–885, 2001.

[7] J. Antoni, "Blind separation of vibration components: Principles and demonstrations," Mechanical Systems and Signal Processing, No. 19, pp. 1166–1180, 2005.

[8] Y. J. Shen and S. P. Yang, "A new Blind-Source-Separation method and its application to fault diagnosis of rolling bearing," International Journal of Nonlinear Sciences and Numerical Simulation, Vol. 7, No. 3, pp. 245–250, 2006.

[9] H. L. Nguyen Thi and C. Jutten, "Blind source separation for convolutive mixtures," Signal Processing, No. 45, pp. 209–229, 1995.

[10] W. Huang, S. Wu, F. Kong, Q. Wu, and P. Li, "Blind source separation for convolutive mixtures and its application to machine vibrations," International Conference on Measuring Technology and Mechatronics Automation, Vol. 1, pp. 598–601, 2009.

[11] Q. He, Z. Feng, and F. Kong, "Detection of signal transients using independent component analysis and its application in gearbox condition monitoring," Mechanical Systems and Signal Processing, No. 21, pp. 2056–2071, 2007.

*Wireless Sensor Network*, **2009, 1, 458-466**
doi:10.4236/wsn.2009.15055 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# A Mobile-Agent-Based Adaptive Data Fusion Algorithm for Multiple Signal Ensembles in Wireless Sensor Networks

**Tianjing WANG[1], Zhen YANG[2], Guoqing LIU[1]**
[1]*College of Science, Nanjing University of Technology, Nanjing, China*
[2]*Institute of Signal and Information Processing, Nanjing University of Posts&Telecommunications, Nanjing, China*
*Email*: *tjingwang@126.com*
*Received June* 8, 2009; *revised August* 17, 2009; *accepted August* 18, 2009

## Abstract

Distributed Compressed Sensing (DCS) is an emerging field that exploits both intra- and inter-signal correlation structures and enables new distributed coding algorithms for multiple signal ensembles in wireless sensor networks. The DCS theory rests on the joint sparsity of a multi-signal ensemble. In this paper we propose a new mobile-agent-based Adaptive Data Fusion (ADF) algorithm to determine the minimum number of measurements each node required for perfectly joint reconstruction of multiple signal ensembles. We theoretically show that ADF provides the optimal strategy with as minimum total number of measurements as possible and hence reduces communication cost and network load. Simulation results indicate that ADF enjoys better performance than DCS and mobile-agent-based full data fusion algorithm including reconstruction performance and network energy efficiency.

**Keywords:** Wireless Sensor Networks, Mobile Agent, Compressed Sensing, Distributed Compressed Sensing, Joint Sparsity, Joint Reconstruction

## 1. Introduction

Wireless Sensor Networks (WSN) are an emerging technology that promises an ability to monitor the physical world via spatially distributed networks of small and inexpensive wireless sensor nodes that have the ability to self-organize into a well-connected network. The communication tasks consume the limited power available at such sensor nodes and, therefore, in order to ensure their sustained operations, the power consumption must be kept to minimum. Different from transmission cost, the computational cost may be negligible for some applications. For example, WSN monitoring field temperature may use simple functions which essentially are of insignificant cost. Consequently, a major challenge in the design of WSN is developing schemes that extract relevant information about the sensor data with the minimum energy consumption, especially with transmission cost. In order to reduce transmission of sensor data, a new framework for single signal sensing and compression has developed recently under the rubric of Compressed Sensing (CS) [1–3]. The implications of CS are promising for many applications, especially sensing signals that have a sparse representation in some basis [4–6]. Based on the intra-signal correlations (between samples in each signal), CS can perfectly reconstruct a compressible signal from remarkably few linear measurements. In WSN, however, a large number of sensor nodes presumably observe related phenomena and are programmed to perform a variety of signals acquisition tasks. These signals have high correlations and need to be jointly processed, so the independently encoding and decoding theory and practice of single signal in a CS framework can not satisfy such applications. Fortunately, the ensemble of signals that sensor nodes acquired can be expected to possess some joint structure, or inter-signal correlations (between samples across signals), in addition to the intra-signal correlation of single signal. Most existing coding algorithms [7,8], however, exploit only inter-signal correlations and not intra-signal correlations. A new Distributed Compressed Sensing (DCS) theory exploits both intra-signal and inter-signal correlation structures of multiple signal ensembles [9–11]. In a typical DCS scenario, each individual node independently encodes its signal by CS framework and then transmits

*Supported by the Young Teacher Academic Fund of Nanjing University of Technology

the measurements to a distant Sink node (Sink) by multiple skips. Under the right conditions, Sink can jointly reconstruct multiple signal ensembles precisely. However, we note that the network traffic may be extremely heavy in DCS, resulting in poor performance of the system, when the number of sensor nodes is big and the amount of sensing signals is large. Furthermore, nodes can not know the suited minimum number of measurements that they need to transmit to Sink under DCS framework. In order to guarantee perfect reconstruction, each node has to transmit enough measurements. This means that DCS only utilizes the inter-signal correlations in jointly decoding processes, but not in jointly encoding processes.

In this paper, we design a mobile-agent-based Adaptive Data Fusion (ADF) algorithm for multiple signal ensembles which is inspired by DCS. Instead of passing large amounts of independently encoding measurements by DCS in each individual node over the network, Mobile Agent (MA) can fuse multiple signals from node to node along a shortest path, based on Global Closest First (GCF) heuristics algorithm [12]. According to the sparse property of single signal and the joint sparsity of multiple signal ensembles under the results of data fusion, each node can determine the minimum number of measurements needed to transmit to Sink and effectively reduces the transmission cost. Extensive experiments in session 4 demonstrate that the energy efficiency and the reconstruction performance of ADF are more excellent than DSC and mobile-agent-based Full Data Fusion (FDF) algorithm.

The organization of this paper is as follows. Section 2 overviews a joint sparsity model in DCS. Single signal and multiple signals reconstruction methods are discussed respectively. Section 3 introduces our two mobile-agent-based data fusion algorithms: Full Data Fusion (FDF) and Adaptive Data Fusion (ADF). Detailed theoretical analyses indicate that FDF and ADF are more energy efficient than DCS, sufficiently utilizing intra-signal and inter-signal correlation of multiple signal ensembles. Experiments in Section 4 confirm that ADF indeed reduces large amounts of total transmission cost and the number of total measurements compared to DCS and FDF. Section 5 describes conclusions.

## 2. Distributed Compressed Sensing

### 2.1. Joint Sparsity Model

The recently introduced theory of DCS enables a new distributed coding algorithm that exploits both intra- and inter-signal correlation structures of multiple signals [9]. In this paper, we focus on a Joint Sparsity Models (JSM, sparse common component +innovations), which can be improved by mobile-agent-based data fusion. For example, a practical situation well-modeled by the JSM is a group of sensors $\{S_1,\ldots S_J\}$ measuring temperatures at a number of outdoor locations throughout the day. The temperature readings $x_l(l\in\{1,2,\ldots J\})$ have both temporal (intra-signal) and spatial (inter-signal) correlations. Global factors, such as the sun and prevailing winds, could have an effect that is both common to all sensors and structured enough to permit sparse representation in some basis. More local factors, such as shade, water, or animals, could contribute localized innovations that are also structured (and hence sparse in some basis). A similar scenario could be imagined for a network of sensors recording light intensities, air pressure, or other phenomena. All of these scenarios correspond to measuring properties of physical processes that change smoothly in time and in space and thus are highly correlated.

We adopt language and notation from [9]. Assume that there exists a known basis $\Psi=\{y_i \mid y_i\in R^m, i=1,\mathbf{L},m\}$ in which a signal $x_l=\{x_l(1),\mathbf{L},x_l(m)\}^T\in R^m(l\in\{1,2,\mathbf{L},J\})$ can be sparsely represented as $x_l=\Psi q_l$, where $q_l=(q_l(1),\mathbf{L},q_l(m))^T$ is a sparse coefficient vector and $\| q_l \|_0=k$. Thus, a compressible multiple signal ensemble $x_1,\mathbf{L},x_J$ shares a common sparse component while each single signal contains an innovation sparse component. That is

$$q_l=q_C+\hat{q_l},\quad l\in\{1,2,\mathbf{L},J\} \tag{1}$$

with

$$x_C=\Psi q_C,\| q_C \|_0=k_C$$

and

$$\hat{x}_l=\Psi\hat{q_l},\| \hat{q_l} \|_0=k_l\quad (k_l<k_C) \tag{2}$$

where $k_c$ is a common sparse parameter of $\theta_c$ and $k_l$ is an innovation sparse parameter of $\theta_l$.

Thus, the signal $x_c$ is common to all of $x_l(l\in\{1,2,\mathbf{L},J\})$ and has sparse coefficient vector $\theta_c$ in basis $\psi$, and the signal $\hat{x}_l(l\in\{1,2,\mathbf{L},J\})$ is the unique portions of $x_l$ and has sparse coefficient vector $\hat{q_l}$ in the same basis. Denote that $\Omega_c$ is a tight support set of the nonzero values in $\theta_c$ and $\Omega_l$ is a tight support set of the nonzero values in $\hat{q_l}$.

To make linear measurements, denote the measurement matrix $\Phi_l=(j_{ij})_{n_l\times m}\quad (l\in\{1,2,\mathbf{L},J\})$ for the multisignal ensembles, where a second basis matrix $\Phi_l$ is incoherent with $\Psi$. Thus, a small number of noiseless measurements $y_l=\Phi_l x_l=(y_l(1),\quad \mathbf{L},y_l(n_l))^T(l\in\{1,2,\mathbf{L},J\})$ contain sufficient information for approximate reconstruction [1,2]. Mathematically, this can be reduced to a standard linear algebra problem: we wish to find

$x_l (l \in \{1,2,\mathbf{L},J\})$ which explains the measurements $y_l = \Phi_l x_l = \Phi_l \Psi q_l \quad (l \in \{1,2,\mathbf{L},J\})$.

### 2.2. Joint Reconstruction of Multiple Signal Ensembles via $l_1$ Optimization

To give ourselves a firm footing for analyses, we firstly consider single signal reconstruction based on the CS theory, mainly to exploit intra-signal structure at a single node. CS encodes a workable approximation of the single compressible signal $x_l (l \in \{1,2,\mathbf{L},J\})$ with $n_l = m$ sampling resources and signal reconstruction can be achieved by Basis Pursuit (BP) [1,2] or Orthogonal Matching Pursuit (OMP) [3]. OMP allows faster reconstruction at the expense of more measurements.

Formally, BP needs to solve the following $l_1$ optimization problem

$$(\mathrm{P}_1) \min \| q_l \|_1 \text{ subject to } y_l = \Phi_l x_l = \Phi_l \Psi q_l \ (l \in \{1,2,\mathbf{L},J\}) \quad (3)$$

The simulation results in [2] state that there exists measurements $n \geq ck$ (oversampling factor $c \geq 4$) are required to reconstruct $x_l$ with high probability, using linear programming methods e.g. interior point method and simplex method. That is, an optimal reconstructed signal $x_l^* = \Psi q_l^*$ can be achieved by an optimal sparse solution $q_l^*$ of the problem $(\mathrm{P}_1)$.

In addition to single signal encoding and decoding methods, the jointly encoding and decoding methods of multiple signal ensembles are considered in DCS. DCS expects that 1) $c(k_C + k_l)$ $(l \in \{1,2,\mathbf{L},J\})$ measurements suffice to reconstruct $x_1$, 2) $c(k_C + \sum_{l=1}^{J} k_l)$ measurements suffice to reconstruct multiple signal ensembles $x_1,\ldots, x_J$, for there exists $k_C + \sum_{l=1}^{J} k_l$ nonzero elements in $x_1,\ldots, x_J$. Furthermore, the recovery problem can be formulated using matrices and vectors as

$$q = \begin{bmatrix} q_C \\ q_1 \\ \mathbb{M} \\ q_J \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \mathbb{M} \\ x_J \end{bmatrix}, y = \begin{bmatrix} y_1 \\ \mathbb{M} \\ y_J \end{bmatrix},$$

$$\overset{\Downarrow}{\Phi} = \begin{bmatrix} \Phi_1 & \mathbf{L} & 0 \\ & \mathbf{O} & \\ 0 & \mathbf{L} & \Phi_J \end{bmatrix}, \quad \overset{\Downarrow}{\Psi} = \begin{bmatrix} \Psi & \Psi & \mathbf{L} & 0 \\ \mathbb{M} & & \mathbf{O} & \\ \Psi & 0 & & \Psi \end{bmatrix} \quad (4)$$

In order to sufficiently utilize inter-signal correlation of multiple signal ensembles, we assume that $\Phi_1 = \mathbf{L} = \Phi_J = \Phi$ and then $\overset{\Downarrow}{\Phi}$ can be rewritten as $\overset{\Downarrow}{\Phi} = diag(\Phi, \mathbf{L}, \Phi)$. It is possible to let Sink previously send the same random seed to all sensor nodes in the interesting field and then the same pseudorandom matrix $\Phi$ can be generated using simple algorithm with seed at each node.

With sufficient sampling, DCS can reconstruct multiple signal ensembles by solving the following $l_1$ optimization problem

$$(\mathrm{P}_2) \quad \min \| q \|_1 \quad \text{subject to} \quad y = \overset{\Downarrow}{\Phi} \overset{\Downarrow}{\Psi} q \quad (5)$$

Due to the optimal spare solution $q^*$ of the problem $(\mathrm{P}_2)$, we can get the corresponding reconstructed multi-signal ensembles $x^* = \overset{\Downarrow}{\Psi} q^*$.

## 3. Mobile-Agent-Based Adaptive Data Fusion Algorithm

The DCS theory proposes a framework for joint reconstruction of compressible multi-signal ensembles. However, each single node independently encodes in DCS, which does not sufficiently utilize the joint sparsity of multi-signal ensembles. This operation makes each node have to transmit a lot of measurements. In this session, we focus on reducing measurements required to transmit at each node by mobile-agent-based data fusion in WSN.

A WSN under a DCS framework, as shown in Figure 1, consists of three types of components: Sink, sensor nodes and communication network. With energy restriction, sensor nodes can not directly communicate with Sink. For example, the encoding results of a node $S_{l+1}$ in the interesting field are transmitted to Sink by multi skips routing in Figure 1. Other nodes do the same works. In Figure 1, we model a WSN as a graph $G=(V,E)$, where

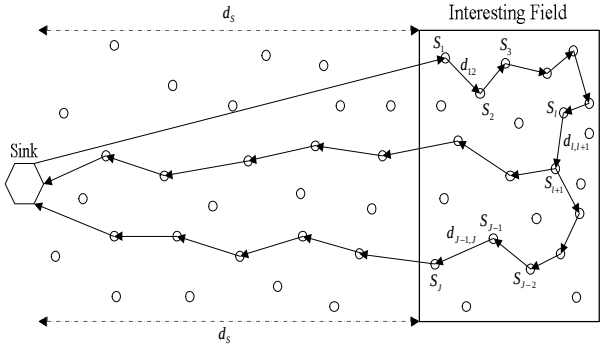



**Figure 1. A WSN under DCS framework. Circles denote sensor nodes. The communication distance between node $S_l$ and node $S_{l+1}(l \in \{1,\mathbf{L},J-1\})$ is $d_{l,l+1}$ and the communication distance between Sink and $S_l$ is approximately as $d_S$ by the shortest multiple skips routing. For being simple, other links representing the communication links among nodes and Sink are omitted. On the other hand, a route $\{S_S, S_1, \mathbf{L}, S_J, \mathbf{L}, S_S\}$ represents MA data fusion routing.**

$V=\{S_S,S_1,\mathbf{L},S_L\}$ and $E$ denotes an edge set representing the communication links between node-pairs or links between Sink and nodes. In many applications, a distant Sink $S_S$ retrieves relevant interesting field information from the sensor nodes. Let $d_{Sl}(l\in\{1,\mathbf{L},L\})$ be the total communication distance between Sink and a node $S_l(l\in\{1,\mathbf{L},L\})$ by the shortest multiple skips routing. For being simple, Sink is assumed to be far away from the interesting field so that $d_{S1}\approx\mathbf{L}\approx d_{SL}\approx d_S$ and $d_S$ **?** $d_{l,l+1}(l\in\{1,\mathbf{L},L-1\})$, where $d_{l,l+1}$ denotes the communication distance between node-pairs $S_l$ and $S_{l+1}$ [13].

We next describe the communication architecture of WSN in Figure 1 to motivate the formulation of two data fusion algorithms for multiple signal ensembles. Assume that a connected subgraph $G'=(V',E')\subseteq G$ is found, where $V'$ contains Sink and encode nodes, i.e., $V'=\{S_S,S_1,\mathbf{L},S_J\}\subset V\ \ (I<H,J<L)$, and $E'$ denotes a edge set representing all communication links in $V'$. For a node $S_l(l\in\{1,\mathbf{L},J\})$, its node weight $w(S_l)$ denotes the amount of data outgoing from $S_l$. An edge $e_l\in E(l\in\{1,\mathbf{L},J-1\})$ is denoted by $e_l=(S_l,S_{l+1})$. The weight of edge $e_l$ is equivalent to the weight of $S_l$, i.e., $w(e_l)=w(S_l)$. Two metrics, $t(e_l)$ and $f(e_l)$, are associated with each edge, describing the transmission cost and fusion cost on the edge, respectively. In many WSN applications, however, the fusion cost may be negligible. For being simple, we do not consider fusion cost in this paper.

The unit cost of the link for transmitting data from $S_1$ to $S_{l+1}$ is abstracted as $c(e_l)=b\,d_{l,l+1}^r+e$, where $\beta$ and $r$ are tunable parameters based on the radio propagation [14]. Thus the transmission cost $t(e_l)$ is

$$t(e_l)=c(e_l)w(e_l) \tag{6}$$

Similarly, we approximately define $e_{Sl}(l\in\{1,\mathbf{L},J\})$ is the communication links between a node $S_l$ and Sink by the shortest multiple skips routing, and then the transmission cost is approximately as $t(e_{Sl})=c(e_{Sl})$ $w(e_{Sl})(l\in\{1,\mathbf{L},J\})$, where $c(e_{Sl})$ and $w(e_{Sl})$ are the unit transmission cost and the weight of $e_{Sl}$. Obviously, we can get $c(e_{Sl})$ **?** $c(e_l)$, for $d_S$ **?** $d_{l,l+1}$.

From above definitions, the total network energy consumption in $G'=(V',E')$ with different computing models can be calculated. We firstly consider total network energy consumption of DCS. According to encoding method of a single signal, we assume that individual node $S_l(l\in\{1,\mathbf{L},J\})$ can gain a sparse coefficient vector $q_l(l\in\{1,\mathbf{L},J\})$ by projecting a sensing signal $x_l(l\in\{1,\mathbf{L},J\})$ into a basis matrix $\Psi$. This operation may consume some computational energy, but it does not affect the total network energy consumption and can be omitted compared to transmission cost.

To conveniently explain the network energy consumption, we assume that $\Omega_C\cap\Omega_l=\varnothing$. Single signal reconstruction, therefore, needs $n_l$ measurements with $n_l=c(k_c+k_l)$ in a CS framework, and then the total network energy consumption of DCS can be calculated as follows

$$C_{DCS}=\sum_{l=1}^{J}c(e_{Sl})w(e_{Sl})=\sum_{l=1}^{J}(b\,d_S^r+e)c(k_C+k_l) \tag{7}$$

where the number of measurements $n_l=c(k_C+k_l)$ directly denotes the weight $w(e_{Sl})$.

We consider the network energy consumption of two mobile-agent-based data fusion algorithms. Generally speaking, MA is a special kind of software with small size, whose transmission cost can be omitted compared to transmission cost of larger amount of sensing information. We will not consider MA transmission cost. Sink predetermined the MA routing $\{S_S,S_1,\mathbf{L},S_J,\mathbf{L},S_S\}$ by GCF. The detailed mobile-agent-based data fusion process is presented as follows.

In initialize stage, MA migrates to$S_1$and obtains measurements $y_1$ with $n_1=c\,\|q_1\|_0$. Subsequently, it migrates to $S_2$ and finds $\theta_c\hat{q}_1$ and $\hat{q}_2$ by contrasting sparse coefficients $\theta_1$ and $\theta_2$. Thus, it can calculate the common measurements $y_c$ corresponding to the common sparse component $\theta_1$ of the signal ensembles. This means that measurements $y_1$ and $y_2$ can be divided into $y_1=y_C+\hat{y}_1$ and $y_2=y_C+\hat{y}_2$, where $\hat{y}_1$ and $\hat{y}_2$ correspond to the sparse innovation component $\hat{q}_1$ and $\hat{q}_2$. MA carrying measurements $y_C$, $\hat{y}_1$ and $\hat{y}_2$ with $w(e_2)=c(k_C+k_1+k_2)$ migrates to $S_3$. We then repeat above process on remaining nodes along MA routing until MA returns to Sink. Such process is a typical mobile-agent-based data fusion process. In this paper, we term this process a mobile-agent-based Full Data Fusion (FDF) algorithm compared to the following adaptive data fusion process. The total network energy consumption of FDF is shown as follows

$$\begin{aligned}C_{FDF}&=\sum_{l=1}^{J-1}c(e_l)w(e_l)+c(e_{SJ})w(e_{SJ})\\&=\sum_{l=1}^{J-1}(b\,d_{l,l+1}^r+e)c(k_C+k_1+\mathbf{L}+k_l)\\&\quad+(b\,d_S^r+e)c(k_C+k_1+\mathbf{L}+k_J)\end{aligned} \tag{8}$$

We interest in comparing the total energy consumption

of DCS and FDF. The total transmission cost can be calculated as follows

$$C_{DCS} - C_{FDF} = (b\,d_S^r + e)(J-1)ck_C - \sum_{l=1}^{J-1}(b\,d_{l,l+1}^r + e)ck_l$$
$$= \sum_{l=1}^{J-1}((b\,d_S^r + e)ck_C - (b\,d_{l,l+1}^r + e)ck_l) \quad (9)$$

Note that $d_S$ **?** $d_{l,l+1}$ and $k_C > k_l$, it is easy to get

$$C_{DCS} > C_{FDF} \quad (10)$$

As expected, the inequality (10) means that the total transmission cost of DCS is much larger than FDF. By analyzing FDF in detail, we find another challenge that MA will carry more and more amount of data fusion results along the MA routing. However, data fusion at a node $S_l$ ($l \in \{2, \mathbf{L}, J\}$) only needs the common measurements $y_C$. If MA still carries all front innovation measurements $\hat{y}_1, \mathbf{L}, \hat{y}_{l-1}$, it is not in favor of saving communication cost. This result brings a question whether we can avoid transmitting an amount of the innovation measurements. In this regard, we establish the following data fusion algorithm named mobile-agent-based Adaptive Data Fusion (ADF) algorithm. Different from in FDF, MA only carries the common measurements $y_C$ in ADF. Concretely, MA can calculate $\hat{q}_l$ ($l \in \{2, \mathbf{L}, J\}$) by contrasting $q_C$ and $q_l$ ($l \in \{2, \mathbf{L}, J\}$), when it carrying $y_C$ migrates to $S_l$ ($l \in \{2, \mathbf{L}, J\}$). This allows $y_l$ ($l \in \{2, \mathbf{L}, J\}$) to be divided into $y_l = y_C + \hat{y}_l$ in which the number of the innovation measurement $\hat{y}_l$ is $\hat{n}_l = ck_l$. MA carrying measurements $y_C$ with $w(e_l) = ck_C$ continuously migrates to $S_{l+1}$. On the other hand, measurements $\hat{y}_l$ are directly transmitted to Sink. We then repeat above process on remaining nodes along the same MA routing as FDF until MA returns to Sink. Then, the total network energy consumption of ADF is expressed as follows

$$C_{ADF} = (b\,d_{12}^r + e)c(k_C + k_1)$$
$$+ \sum_{l=2}^{J-1}((b\,d_{l,l+1}^r + e)ck_C + (b\,d_S^r + e)ck_l) + (b\,d_S^r + e)c(k_C + k_J) \quad (11)$$

We also attend to compare the total energy consumption of FDF and ADF as follows

$$C_{FDF} - C_{ADF} = \sum_{l=1}^{J-1}(b\,d_{l,l+1}^r + e)ck_l > 0 \quad (12)$$

$$\text{i.e.,}\quad C_{FDF} > C_{ADF} \quad (13)$$

From (10) and (13), it can be shown that

$$C_{DCS} > C_{FDF} > C_{ADF} \quad (14)$$

From (14), we can easily obtain the benefits of ADF. Firstly, based on DCS, ADF also sufficiently utilizes both temporal (intra-signal) and spatial (inter-signal) correlations of multi-signal ensembles to analyze single signal sparsity structure and multi-signal ensembles jointly sparsity structure. These sparsity structures make it possible to perform data fusion of multi-signal ensembles. Second, the innovation measurements are allowed to directly transmit to Sink, while MA only carries the common measurements. It benefits reducing transmission cost. So we can say that ADF provides the optimal strategy for minimizing total transmission measurements and transmission cost compared to DCS and FDF.

According to the above discussion, we can reconstruct multi-signal ensembles by solving the following $l_1$ optimization problem

$$(\mathrm{P}_3)\quad \min \|q\|_1 \quad \text{subject to} \quad \hat{y} = \hat{\Phi}\hat{\Psi}q \quad (15)$$

$$\text{where } q = \begin{bmatrix} q_C \\ q_1 \\ \mathbf{M} \\ q_J \end{bmatrix},\quad \hat{y} = \begin{bmatrix} y_C \\ \hat{y}_1 \\ \mathbf{M} \\ \hat{y}_J \end{bmatrix},\quad \hat{\Phi} = \begin{bmatrix} \Phi_C & 0 & \mathbf{L} & 0 \\ 0 & \Phi_1 & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{L} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & \mathbf{L} & \Phi_J \end{bmatrix},$$
$$\hat{\Psi} = \begin{bmatrix} \Psi & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 0 & \mathbf{L} & \Psi \end{bmatrix}.$$

We can obtain $x^* = (x_C^*, \hat{x}_1^*, \mathbf{L}, \hat{x}_J^*)^T$ with an optimal sparse solution $q^* = (q_C^*, q_1^*, \mathbf{L}, q_J^*)^T$, where $x_C^* = \Psi q_C^*$ and $\hat{x}_l^* = \Psi \hat{q}_l^*, (l \in \{1, \mathbf{L}, J\})$. Furthermore, multi-signal ensembles can be reconstructed by $x_l^* = x_C^* + \hat{x}_l^*$, $(l \in \{1, \mathbf{L}, J\})$.

The above results focus on theoretical analyses of saving transmission cost in ADF. On the other hand, we are interested in comparing the joint reconstruction performance of DCS, FDF and ADF. The following simulation results are presented illustrating the better joint reconstruction performance of ADF.

## 4. Simulation

In our setup, sensor nodes are randomly distributed in a region of a $50m \times 50m$ square. The distance between Sink and the interesting field is $d_S = 400m$. When considering transmission cost, we set $b = 100pJ/bit/m^2$, $r = 2$ and $e = 100nJ/bit$ [12] in (6). Furthermore, we consider a series of example multiple signal ensembles $x_1, \mathbf{K}, x_J$ that satisfy the conditions of joint sparsity model. The signal components $x_C, \hat{x}_1, \mathbf{K}, \hat{x}_J$ are assumed to be sparse in Discrete Cosine Transform (DCT) matrix $\Psi$ with sparse parameters $k_C, k_1, \mathbf{K}, k_J$, re-

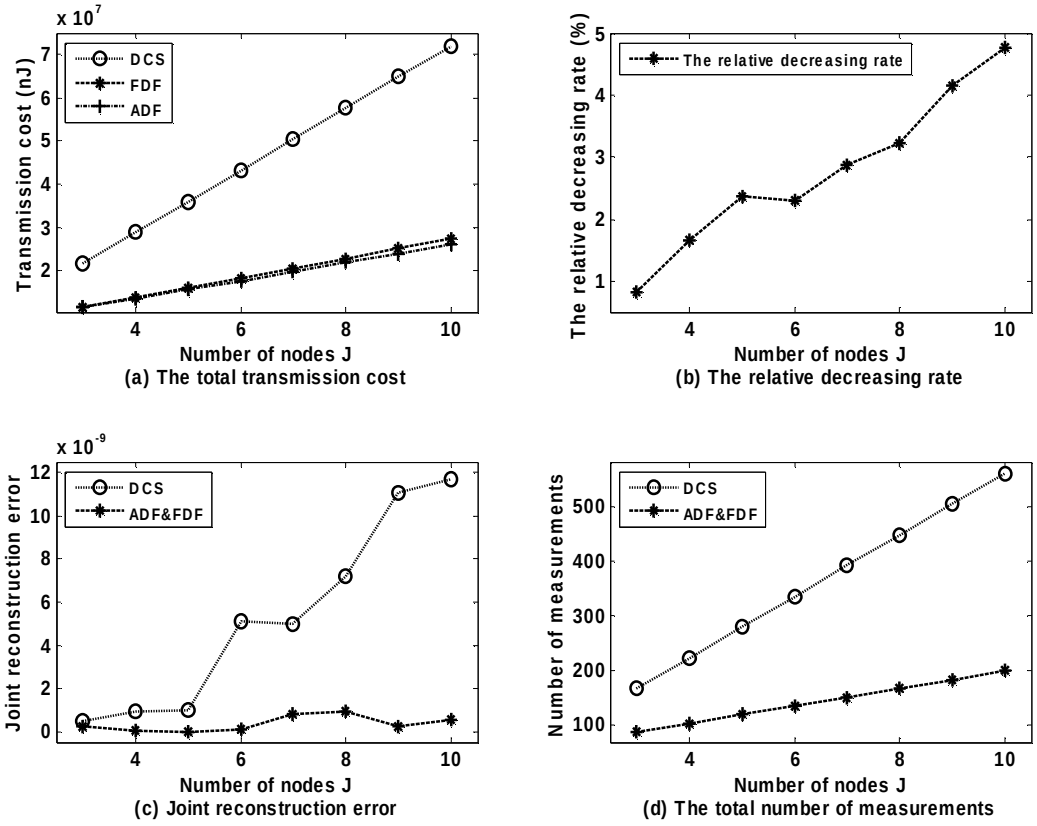



**Figure 2. Effect of the number of nodes on joint reconstruction performance of multiple signal ensembles. We choose signals with** $m = 50$, $k_C = 10,\ k_l = 4(l \in \{1, \mathbf{L}, J\})$ **and** $n_C = 4k_C = 40,\ \hat{n}_l = 4k_l = 16\ (l \in \{1, \mathbf{L}, J\})$**. (a) The total network transmission cost, (b) The relative decreasing rate, (c) Joint reconstruction error, (d) The total number of measurements.**

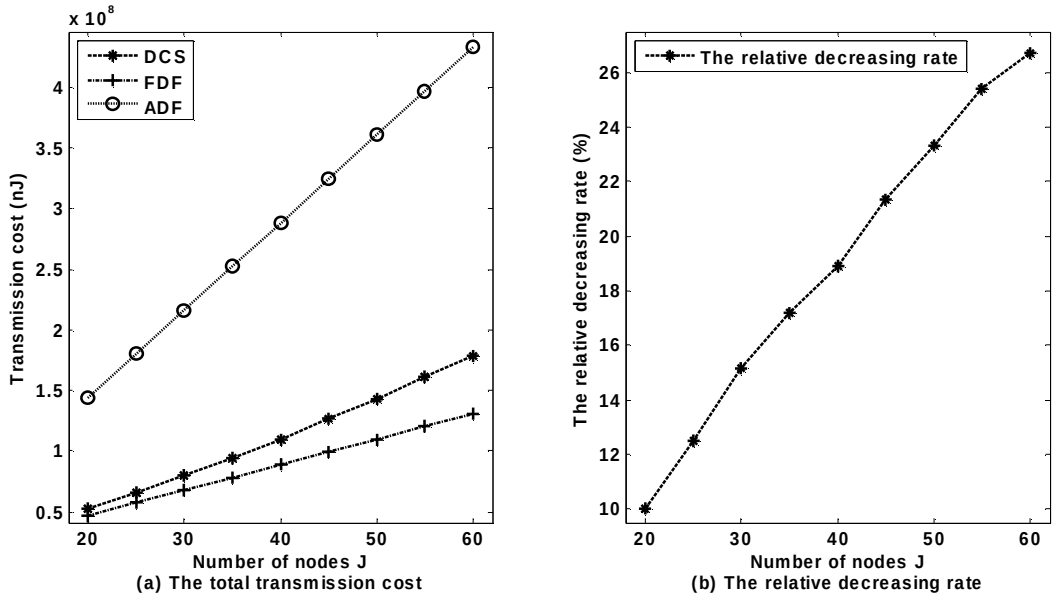



**Figure 3. Effect of the number of nodes on energy consumption of multiple signal ensembles. We choose signals with** $m = 50$, $k_C = 10,\ k_l = 4(l \in \{1, \mathbf{L}, J\})$ **and** $n_C = 4k_C,\quad \hat{n}_l = 4k_l (l \in \{1, \mathbf{L}, J\})$**. (a) The total transmission cost, (b) The relative decreasing rate.**

spectively. We assign random Gaussian values to the nonzero coefficients $q_C, \hat{q}_1, \mathbf{K}, \hat{q}_J$, and the locations of nonzero are chosen at random. As a measure of the reconstruction performance, the joint reconstruction error $e = \sum_{l=1}^{J} \| x_l - x_l^* \|_2$ is designed. The Interior Point (IP) method in "Matlab" is used to solve the problem ($P_2$) and the problem ($P_3$).

Our first experiment chooses signals with the length $m$=50 and sparse parameters $k_C = 10,\ k_l = 4 (l \in \{1, \mathbf{L}, J\})$. Then the corresponding numbers of measurements are chosen by $n_C = ck_C = 40,\ \hat{n}_l = ck_l = 16 (l \in \{1, \mathbf{L}, J\})$, where $c = 4$. Without loss of generality, assume that one measurement produces 8 bit packet [12]. With increasing number of nodes $J$, the total number of measurements of DCS is greatly larger than FDF and ADF in Figure 2(d). This result causes the transmission cost of DCS also greatly larger than FDF and ADF in Figure 2(a). To further illustrate the advantage of ADF, we consider the relative decreasing rate calculated as $t = \frac{\text{TC(FDF)-TC(ADF)}}{\text{TC(FDF)}}$ in which TC(FDF) and TC(ADF) are the total transmission cost of FDF and ADF, respectively. Figure 2(b) clearly shows that the relative decreasing rate linearly increases with *J*. This means that the energy efficiency of ADF is more distinctness with increasing number of nodes. In Figure 2(c), we emphasize on comparison of reconstruction performance between DCS and ADF (FDF). ADF and FDF enjoy less joint reconstruction errors than DCS, though ADF and FDF use less number of measurements than DCS. So we can say that ADF performs much better than DCS and FDF.

WSN typically consists of a large number of sensor nodes, so we need consider energy consumption of much more nodes in DCS, FDF and ADF to further observe the advantage of ADF. We repeat the first experiment while the number of nodes varies from 20 to 60. As expected, the transmission cost of DCS is further larger than FDF and ADF as $J$ increases in Figure 3(a). Comparing Figure 3(b) with Figure 2(b), we note that the relative decreasing rate scales linearly with $J$. The energy savings of ADF can be as large as 27%. These results identify that ADF is an optimal strategy with minimum total number of measurements and total transmission cost, which consist with the front theoretical conclusions in Section 3.

Experiments in Figure 3 bring another question whether we can guarantee better joint reconstruction performance as the number of nodes *J* increases. In our joint decoding simulations, we find that computational time and complexity will greatly increase as *J* increases. So measurements in Sink should be grouped according to applications. This operation consists with the idea in [9]. In the next experiment, we use *J*=40 nodes and their measurements are separated in 5 groups. Average reconstruction error $\bar{e} = (\sum_{t=1}^{5} e_t)/5$ is designed to measure the

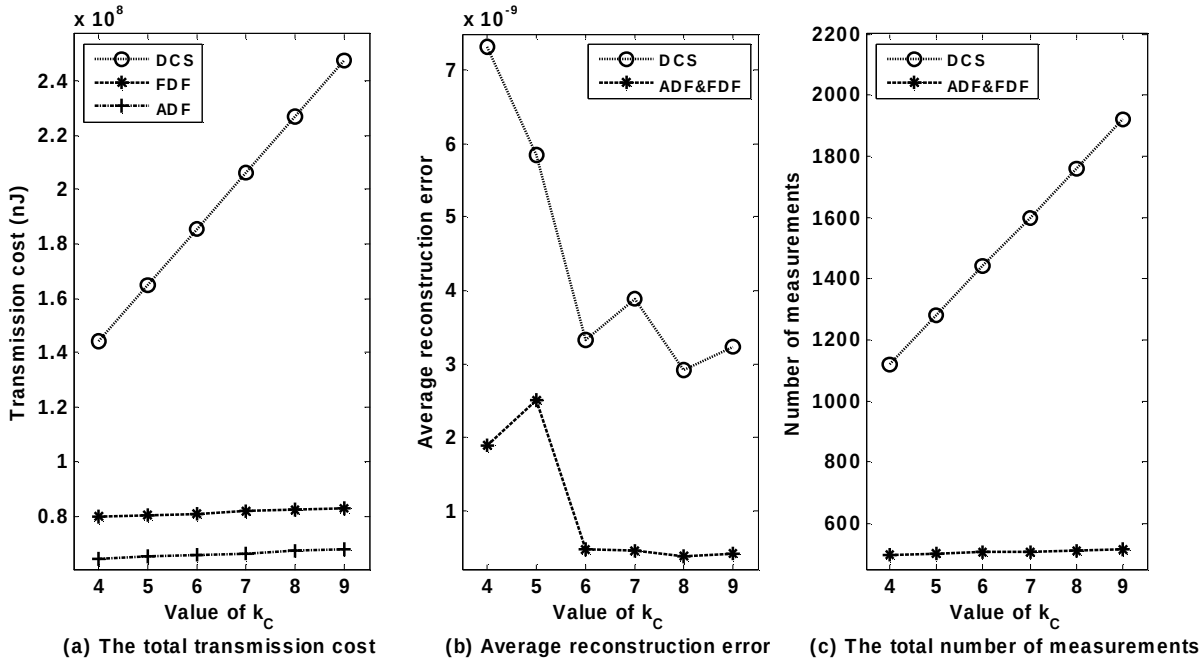



**Figure 4. Effect of the number of common sparse parameters $k_c$ on joint reconstruction of multiple signal ensembles. We choose $m = 50$, fix $k_l = 3 (l \in \{1, \mathbf{L}, J\})$, and vary common $k_c$ from 4 to 9 and then choose $n_C = 4k_C, \hat{n}_l = 4k_l (l \in \{1, \mathbf{L}, J\})$. (a) The total transmission cost, (b) Average reconstruction errors, (c) The total number of measurements.**

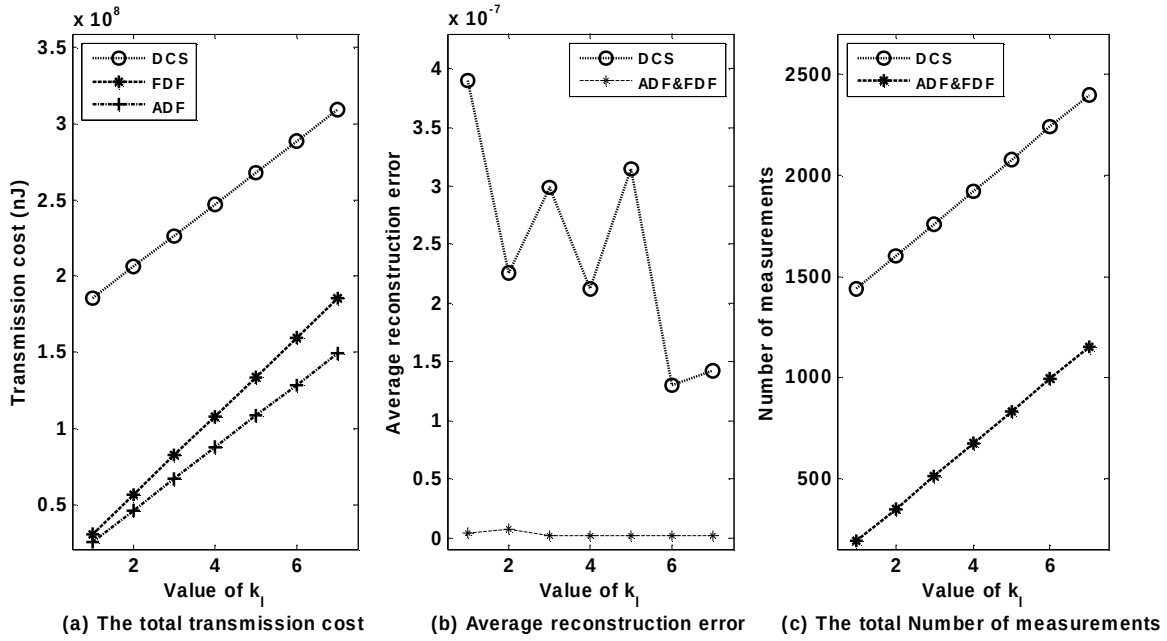



**Figure 5. Effect of the number of innovation sparse parameters $k_l$ on joint reconstruction of multiple signal ensembles. We choose $m = 50$, fix $k_C = 8$ and vary $k_l (l \in \{1, \mathbf{L}, J\})$ from 1 to 7, and then choose $n_C = 4k_C$, $\hat{n}_l = 4k_l (l \in \{1, \mathbf{L}, J\})$. (a) The total transmission cost, (b) Average reconstruction errors, (c) The total number of measurements.**

reconstruction performance, where $e_t (t = 1, \mathbf{L}, 5)$ is the joint reconstruction error of every group. We perform experiments with the length of signals $m = 50$ and innovation sparse parameters $k_l = 3$ ($l \in \{1, \mathbf{L}, J\}$), while the common sparse parameter $k_c$ varies from 4 to 9. The obtained results in Figure 4 (a)-(c) show the similar conclusions in Figure 2 and Figure 3. The number of common sparse parameters $k_c$ greatly affects the total transmission cost of DCS while not FDF and ADF, for DCS need each node to transmit the common measurements while FDF and ADF avoid this operation by data fusion. This sufficiently reveals the advantage of mobile-agent-based data fusion algorithms. Moreover, the average reconstruction error of DCS, FDF and ADF decrease benefiting from the increasing number of measurements. This means joint reconstruction performance can be improved by increasing the total number of mea- surements.

The finally experiments focus on effect of the number of innovation sparse parameters $k_l$ on joint reconstruction of multiple signal ensembles. We repeat the front experiments with the length of signals $m$=50 and the common sparse parameter $k_c$=8, while the innovation sparse parameter $k_l$ ($l \in \{1, \cdots J\}$) varies from 1 to 7. As $k_l$ ($l \in \{1, \cdots J\}$) increasing, the total transmission cost and the number of measurements of FDF and ADF fleetly increase compared to Figure 4. Figure 5(a) and Figure 5(c) reveal that the performance of data fusion is influenced by the innovation sparse parameters $k_l$, for $k_l$ represent the differences among multi-signal ensembles. At the expense of more measurements and energy cost, we can obtain multi-signal ensembles with more details. At the same time, we gain the better joint reconstruction performance in Figure 5(b), for the total number of measurements increases.

As can be seen, above experiments imply that ADF sufficiently takes advantage of intra- and inter-signal correlation of multi-signal ensembles by mobile-agent-based data fusion. So ADF enjoys better performances than DCS and FDF.

## 5. Conclusions

Distributed Compressed Sensing (DCS) extends the theory and practice of Compressed Sensing (CS) to multi-signal ensembles. A joint sparsity model for multi-signal ensembles with both intra- and inter-signal correlation captures the essence of real physical scenarios. This paper provides a new mobile-agent-based Adaptive Data Fusion (ADF) algorithm. ADF can greatly reduce the total number of measurements for successful joint reconstruction compared with DCS. Moreover, ADF can greatly reduce transmission cost and network load. Extensive experiments demonstrate that it indeed leads to

better performance than DCS and mobile-agent-based Full Data Fusion (FDF).

## 6. References

[1] D. Donoho, "Compressed sensing," IEEE Trans. on Information Theory, Vol. 52, No. 4, pp. 1289–1306, April 2006.

[2] Y. Tsaig and D. Donoho, "Extensions of compressed sensing," Signal Processing, Vol. 86, No. 3, pp. 533–548, March 2006.

[3] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Transactions on Information Theory, Vol. 53, No. 12, pp. 4655–4666, December 2007.

[4] Y. C. Kim, S. S. Narayanan, and K. S. Nayak, "Accelerated three-dimensional upper airway MRI using compressed sensing," Magnetic Resonance in Medicine, Vol. 61, pp. 1434–1440, 2009.

[5] M. Mishali and Y. C. Eldar, "Blind multi-band signal reconstruction: compressed sensing for analog signals," IEEE Transactions on Signal Processing, Vol. 57, No. 30, pp. 993–1009, March 2009.

[6] M. F. Duarte, S. Sarvotham, D. Baron, and M. B. Wakin, "Distributed compressed sensing of jointly sparse signals," in 39th Asilomar Conference on Signals, Systems and Computers, pp. 1537–1541, 2005.

[7] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," IEEE Trans. on Information Theory, Vol. 49, pp. 626–643, 2003.

[8] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," IEEE Signal Processing Magazine, Vol. 21, pp. 80–94, September 2004.

[9] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," http://www.dsp.ece.rice.edu/cs.

[10] J. Meng, H. Li, and Z. Han, "Sparse event detection in wireless sensor networks using compressive sensing," The 43rd Annual Conference on Information Sciences and Systems (CISS'09), Baltimore, MD, 2009.

[11] A. H. Phan, A. Cichocki, and K. S. Nguyen, "Simple and efficient algorithm for distributed compressed sensing," IEEE Machine Learning for Signal Processing (MLSP' 08), pp. 61–66, October 2008.

[12] Q. Wu, N. S. V Rao, J. Barhen, S. S. Iyengar, V. K. Vaishnavi, H. Qi, and K. Chakrabarty, "On computing mobile agent routes for data fusion in distributed sensor networks, " IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 6, pp. 740–753, June 2004.

[13] W. Bajwa, J. Haupt, and A. Sayeed, "Compressive wireless sensing," IPSN'06, Nashville, Tennessee, USA, pp. 19–21, 2006.

[14] H. Luo, J. Luo, Y. Liu, and S. K. Das, "Adaptive data fusion for energy efficient routing in wireless sensor networks," IEEE Transactions on Computers, Vol. 55, No. 10, pp. 1286–1299, October 2006.

*Wireless Sensor Network*, 2009, 1, 467-474
doi:10.4236/wsn.2009.15056 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# EasiSim: A Scalable Simulator for Wireless Sensor Networks

**Haiming CHEN[1,2], Li CUI[1*], Changcheng HUANG[3], He ZHU[1,2]**
[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[2]*Graduate School of the Chinese Academy of Sciences, Beijing, China*
[3]*Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada*
*Email*: {*chenhaiming, lcui, zhuhe*}*@ict.ac.cn, huang@sce.carleton.ca*
*Received July* 11, 2009; *revised September* 21, 2009; *accepted September* 23, 2009

## Abstract

Traditional simulators have deficiencies of scalability, thus they are not so efficient in running simulations with large-scale networks. In this paper, we present a new simulator, namely EasiSim, specifically for evaluating sensor net-works on a large scale. EasiSim is featured by organizing its core components, including nodes, topology and scenario, in a hierarchically structured approach. The hierarchically structured organization enables nodes to process all the concurrent events in one batch, hence reducing the running time by an order of magnitude. Moreover, we propose a visualization scheme based on a Client/Server model which separates the graphical user interface (GUI) from the simulation engine, and therefore the scalability of the simulator will not be decreased by complex GUI. Extensive simulations show that EasiSim outperforms ns-2 in terms of *real running time* and *memory usage*.

**Keywords:** Wireless Sensor Networks, Simulator, Scalability, Component Organization, Visualization Scheme

## 1. Introduction

With the development of MEMS and wireless communication, sensor networks have become a hot research topic. Simulation is an effective approach for evaluating the performance of networks. A series of simulation tools have been developed to ease design and deployment of network protocols. Most of these existing simulators are established for traditional wired or wireless networks, so they cannot fulfill the requirements of modeling sensor networks precisely and running large-scale experiments efficiently. We will start with a brief overview on existing network simulators below.

### 1.1. Related Works

ns-2 [1] is a discrete event driven general-purpose network simulator originally developed for modeling the transport control protocols and routing algorithms of wide-area Internet. The CMU Monarch project's extension made ns-2 support the wireless and mobile networks. Although ns-2 has evolved substantially over the past few years, the basic architecture remains the same. Its split-programming model and *object-oriented* architecture make it extensible, but not so scalable.

OPNET [2] is another discrete event driven general-purpose network simulator. OPNET is featured by its GUI-based modeler, which provides an intuitive way to customize modules, like different sensor-specific hardware units. However, the GUI-based modeler sacrifices simulation efficiency for convenience of customization. Therefore, it suffers from the same problem of scalability as ns-2.

Other general-purpose simulation platforms, like OMNeT++ [3] and J-Sim [4], are also widely used in simulating the traditional wired and wireless networks. Since these two simulators were designed from the beginning with module reusability in mind, they put less emphasis on the scalability of simulator.

Based on the simulation framework for sensor network (SensorSim) proposed by Park [5], all the above mentioned general-purpose network simulators have been extended to support for sensor networks, but never been modified in terms of their architectures to address the problem of scalability.

The problem of scalability is more severe for bit-level (or instruction-level) emulators, like TOSSIM [6] and ATEMU [7], than for the above mentioned packet-level simulators.

### 1.2. Our Contributions

In this paper, we design and implement a new simulator, namely EasiSim, specifically for sensor networks with

*scalability* as our first goal. Differing from traditional object-oriented and component-based simulators, EasiSim is a discrete event driven *structure-based* simulator. The main contributions of this paper can be summarized into following three aspects.

1) To enables the node to process all the concurrent events in one batch, we propose a three-dimension sorted linked list (3D list) to organize all the *nodes* into a hyper-structure called *topology*. In this way, the number of events generated during simulation will be reduced by an order of magnitude, thus to decrease the running time.

2) To ease extension of the simulator, we design a *scenario* structure to integrate topology with all other components of the simulator, such as discrete event queue and clock. The hierarchical organization of the components, including nodes, topology and scenario, makes EasiSim not only scalable but also extensible.

3) To visualize the progress of a simulation, we propose a *visualization* scheme based on a client-server model, which separate the graphical user interface (GUI) from other simulation modules, and make GUI and simulation engine run in a distributed way. Therefore, our proposed visualization scheme will not affect performance of the simulator in terms of running time.

The rest of the paper is organized as follows. We describe the component organization in Session 2. Details of integrating the components of the simulator are presented in Session 3. In Session 4, we elaborate on the visualization scheme of the simulator. We evaluate the performance of the simulator in Session 5. At last, we make a brief conclusion and point out future work in Sessions 6 and 7 respectively.

## 2. Component Organization

### 2.1. Basic Principle

We design our simulator based on a sequential, discrete event driven framework. As shown in Figure 1, the simulator mainly consists of following five components.

1) Clock is a 64-bit integer to represent the current time of the simulator.

2) Random number generator is a collection of functions to generate the commonly used random numbers, such as uniformly distributed numbers.

3) Entity is the object to be simulated. For a sensor network, the entity is nothing but a collection of nodes.

4) Discrete event queue is a sorted event list ordered by the time when the event is scheduled to be processed.

5) Event dispatcher is a procedure responsible for fetching the latest event to be processed from the head of the discrete event queue and invoking the corresponding event processing procedure.

The time flow and event flow link the components together. It is worth noting that: 1) the clock of the simulator should be updated to the scheduled time of the dispatched event, before invoking the corresponding event processing procedure; 2) in the event processing procedure, states of more than one related nodes may be required to be updated concurrently and some future events are supposed to be scheduled.

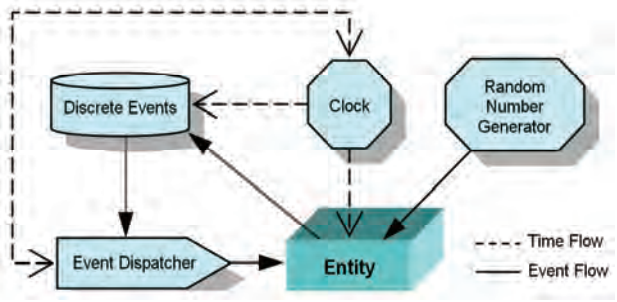



**Figure 1. Component architecture of the sensor network simulator.**

In the existing object-oriented and component-based network simulators, each node is typically modeled as an object. To update states of several nodes simultaneously, it is required to generate some concurrent events and invoke all the corresponding nodes' methods in sequence. This can lead to lots of overhead in terms of running time and memory usage. In wireless networks, packet transmission is such an event that requires updating the states of all the neighbors of the transmitting node, and it creates a large number of events which accounts for a large proportion of all the events in a sensor network. So we propose a new modeling method to establish an efficient structure to support updating several nodes' states in one batch.

We model each node in the sensor network as a structured variable rather than an object. Based on the node structure, we design a three-dimension sorted linked list to organize all the nodes into a hyper-structure called *topology*. We will describe the details of the node and topology structures next.

### 2.2. Node Structure and Topology Structure

Each node is modeled as a variable with the same structure, as shown in Figure 2. The foremost three fields refer to the type, identity and location of the node respectively. The next four fields contain all the necessary information about the settings and states of the protocol in each layer, which are followed by six pointers to link the nodes into a three-dimension sorted list (3D list). Each dimension is a doubly linked list sorted by the identity, the x-coordinate, and the y-coordinate of the node, respectively. This three-dimension sorted linked list is defined as a structure named topology. In the topology structure, it contains three pairs of pointers to indicate the head and tail of each dimension of the sorted linked

```
struct _node{
        NODETYPE nodeType;
        NODEID      nodeID;
        COORDINATE locate;

        PHYDATA phyData;
        MACDATA macData;
        ROUTEDATA routeData;
        APPDATA appData;

        struct _node *nextNodeByID;
        struct _node *preNodeByID;
        struct _node *nextNodeByX;
        struct _node *preNodeByX;
        struct _node *nextNodeByY;
        struct _node *preNodeByY;

        PTOPO pTopo;
};
```

**Figure 2. Definition of the node structure.**

list. The last field in the node structure is just the pointer to the topology, which provides the node with a convenient way to operate on the global information or any other node.

When initializing the simulator, each node is assigned a unique integer number as its identity, and a pair of coordinates as its location. According to the values of these attributes, the node is inserted into the three-dimension sorted linked list in ascending order, while the pointers in the topology structure keep track of the head and tail of each dimension of the list.

## 2.3. Advantages of the Organization

Suppose $N$ nodes are uniformly scattered in a square area of $S$ square meters, and the transmission range of each node is $R$ meters. If the radio propagation is modeled as disk, in the traditional simulators, it is required to compute distances between the transmitting node and all the $N$ nodes in the square area.

With support of the 3D list, when a transmitting event is scheduled during simulation running time, the transmitting node can be found in the list quickly by its identity, and then all its neighbors can be determined rapidly by traversing the list and calculating the distance (or signal attenuation) between the current node (receiving node) and the transmitting node because the list is sorted based on the physical locations of the nodes. Illustrated in Figure 3, the following steps are involved to determine the set of nodes affected by the propagation.

1) Locate the transmitting node in the list by any fast search algorithm.

2) Traversing forward and backward from the transmitting node along either the x-coordinate or the y-coordinate and comparing the x-coordinate and the y-coordinate of the current node in the list and the transmitting node to determine whether the node is located in the intersection region of the light gray area (marked as A) and the dark gray area (marked as area B). Because the list is sorted, only those nodes in the light shaded area are involved to compare their coordinates with the transmitting node. So the number of nodes involved in this step is reduced to $N \cdot \frac{2R \cdot \sqrt{S}}{S} = N \cdot \frac{2R}{\sqrt{S}}$.

3) Compute the distance from the transmitting node to each of the nodes in the intersection region of the light gray area and the dark gray area, to see whether it can hear from the transmitting node. The number of nodes involved in the step is reduced further to $N \cdot \frac{(2R)^2}{S}$.

Let $T_1$ be the benchmark of computation time, which is the time to do addition or subtraction operation. Multiplication operation time and relation operation time are $p$ times and $q$ times as much as the benchmark respectively, where both $p$ and $q$ are larger than one. Since it involves two subtraction operations and two comparison operations to determine whether one node is in the intersection area of A and B, the time to check all the nodes in the light gray area is $N \cdot \frac{2R}{\sqrt{S}} \cdot 2 \cdot (1+q) \cdot T1$. For the same reason, we can know the time to determine the set of nodes that can hear from the transmitting node is $N \cdot \frac{(2R)^2}{S} \cdot [3(1+p)+q] \cdot T1$.

So the computational time to determine the neighbors of the transmitting node is reduced by a percentage of $\left[\frac{4R(1+q)}{\sqrt{S}(3+3p+q)} + \frac{4R^2}{S}\right] \times 100$ with adopting the 3D list. For example, the time can be reduced by about 86.1%, when $p$ is 2 and $q$ is 3.

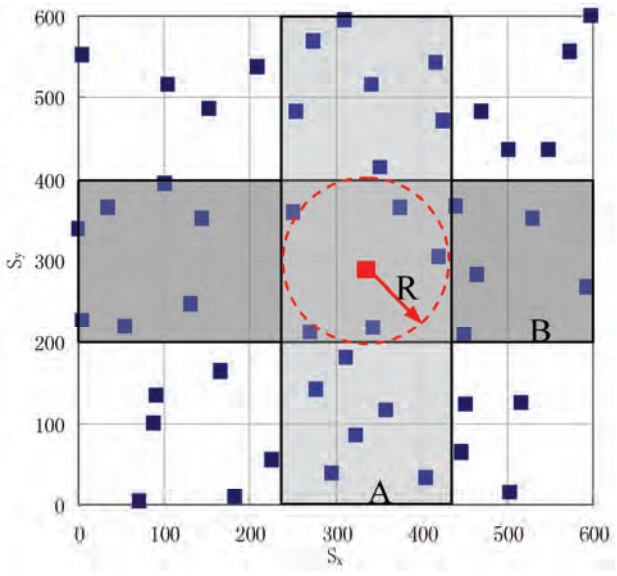



**Figure 3. The set of nodes affected by the propagation.**

## 3. Component Integration

To implement a discrete event driven simulator, we should integrate all the components together, mainly including discrete event queue and clock, besides nodes and topology. As described in the above section, we have integrated the nodes into the topology by organizing all the nodes in the network into a three-dimension sorted linked list. Each dimension is a doubly linked list, whose head and tail are indicated by a pointer respectively. In each node, there is also a pointer to the topology structure, which provides the node with a convenient way to operate on the other nodes.

As shown in Figure 4, the topology and other components, such as event queue and clock, are integrated into an upper-layer structure named scenario. The scenario structure mainly includes three fields, which are *topo*, *cur_time* and *fel*. The *topo* is the topology structure mentioned above, which keeps the pointers to the heads and tails of the three-dimension sorted linked list, and a pointer to the scenario. Through this pointer, nodes can access to all the data in the scenario directly. The *cur_time* is a 64-bit integer to indicate the current time of the simulator. *fel* is the event queue to organize all the events scheduled to be processed in the future. In the discrete event driven simulator, events are dynamically generated and released to drive the running of the simulator, which will involve lots of memory accesses, so the organization of the events should also be carefully designed.

### 3.1. Structure of the Event List

The traditional approach of managing the events is allocating memory once a new event is scheduled, and releasing it once it is processed. Since memory allocating operation is time consuming, we structure the event list as a 2-Dimension linked list to reduce the frequency of allocating memory. Each dimension is a sorted linked list. One is for organizing the future events (named scheduled list), and the other is for collecting the released events (named free list). When a new event is to be generated, the free list is firstly checked to see whether there is a free event that can be "reused". If yes, the event will be renewed and moved to scheduled list, otherwise, a system call is invoked to allocate memory which is inserted into the scheduled list.

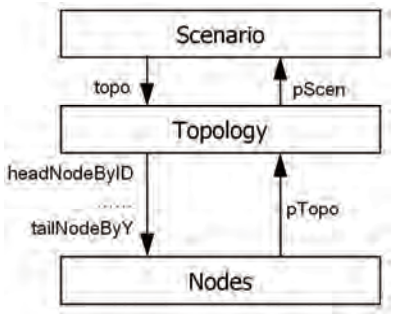



**Figure 4. Architecture to integrate the components of the simulator.**

Supposing the instant number of events scheduled to be processed at time $t$ during running time is $N_t$, the total times to invoke system calls for the events is $\max\{N_t\}$ with the aid of the free list, while it is $\text{sum}\{N_t\}$ without the aid of the free list. The peak volume of memory allocated for the events are both $\max\{N_t\}$*L for the simulator, where L is length of the event.

So the structure of the event list not only reduces the frequency of allocating memory, but also inherits the merit of the traditional approach in memory usage.

## 4. Visualization of the Simulator

Visualization is also an important component of the simulator, though it is not indispensable. The traditional way to implement visualization is off-line replaying, like ns-2/Nam, which displays the flow of packets according to the trace file. Because the replaying depends on the trace file dumped during simulation, the *off-line* approach is time consuming and may influence the performance of the simulator in terms of scalability. Other tools integrate the graphic user interface with the simulation engine, which has bad effects on the scalability of the simulator.

We propose an *on-line* approach to show the progress of simulation by a separate process locally or remotely, based on a Client/Server model. A server process, which can be viewed as the graphic user interface (GUI) of the simulator, is firstly launched in the local machine or a remote machine. If the user requests to display the process of the packet flow or to visualize the states of the nodes, he can let the simulator connect with the server before starting running, by specifying the address and port on which the server is listening. During simulation, the server process is responsible for receiving and parsing the packets encapsulating the requests of visualization, which are generated and sent by the client.

For example, when the simulator finishes initialization, all the created nodes need to be displayed in the GUI. So the simulator will send a packet, like *Node sensor 0 100.0 200.0,* to the server. The first word *Node* in the packet tells the GUI to draw a node, and *sensor* indicates the type of the node. Different types of nodes, such as sensors and sink, may be illustrated by different shapes in the GUI. The number following the type of the node is its identity. When the server receives the packet, it will draw a circle or a rectangle at the coordinate of (100.0, 200.0) to represent the node. Other packets are also defined to visualize other objects in the simulator, such as

radio propagation, link establishment and packet flowing.

In this way, the visualization of the simulator can be implemented without making significant modification to the established simulation engine. Since the GUI server can be run in a remote machine and the simulator can communicate with it in asynchronous way, the visualization will not affect the performance of the simulator.

## 5. Performance Evaluation

The performance of our proposed simulator, named EasiSim, has been evaluated in terms of the following metrics.

1) Real running time: real running time is the direct indicator of scalability. It is obvious that the real running time can be influenced by the *number of nodes* and the simulation time. So we will firstly examine the trend of change on real running time as the number of nodes increases, and then influence of the *configured simulation time* on the real running time will be examined.

2) Memory usage: total memory required to run simulation can also influence the scalability of the simulator, since more memory usage means more frequent operations on the system resource, which is very time consuming.

To compare its performance with ns-2, we first implement a unified suite of protocols, from the physical layer to the application layer, in both simulators. Details about how the protocols are designed and implemented have been presented in another paper [8]. Since the running time of simulation is directly related with the protocols implemented in the simulator, here we give a brief introduction of the protocols implemented by us, especially the MAC protocol and the routing protocol.

### 5.1. Protocol Implementation

#### 5.1.1. Media Access Control (MAC) Protocol

Referring to the media control model integrated in the TinyOS [9], we designed a lightweight media access control protocol, named *TinyMAC*. Since the protocol should be refined further, we firstly established a simple model, which does not take the power saving mechanism into consideration. The finite state automaton of the protocol is shown as Figure 5.

From Figure 5, we can see that there are four states, namely idle (IDLE), initial back-offing (INIT-BO), congestion back-offing (CONG-BO) and transmitting (TRANS), existing in the automaton. The words beside the arcs, which are capitalized with C, stand for the conditions to trigger the change of the states. For example, supposing the current state is idle and a packet needs to be sent, if the current state of the physical layer is idle (that is C1), the node changes its state into initial back-offing; otherwise (that is C5), the node starts congestion back-offing. When the congestion back-off ends (that is C6), it returns to idle and repeats the above described process. During initial back-offing, notification of state change (from idle to busy, that is C2) from the physical layer can stop its progress, but cannot make the current state changed. When the initial back-off ends (that is C3), the node starts transmitting the packet in the buffer. The node does not return to idle until the last bit of the packet is sent (this is C4).

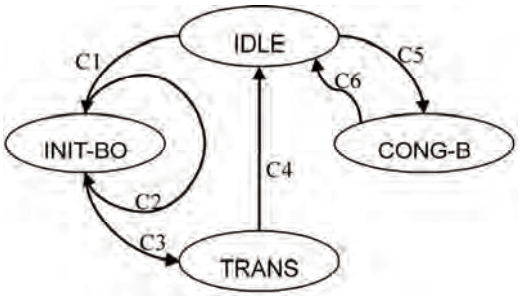



**Figure 5. Finite state automaton of the TinyMAC protocol.**

In order to reduce the complexity of the media access protocol in the sensor network, we temporarily exclude the reliable transmission mechanism from the link layer. Referring to the media control model implemented in TinyOS, we adopt the following equations to determine the length of the initial back-off and congestion back-off.

$$InitBO=\text{randInt}(0,31)+1 \quad (1)$$

$$CongBO=\text{randInt}(0,15)+1 \quad (2)$$

The function of randInt($a$, $b$) is defined to return a random number between $a$ and $b$. So the maximum length of the initial back-off is 32 bytes, which implies about 13.3 milli-seconds when the channel bandwidth is 19.2 Kbps. For the same reason, the maximum length of congestion back-off is about 6.7 milli-seconds.

#### 5.1.2. Routing Protocol

Routing protocols are considered specific for the sensor network, since it is suggested to provide support for data aggregation or fusion to reduce the volume of data and energy consumption [10]. In this paper, we don't put our efforts to design a data-centric or power-efficient routing protocol for the sensor network. Instead, we design a lightweight routing protocol based on flooding, which is referred to as *TinyFlood*, to evaluate performance of simulators.

Flooding is considered to be the simplest scheme to route data in the multi-hop network. But it is also well known that flooding can lead to inner explosion and overlapping, which can waste a lot of bandwidth and other network resources. To reduce the overhead resulting from the uncontrolled flooding, we make following improvements to avoid flooding loop and infinite re-forwarding of outdated packet.

One way to avoid looping packet in flooding is making the node remember the previously flooded packets. When an intermediate node receives a flooded packet, it firstly checks whether it has been flooded by the current node; if not, it will relay the packet by flooding and keep the packet in the memory; otherwise, it will drop the packet. For nodes in the sensor network, memory is severely constrained, so tracing the previously flooded packet seems impractical.

In *TinyFlood*, we make the packet carry the information of the route through which it has been flooded. When an intermediate node receives a flooded packet from the upstream, it checks whether itself has been included in the trace list of the packet; if not, add itself to the trace list and re-flood the packet to the downstream; otherwise, it will drop the received packet. Supposing the upper bound of the number of nodes in the sensor network is 65536, the length of identity of each node will not exceed 2 bytes. So the maximum length of the extra payload added to the packet depends on the TTL (Time To Live) configured by the user, which can be expressed as Equation (3). Through this improvement, the problem of flooding loop is solved without requiring large memory.

$$L_{max\_extra\_payload}=2\times \mathrm{TTL} \quad (3)$$

Considering the limited bandwidth of the sensor network, each packet cannot be so long, or else the network will always in burst state. Supposing the total packet size can not exceed 36 bytes, and the average length of data generated by the sensor node is 24 byes, it can be easily deduced from (3) that the TTL can not be more than 6.

Next, we run simulations to compare performance of our designed simulator with ns-2 in terms of the above mentioned metrics. Simulations were run on a Pentium-IV3.0 GHz processor with 1 Gbytes of RAM memory. The GUI ran in a remote machine.

## 5.2. Real Running Time versus Number of Nodes

The experiments were set up by putting 10 to 1000 nodes uniformly in a 1000 by 1000 meter square field. The transmission range of the node is 250 meters. The node in the left bottom corner is chosen to collect data and send the readings to the sink, which is in the right top corner of the field. The sensor nodes were configured to send the readings every 1 minute, and the simulation time is one hour. The length of the packet is 36 bytes, and the physical rate of the node is 19.2 kbps.

The running time is calculated as the difference between the time when the simulator finished initialization and the time when the simulation ends. All the results are the average of 5 runs with 95% confidence intervals.

As shown in Figure 6, for both EasiSim and ns-2, the running times are below 5 second when less than 100 nodes are put in the field. As more nodes are added to the

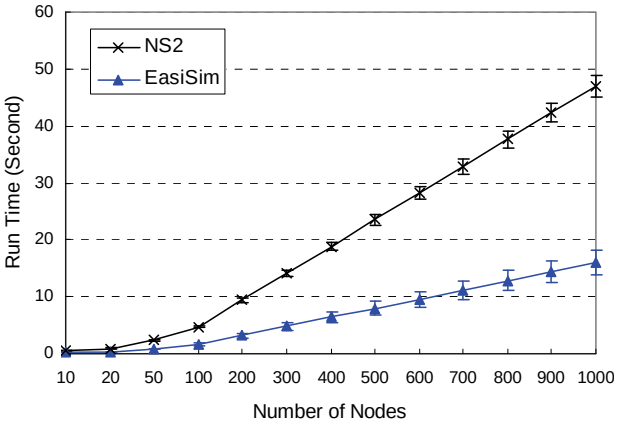



**Figure 6. Real running time versus number of nodes.**

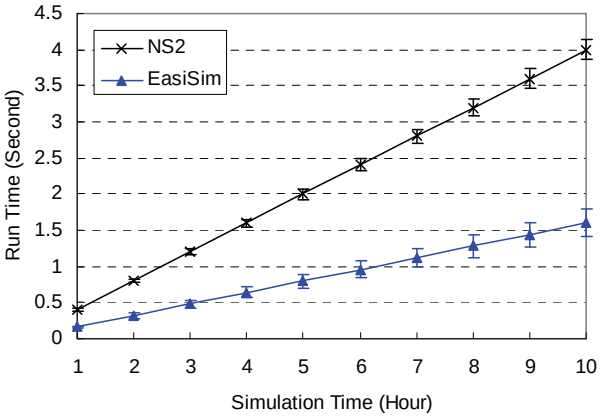



**Figure 7. Real running time versus simulation time.**

scenario, the simulation times increase superlinearly. This can be attributed to the event explosion when the nodes become denser. However, the run time of EasiSim increases much more slowly than that of ns-2. This can be owed mainly to the efficient approach to merge the concurrent events described in Session 2.

## 5.3. Real Running Time versus Simulation Time

In this experiment, we put 10 nodes uniformly in a 1000 by 1000 meter square field, and run the simulation with the same parameters as described in the former experiment for 1 hour to 10 hours.

As illustrated by Figure 7, the running times with both EasiSim and ns-2 rise with the increase of simulation time, because more events are generated to be processed. However, the running time of EasiSim is much less than that of ns-2. Since the profits gained from event merge can be neglected here, we can attribute the advantage of EasiSim to its structure based modeling method. Because all the data representing the state of each node is stored in a structured variable, rather than in an object, they can be accessed directly by the processing procedure without

invoking other methods, the processing time can be reduced a lot.

### 5.4. Memory Usage

The setting up of experiments to evaluate the memory usage of the simulator is the same as we described in the above subsection. Here, we record the volumes of memory space allocated for the nodes and the events in the simulator. Figure 8 shows the results of the experiments. We can see that EasiSim is always more memory efficient than ns-2. The main reason leading to the result can be concluded as follows.

In ns-2, every component of the node is modeled by an object, and the components then comprise the node. Each object in ns-2 has a shadow in memory, so ns-2 needs twice more spaces than EasiSim to store the nodes in the network.

## 6. Conclusions

This paper presented a new simulator called EasiSim, for simulating sensor networks at large scales.

EasiSim is featured by the *structure-based* modeling method and the *hierarchical* organization of the components. As the fundamental components, the *node* structures are firstly organized into a three-dimension sorted linked list. Pointers to the head and the tail of each dimension of the 3D list are then organized into the hyper-structure called *topology*, through which all the nodes involved in the current event can be operated directly. In this way, some concurrent events can be merged and thereby the running time can be reduced by an order of magnitude. The topology structure is then integrated with other components of the simulator, such as the discrete event queue and the simulation clock, into the top-level structure named *scenario*.

We evaluate the scalability of our designed simulator in terms of *real running time* and *memory usage*. The results show that it takes less time and less memory for EasiSim than for ns-2 to complete simulations with the same number of nodes and configured simulation time.

Furthermore, we proposed a visualization scheme based on a client-server mode, which enable the simulation and GUI processes to run in a distributed way. Therefore, our proposed visualization scheme does not decrease the performance of the simulator in terms of scalability.

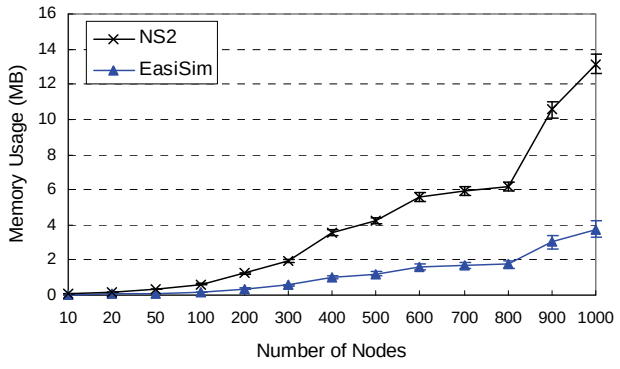



**Figure 8. Memory usage versus number of nodes.**

## 7. Future Work

So far, we have established a scalable simulation platform for sensor networks. To evaluate the performance of the simulator, we also implemented the disk radio propagation module, the MAC protocol and the flooding protocol in the simulator.

As for our future works, we plan to extend the modules, including the radio channel modules, the environment modules and the networking protocol modules, to make the simulator support for modeling the sensor networks more precisely. A practical battery and energy module is also supposed to be implemented in the future days, since it is of vital importance for modeling the power efficiencies of different protocols and life time of the sensor nodes.

As more modules added to EasiSim, the scalability of the simulator will be reevaluated and its performance will be improved step by step. Besides that, the visualization scheme will be refined and its effects on the scalability of the simulator will be investigated more deeply.

## 8. Acknowledgements

This research is supported in part by the Chinese Academy of Sciences (CAS) - Croucher Funding Scheme for Joint Laboratories under Grant No. GJHZ200819, the National Basic Research Program of China (973 Program) under Grant No.2006CB303000, and the CAS Knowledge Innovation Program under Grant No. KGCX2-YW-110-3.

## 6. References

[1] The Network Simulator–ns-2. http://www.isi.edu/nsnam/ns.

[2] F. Desbrandes, S. Bertolotti, and L. Dunand. “OPNET 2.4: An environment for communication network modeling and simulation,” In Proceedings of the European Simulation Symposium, pp. 609–614, Delft, Nertherlands, October 1993.

[3] A. Varga. “The OMNeT++ Discrete event simulation system,” In Proceedings of the European Simulation Multiconference (ESM’01), Prague, Czech Republic, June 2001.

[4] H. Tyan. “Design, realization and evaluation of a component-based compositional software architecture for network simulation,” PhD thesis, Ohio State University, 2002.

[5] S. Park, A. Savvides, and M. B. Srivastava. "SensorSim: A simulation framework for sensor networks," In Proceedings of MSWiM'00, pp. 104–111, Boston, Massachusetts, USA, 2000.

[6] P. Levis, N. Lee, M. Welsh, and D.Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," In Proceedings of SenSys'03, pp. 126–137, November 2003.

[7] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J.S. Baras. "ATEMU: A fine-grained sensor network simulator," In Proceedings of SECON'04, pp. 145–152, October 2004.

[8] H. Chen, C. Huang, and L. Cui. "Lightweight protocol suite for wireless sensor networks: Design and evaluation," in Proceedings of IEEE ISCIT'07, pp. 1155–1160, Sydney, Australia, October 2007.

[9] A. Woo S. Hollar D. Culler J. Hill, R. Szewczyk, and K. Pister. "System architecture directions for networked sensors," In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'00), pp. 93–104, November 2000.

[10] B. Kr ishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in Proceedings of the IEEE INFOCOM, 2002.

*Wireless Sensor Network*, **2009, 1, 475-481**
doi:10.4236/wsn.2009.15057 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Novel Tag Anti-Collision Algorithm with Adaptive Grouping

**Jun DING, Falin LIU**
*Deptartment of Electronic Engineering and Information Science*
*University of Science and Technology of China, Hefei, China*
*Email*: *dingju*11@*mail.ustc.edu.cn, liufl@ustc.edu.cn*
*Received May* 19, 2009; *revised July* 23, 2009; *accepted July* 25, 2009

## Abstract

For RFID tags, a Novel Tag Anti-collision Algorithm with Grouping (TAAG) is proposed. It divides tags into groups and adopts a deterministic method to identify tags within group. TAAG estimates the total number of tags in systems from group identifying result and then adjusts the grouping method accordingly. The performance of the proposed TAAG algorithm is compared with the conventional tag anti-collision algorithms by simulation experiments. According to both the analysis and simulation result, the proposed algorithm shows better performance in terms of throughput, total slots used to identify and total cycles.

**Keywords:** Tag Anti-Collision, RIFD, Tag Estimate, Grouping

## 1. Introduction

With the development of related technologies, RFID systems are playing an important role in many areas such as library, transportation etc. A RFID system is composed of one or more readers and a number of tags. One of the main issues in a RFID system is the fast and reliable identification of all tags within the reader range. Since all the tags share the same transmission medium, multiple-access must be considered for achieving better identification. In RFID system, this multiple-access problem is called tag anti-collision problem.

There are two major methods of tag anti-collision currently. One is ALOHA based protocols and the other is binary tree based protocols. The tree based protocols, although having the advantage of high throughput, suffer from the problem of insecurity, and this disadvantage restricts their application to some extent, while the ALOHA based protocols have the problem of low average throughput.

The basic ALOHA method, such as the BFSA (Basic Framed Slotted ALOHA), for its simple implementation, is used frequently in practical applications [1]. But this method can only work well when the total number of tags is relatively small. As the number of the tags increases, the probability of tag collision becomes higher and the time used to identify the tags increases rapidly. Refined ALOHA based algorithm, such as Dynamic Framed slotted ALOHA (DFSA), can achieve a better performance to some extent. However, it also has its disadvantages, which restrains its application. To achieve better identification, we propose a new algorithm, which can achieve a better anti-collision performance with a relatively simple implementation.

The following parts of this article are organized as follows: Section 2 introduces several anti-collision algorithms and points out their defects and limitations. Section 3 proposes a novel anti-collision algorithm named Novel Tag Anti-collision Algorithm with Adaptive Grouping (TAAG) and presents the analysis of this algorithm. Section 4 gives the results of simulation. And Section 5 presents the conclusions.

## 2. Current Algorithms

### 2.1. Basic Framed Slotted ALOHA (BFSA)

BFSA algorithm uses a fixed frame size and dose not change the size during the process of identification. In BFSA, the reader offers the information about the frame size and the random number which is used to select a slot within a frame. Each tag selects a random number and will send its ID during the slot decided by the random number [2].

As most RFID systems use passive tags, frame size is limited in BFSA algorithm [3], therefore there are several defects inherent in this algorithm. Reference [4] proves that when the frame size $L$ equals total number of

tags *n*, namely *L*=*n*, this anti-collision algorithm can yield the greatest throughput. Nevertheless, more often than not, the total number of tags in the interrogation zone of a reader is unknown. So the frame size cannot be specified definitely. On the one hand, if the size is too large, then the number of empty slots increases obviously and therefore the system throughput drops. On the other hand, if the frame size is too small, then the number of collision slots increases rapidly and thus system throughput also decreases.

### 2.2. Dynamic Framed slotted ALOHA (DFSA)

DFSA depends on changing the frame length to achieve more efficient identification [5]. To specify the frame length, DFSA has to use some information including number of successful identifying slots, empty slots and collision slots to estimate the total number of tags in one RFID system. By letting the frame length equal the total number of tags, the RFID system can achieve the best identifying efficiency [4,6].

Firstly, DFSA adopts the initial frame length to identify tags in system. Secondly, DFSA utilizes the identifying result of the previous frame to estimate the total number of tags in system by various tag estimation methods [6−9]. Finally, DFSA specifies frame length according to total number of tags. As a result, the frame length varies with the change of number of tags. DFSA has various versions depending on different tag estimation methods used. This algorithm ameliorates the defects mentioned above in BFSA. However, DFSA also has its disadvantages. In practical applications, the frame length set by the reader is always a specific value because of limitation of implementation. It is difficult to set the frame length to just equal the total number of tags because the total number itself is generally not known beforehand. Common practice is to set the frame length to be some specific values, such as 1, 8, 32, 64, 128 and 256. Thus the performance generally can not reach the theoretical level.

## 3. Novel Tag Anti-Collision Algorithm with Adaptive Grouping (TAAG)

For many application situations of RFID systems, the total number of tags in system is unknown beforehand and thus specifying the frame length becomes a problem. However, the frame length is a key factor to the anti-collision performance of these ALOHA-based algorithms. BFSA disregards this problem while DFSA does not solve this problem either. This problem limits the performance of this type of anti-collision algorithm. Binary-tree based algorithms do not have this problem and thus the system can achieve a greater throughput. However, they also have their shortages. The average identifying time for one tag is rather long, it takes $\log_2 n+1$ times read processes to identify one tag at average and thus the information leakage is much more serious and this in turn threats the security of RFID system. Considering the two aspects mentioned above, we propose a new tag anti-collision algorithm named TAAG.

### 3.1. Description of TAAG

Before presenting the details of TAAG first, we have to illustrate the operational status of tags used in the algorithm by Figure 2.

READY: When the tags are in the interrogation zone of the reader, if the tags have received sufficient energy to support its work, then this status is called “READY”.

STANDBY: The reader initializes all the tags at “READY” in its interrogation-zone and select one group of tags (whose highest *M* bits of register are all zeros) to process “Subrountine1”, which will be described in the later part of this section. We name the status of tags selected as “STANDBY”

QUIET: The status of tags, which have been read by the reader and will not take part in the present and the following arbitrage process, is called as “QUIET”.

This novel Tag anti-collision Algorithm uses an 8-bit register, (registers with other sizes, such as, 16 bits, 4 bits can also be chosen depending on the situation for which this system is designed), and a random number generator (RG) to generate 0/1 randomly. The register (REG) stores bits of 0/1 generated by RG sequentially from *R7* to *R0* until the register is full. These 8 bits, called ID for collision algorithm (IDC), not the intrinsic ID of tags, are used for our anti-collision algorithm. The RG generates *M* bits 0/1 random numbers for the highest

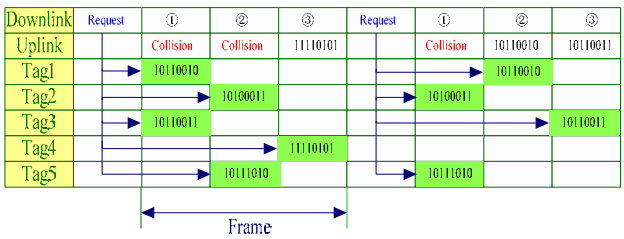



**Figure 1. Anti-collision process of BFSA.**

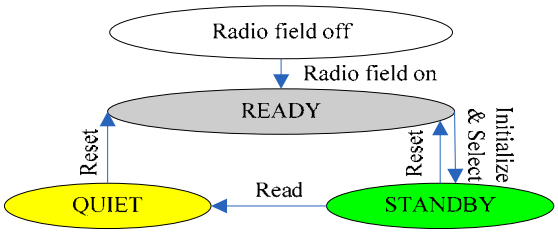



**Figure 2. Status transfer.**

*M* bits of REG and also generates *A* bits 0/1 random numbers for the lowest *A* bits of REG. For ease of simple implementation, both *M* and *A* are limited to be no more than 4 bits (for 8 bits register) in this paper. "*M*" is used for grouping while "*A*" for avoiding collision within group. The other bits (if there are still other bits except for the *M* bits and the *A* bits of REG) are set to be zeros. The structure is illustrated in Figure 3.

Now some commands definition need to be given before precisely describing the Subroutine1 as follows:

Subtract command: Receiving this command, the tag identified turns into "QUIET". For those tags still at "STANDBY", if the highest M bits are all zeros but not the whole IDC, the lowest 4 bits of register subtracts 1.

Plus command: After receiving this command, if the IDC of tag are all zeros, its register pluses 1 or 0 randomly. For those tags still at "STANDBY", if the highest M bits are all zeros but not the whole IDC, the lowest 4 bits of register pluses 1. However, if the lowest 4 bits of IDC are already "1111", the register will keep unchanged.

Note: The above mentioned IDC is used only to avoid collision, and even different tags can share the same IDC. It is the intrinsic ID that will be sent to the reader for identifying each tag.

Subroutine 1-Group Tag Identify: If the IDC of tag are all zeros, the tag sends its intrinsic ID to the reader.

Within any one slot:

- If Only one tag sends its intrinsic ID to reader (Successful transmission slot, abbreviated as S slot)

——Reader sends confirming message by Subtract command

——Count subtracts 1

- Else if More than one tags send their intrinsic IDs (Collision slot, abbreviated as C slot)

——Reader sends confirming message by Plus command

——Count pluses 1

- Else No tag sends its intrinsic ID (Empty slot, abbreviated as E slot)

——Reader sends confirming message by Subtract command

——Count subtracts 1

Figure 4 presents the flow chart of Subrountine1. Note that $a_1$ represents number of S slots, $a_m$ represents number of C slots, $a_0$ represents number of E slots.

Before giving the steps of TAAG algorithm, several important parameters: M, A, Count, and R need to be clarified or explained.

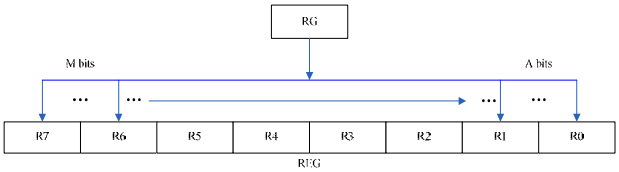



**Figure 3. Anti-collision structure of tag.**

M: The value of M decides that the highest M bits of REG of tag are used for grouping. The original value of M can be set to 4 (other values from 1-3 can also be used).

A: The value of A decides that the lowest A bits of REG of tags are used for avoiding collision within group. The original value of A can be set to 4 (other values from 1-3 can also be used).

Count: The value of Count decides when the Subroutine1 ends (if Count=0, then Subroutine1 ends), the originnal value of Count is $2^A$, namely Count=$2^A$.

R: It represents the number of groups which remains unidentified in system, the original value of R is $2^M$, namely R=$2^M$.

After those parameters were introduced, steps of TAAG can be given as follows. And Figure 5 presents the flow chart of TAAG algorithm.

Step 1: The reader initializes all tags at READY status with the parameters M, A.

From the structure of tags illustrated previously in Figure 3, we can easily know that the RG can generate M bits 0/1 random numbers and send them to the highest M bits of REG of tags at READY, those M bits will be used as the sequence number of groups, for example, if M=2, then 00 will be the sequence number of first group, 01 will be the second, etc. Meanwhile, the RG also generates A bits 0/1 random numbers and sends them to the lowest A bits of REG of tags at READY, these A bits will be used for avoiding collision within group.

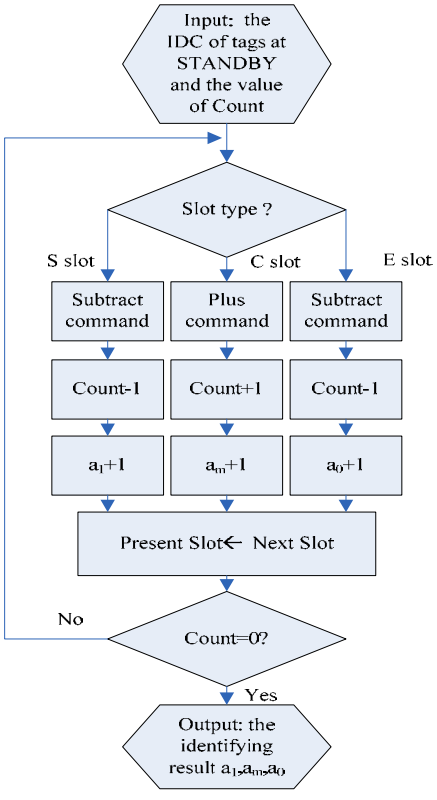



**Figure 4. Flow chart of subroutine1-group tag identify.**

Step 2: The reader selects one group of tags.

If the highest M bits of REG of a tag are all zeros, which means that the tag belongs to group 0 (sequence number of group 0 is 00…0(all the M bits are 0), the reader selects the group 0, the other tags will not be selected because their highest M bits of REG, which represents their sequence number of group, are not all zeros. Only the tags selected turn to STANDBY status. By doing so, there will be one and only one group of tags at STANDBY status.

Step 3: The reader utilizes Subroutine1 to identify the tags at STANDBY status, in other words, to identify the selected group of tags.

Input: the IDC of tags at STANDBY status and the value of Count, Output: the identifying result of the group, $a_1$, $a_0$, $a_m$.

Step 4: If original values of M, A have not been adjusted.

---Depending on the identifying result of group $a_1$, $a_0$ ,$a_m$, adjusts the value of M, A to fit for various application situations. The original values of M, A, more often than not, are not suitable for specific application situations. (The method of adjusting the values of M, A will be demonstrated at the end of this section.)

--- Go back to Step1 (Initializes the tags in READY with the adjusted, more suitable, parameters.)

Else if original values of M, A have been adjusted

---Go to Step 5

Step 5: If R>0

--- The sequence number of groups (the highest M bits of REG of tags) subtracts 1. In this process, all the sequence number of groups will subtract 1, group 1, whose sequence number is 00…01(M bits), turns into group 0 (00…0)(M bits), the group 2 turns into group 1,…,group R turns into group R-1.

--- Go back to Step 2

Else

--- END TAAG

The method of adjusting *M*, *A* mentioned above in Step 4 needs to be clarified at the end of this section as follows:

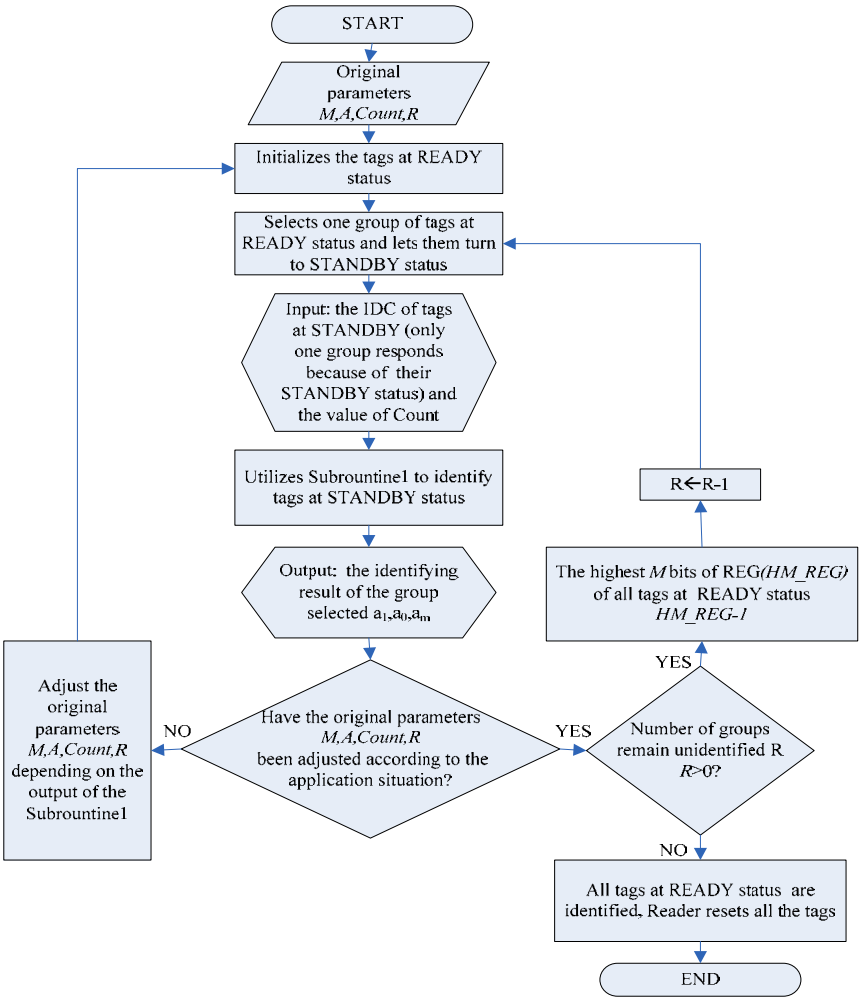



**Figure 5. Flow chart of TAAG.**

Define each combination of the lowest A bits of REG of tags as one Bin, then the number of Bins is $2^A$. If this number equals the number of tags $n$ within a group, i.e. $n=2^A$, the system can achieve the best performance. Based on this, we can specify the value of $A$.

The proof is basically the same as that of Slotted ALOHA [4–6,9]. When the possibility of S slot is the largest, the system can achieve best performance because both Collision slots and Empty slots degrade the performance of system.

Let $L=2^A$, then the probability of occurrence that a Bin is occupied by only one tag (S slot) can be written as:

$$p_1 = C_n^1(\frac{1}{L})^1(1-\frac{1}{L})^{n-1} \quad (1)$$

To get the maximum value of $p_1$, differentiate $p_1$ with $L$,

$$\frac{dp_1}{dL} = \frac{n(n-L)(L-1)^{n-2}}{L^{n+1}} = 0 \quad (2)$$

We can find $n=L$ easily. According to the above principle, we can adjust *M, A, Count*$= 2^A$ *and* $R=2^M$ appropriately as illustrated in Table. 1.

If the total number of tags can not be found in Tab.1, we can substitute it with the closest value. After the values of M, A were adjusted, the values of *Count*, *R* are changed accordingly.

### 3.2. Analysis of TAAG

This algorithm groups tags at the very beginning, and this process possesses lots of advantages. Firstly, grouping tags can decrease the collision because of smaller number of tags within one group, especially when the total number of tags in system is large. Secondly, identifying one group of tags can bring us useful information about the estimation of total tags number, and based on the information, we can adjust the parameters of our anti-collision algorithm appropriately so that it can fit for various applications

Furthermore, the algorithm utilizes Subroutine1-Group Tag Identify to identify the tags within group. There are also many advantages:

**Table 1. Adjust the values of *M, A*.**

| M \ A | 4 | 3 | 2 | 1 |
|---|---|---|---|---|
| 4 | n>=256 | | | |
| 3 | n=128 | | | |
| 2 | n=64 | | | |
| 1 | n=32 | n=16 | n=8 | n<=4 |

1) The implementation is simple, which is beneficial for the extensive application of this tag anti-collision algorithm.

2) Subroutine1 can identify all the tags in system and no tag will be unidentified. This solves the problem existing in the ALOHA-based algorithm where a tag may be regarded as out of the reader's interrogation zone if it can not be identified for a long period. In Subroutine1, if a slot is S slot, then *Count*-1, while if the slot is C slot, the *Count*+1. If there are $x$ C slots, then *Count*+$x$. As a result, Subroutine1 can guarantee that all tags can be identified. Example to illustrate such a process is shown in Table 2.

In the example, $A$=2, *Count*=$2^A$=4, the lowest 4 bits are used for avoiding collision within group and there will be 16 Bins. The 0000 represents Bin1, similarly, we can get other Bins codes like Bin1. T1 represents tag No.1, T2 represents tag No.2, etc. According to Subrountine1, the tags T1, T2, T3 in Bin1 should first be identified, but now they are in the collision Slot. So the IDCs of T1, T2, T3 should plus 0/1 randomly. Because the result has many possibilities, here in Tab. 2 only one possibility is illustrated. The multiple possibilities will also exist when identifying T1 and T2 in Bin1 at Slot 2. Here we choose the possibility that uses the least number of slots, just to show how the algorithm works.

3) Adopting Subroutine1, we can get the estimation of tags remaining unidentified depending on the $a_1$, $a_0$, $a_m$ acquired from the first group's identification. Since all tags can be identified in Subroutine1, the number of S slots $a_1$ must equal the number of tags within group.

## 4. Results

Before presenting the results of TAAG, three main issues involved in tag anti-collision algorithm performance should be kept in mind:

**Table 2. Subroutine1 identifying process.**

| Slot | Slot1 | Slot2 | Slot3 | Slot4 | Slot5 | Slot6 | Slot7 | Slot8 |
|---|---|---|---|---|---|---|---|---|
| Count=4(c) | Count+1=5 | Count+1=6 | Count-1=5 | Count-1=4 | Count-1=3 | Count-1=2 | Count-1=1 | Count-1=0 |
| Uplink | Collision | Collision | T1 | T2 | T3 | T5 | Empty | T4 |
| Bin1(0000) | T1, T2, T3 | T1, T2 | T1 | T2 | T3 | T5 | | T4 |
| Bin2(0001) | T5 | T3 | T2 | T3 | T5 | | T4 | |
| Bin3(0010) | | T5 | T3 | T5 | | T4 | | |
| Bin4(0011) | T4 | | T5 | | T4 | | | |
| Bin5(0100) | | T4 | | T4 | | | | |
| Bin6(0101) | | | T4 | | | | | |

Throughput: the ratio of S slots to the sum of S, E, C slots. The Throughput expresses the ratio of usage of channel.

$$Throughput = \frac{S}{S + E + C} \quad (3)$$

Number of slots (NOS): Total number of slots used to read all the tags. Smaller NOS means that reader can identify all the tags in fewer numbers of slots. In other words, the speed of read process will be fast.

Required Cycles: More required cycles mean that the system spends more time dealing with the communication handshaking between the reader and tags, which is detrimental for the identifying speed of tag anti-collision algorithm.

Now we will show the performance of TAAG. In the following simulation, the original parameter "$M$=4", "$A$=4", $L$ represents the frame size and the simulations are based on Monte Carlo technique.

1) From Figure 6, we may find that if the total number of tags in system is small, BFSA ($L$=128) have larger throughput than BFSA ($L$=256). However, with the increase of the number of tags in system, BFSA ($L$=256) performs better. Figure 6 vividly depicts that DFSA has a larger throughput than BFSA. However, since the limitation of implementation of DFSA in practical application (mentioned in Subection 2.2), the throughput of DFSA is close to that of the BFSA ($L$=256) with the increase of tags in system. The algorithm we proposed, TAAG, has the best performance in throughput. The throughput of this algorithm reaches to the level of about 0.4 when the total number of tags varies in a large range (form 30 to 1,000).

2) From Figure 7, we can observe that BFSA ($L$=128) needs more slots to identify the same number of tags (If the number of tags is relatively large). When the number of tags in system is small, DFSA needs fewer slots than BFSA ($L$=256). However, with the increase of number of tags in system, the performance of DFSA is close to BFSA ($L$=256), as was stated in Subsection 2.2. The total number of slots, which TAAG uses to identify 1,000 tags in system, is about 2800 while BFSA ($L$=256) (or DFSA) has to use about 5500 slots. It is obvious that TAAG has advantage in terms of the number of slots used to identify.

3) From Figure 8, we conclude that TAAG only needs communication handshakes with tags for 2 times, which improves the identifying speed of RFID system for certain.

## 5. Conclusions

We proposed and analyzed Novel Tag Anti-collision Algorithm with Grouping (TAAG). According to the analysis and simulation, we can obtain the following conclusions. Firstly, TAAG anti-collision algorithm has

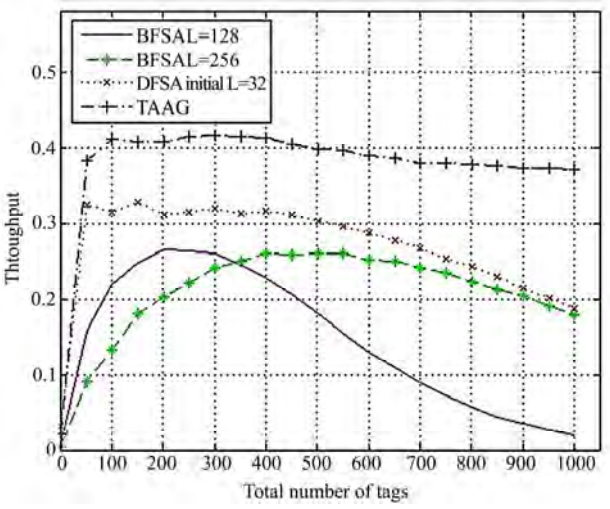



**Figure 6. Throughput vs. number of tags.**

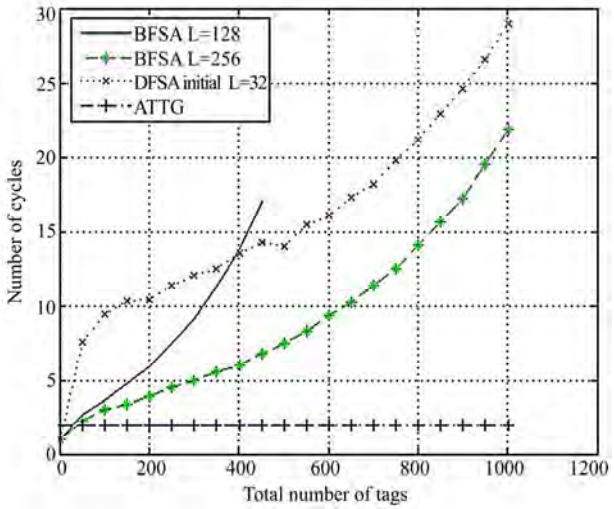



**Figure 7. Number of cycles vs. number of tags.**

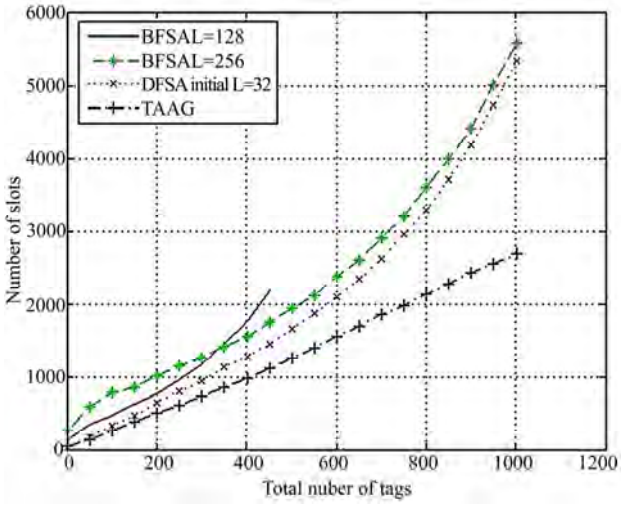



**Figure 8. Number of slots vs. number of tags.**

obvious superiority in terms of all these three parameters of performance, the throughput, the number of slots and the required cycles. Secondly, the implementation of the algorithm is rather simple. The complexity of realization is at the same level as BFSA algorithm. Although the implementation of Reader is relatively complicated, the number of reader in a RFID system is far smaller than the number of tags, so the cost for the whole RFID system will not be changed greatly. Thirdly, compared with Binary-tree based algorithm, TAAG also has its superiority in terms of lower cost and less average time needed to identify a single tag. Finally, the applicability of TAAG is another merit. TAAG can be used for different number of tags in RFID system. And the system can achieve a good performance. Furthermore, because number of slots used to identify is rather small, the identifying speed of RFID system is high. So TAAG can be used in fast identification situations. In summary, TAAG, as a tag anti-collision algorithm, owns its advantages in some applications and deserves to be applied in various RFID systems.

## 6. References

[1] K. Finkenzeller, "RFID handbook," Second Edition, JOHN WILEY & SONS, pp. 195–219, 2003.

[2] PHILIPS Semiconductor, "I-CODEI system design guide: Technical report," May 2002.

[3] S.-R. Lee, S,-D. Joo, and C,-W, Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Service, pp. 166–172, 17–21 July 2005.

[4] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," IEEE Transactions on Automation Science And Engineering, Vol. 6, No. 1, pp. 9–15, January 2009.

[5] F. C. Schoute, "Dynamic frame length ALOHA," IEEE Transactions on Communications, Vol. 31, No. 4, pp. 565–568, April, 1983.

[6] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithm for fast object identification in RFID system," ICPADS' 05, Vol. 2, pp. 63–67, 22–22 July 2005.

[7] H. Vogt, "Efficient object identification with passive rfid tags," Lecture Notes in Computer Science, Pervasive Computing, Vol. 2414, pp. 8–113, January 2002.

[8] D. K. Klair, K.-W. Chin, and R. Raad, "On the accuracy of RFID tag estimation functions," International Symposium on Communication and Information Technologies, ISICT'07, pp. 1401–1406, 17–19 October 2007.

[9] H. Vogt, "Multiple object identification with passive RFID tags," IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 6–9 October 2002.

*Wireless Sensor Network*, 2009, 1, 482-488
doi:10.4236/wsn.2009.15058 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Dynamic Load Balancing with Overlay-Based Reconfiguration for Wireless Sensor Networks

**Hang QIN[1], Li ZHU[2], Zhongbo WU[3]**
[1]*Computer School, Yangtze University, Jingzhou, China; State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China*
[2]*Computer School, Wuhan University, Wuhan, China*
[3]*Department of Computer Science, Xiangfan University, Xiangfan, China; Information School, Renmin University of China, Beijing, China.*
*Email*: {*hangqin*100, *yeah*_1397118 }*@hotmail.com*, *rucwzb*@163.*com*
*Received August* 7, 2009; *revised September* 20, 2009; *accepted September* 25, 2009

## Abstract

Wireless sensor networks are characterized by multihop wireless links and resource constrained nodes. In terms of data collection and forwarding scheduling, this paper investigates the load balancing in sensor nodes and wireless link based on the performance of wireless sensor networks. Leveraging the property of dissimilarity distribution, a method to quantitatively evaluate the benefits of load balancing is presented, in order to access the profitability. Then a novel Dynamic Load Balancing of Overlay-based WSN (DLBO) algorithm has been put forward. In particular, the tradeoff between transferring ratio and the load imbalance among nodes is discussed. The load balancing method in this paper outperforms others based on balancing factor, different nodes number and data scales of applications. The proposed model and analytical results can be effectively applied for reliability analysis for other wireless applications (e.g., persistent data delivery is involved).

**Keywords:** Wireless Sensor Networks; Workload; Dynamic Load Balancing; Dissimilarity Measure; Reconfiguration

## 1. Introduction

Wireless Sensor Networks (WSN) is an accumulation of sensors interconnected by wireless communication channels. Under the control of the network, every sensor node is a small device that can collect data from surrounding area, communicate with each other, and carry out computation. Long distance communication is normally achieved in a multi-hop manner. Thanks to recent advances in remote monitoring system [1–3], such networks are progressing rapidly, and are expected to be popular in applications such as environment monitoring, intrusion detection, and earthquake warning.

Whereas sensor networks scale up in size, effectively managing the distribution of the networking workload will be of great concern [4–5]. Actually, one of the most urgent challenges in designing protocols for a WSN is to make them more energy-efficient by maximizing the network performance without any loss of sensing capability. By extending the workload across a sensor network, load balancing reduces hot spots in sensors nodes and increases the direct communication of the sensor network.

Nonetheless, existing strategies miss one vital measure in the optimization space for routing correlated data, namely, the data communication and nodes dispatching cost. Common intuition tells us that the hot spot problem can be solved by varying the static data apportionment among sensor nodes at different distances to the sink，so that data content can be more evenly dispensed [6–8]. Moreover, as far as the static and mobile sensors in a hybrid WSN is concerned, this is only true to some extent. In terms of the overlay-based WSN, the dynamic load balancing scheme has been shown evenly distribute packet traffic generated by sensor nodes across the different branches of the routing. Therefore, analysis and improvement of sensor nodes dispatching schedule and dissimilarity is the critical metric in collaborative information processing.

## 2. Related Work

The dynamic load balancing routing problem focused on

here is, as indicated in [9], directly bound up with dynamically reallocate incoming external loads at each node [10]. Fatta and Berthold [9] propose distributed process algorithms that may use initiated receiver and still achieve good load balancing, measured by the load balancing index, simultaneously for a dynamic partitioning of the search space. They also show that it naturally tolerates node failures and communication latency and supports dynamic resource aggregation. Furthermore, in [11], the tradeoffs between load balancing and fail-over implementation strategies, and present quantitative performance measurements collected on a commercial multi-homing load balancing system, are addressed in details.

Note that the majority of load balancing strategy developed heretofore is based on such time delays [12], and it proceeds with the assumption that delays are deterministic. In reality, delays are random in such communication media, especially in the case of WLANs. Furthermore, load balancing is attributable to uncertainties associated with the amount of traffic, congestion, and other unpredictable factors within the network. However, the scale of these load balancing methods is partially restricted. For example, the general dispatching mobile sensors problem is overlooked and these nodes are presented as a cluster head usually within a cluster. Thus the load balancing is somewhat limited.

By spreading the workload across a sensor network, H. Dai and R. Han [13] developed a node-centric algorithm that constructs a load-balanced tree in sensor networks of asymmetric architecture, and select the heaviest nodes with the maximum growth space for growth. Y. Wang *et al.* [14] have revealed a notion of a hybrid sensor network consisting of static and mobile sensors. They propose an efficient clustering scheme to group event locations so that the maximum-matching approach can still be applied.

## 3. System Architecture

### 3.1. Overlay-Based Reconfiguration Policy

The sensor nodes are organized through three layers in *Dynamic Load Balancing of Overlay-based* WSN (DLBO). As shown in Figure 1, *Ordinary Sensors Overlay* belongs to the first layer, which keeps the records of some sensor nodes tracking the current data dispatching information. It can be viewed as a large list structure, initialized by the sensor nodes list fetched from *Reconfiguration Tracker* and updated by received data accumulation messages. *Grid Heads Overlay* remains with the second layer which forms a ring-assisted control overlay. The repetitive connection is between any two neighboring nodes, where data collection messages are distributed. *User Overlay* pertains to the third, which constructs the data overlay of DLBO. The data fusion is only interchanged between associates.

For containing the dynamics of overlay topology and enhancing the end-to-end performance of load balancing, a selective approach is presented to raise better dissemination candidates to the upper layer. The sensor node first selects some nearest sensor nodes from *Ordinary Sensors Overlay* as its neighbors according to their distance calculated by the difference between their positioning. According to the feature, some neighbors are selected from the neighbor list into its associate list, in terms of the relevance that can be calculated by the time latency of their dynamic clustering. When a grid head has not been exchanging any data packets in a predefined time interval, it can be removed from the *Grid Head Overlay* and contribute to a procedure to find new triggered grid head; similarly, when a node record of *Ordinary Sensors Overlay* has not been updated for a predefined time interval, it can be dropped from *Ordinary Sensors Overlay*.

### 3.2. The DLBO Framework

Actually, to deploy mobile sensors more efficiently, it should not only cut down the total moving energy but also *balance* the loads of mobile sensors. Therefore, a distributed solution is outlined. The focus here is the load balancing gathering of the data content from the sensor nodes to the sink node. And Figure 2 illustrates a sensor network where source sensors are disseminated on grid and sensed information of the sensors is to be routed to the *grid head*. Arrow lines construct the fusion relation in which sensor nodes respectively aggregate data of different fields, initially. Note that the fusion tree may be changed later due to failures or load balancing, but the resulted relation must also satisfy the above conditions.

Specifically, each mobile sensor acquaints its location and remaining energy to its pertinent grid head. While detecting events, static sensors notify to their grid heads. For acquiring such information, a grid head performs dynamic load balancing algorithm to dispatch mobile

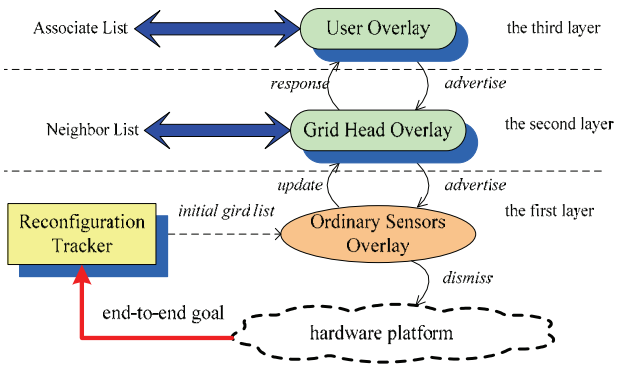



**Figure 1. Overlay management of dynamic load balancing with reconfiguration tracker.**

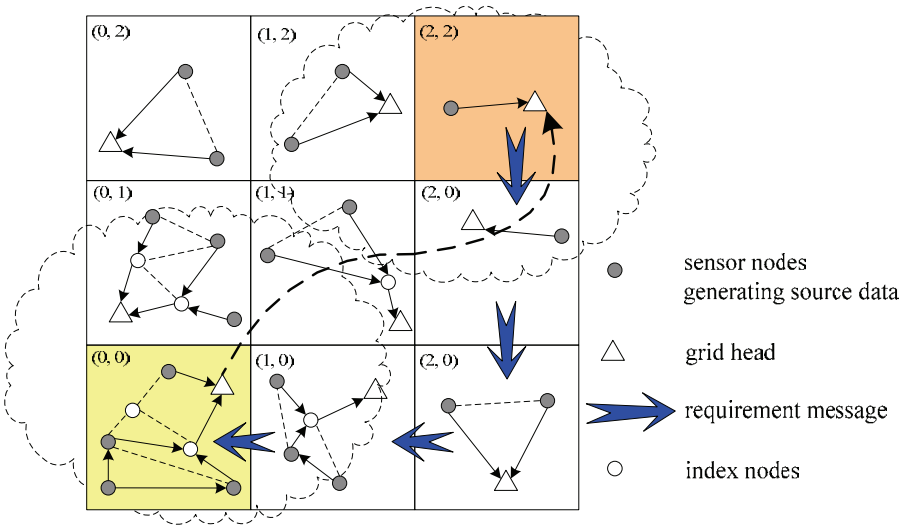



**Figure 2 .The Infrastructure to show how reconfiguration works.**

sensors to the events occurred in its grid. However, if there is no mobile sensor in the grid, the grid head will search available mobile sensors in other grids. Consequently, employing the distributed framework, the area covered by WSN is divided into small *virtual grid quorum*. Based on the adjoining units, the virtual grid quorum is depicted such that all nodes in one grid quorum can communicate with all nodes in the other grid quorum. In each virtual quorum, nodes operate to keep awake and work as grid head, whereas others only need to wakeup periodically. The grid head makes for forwarding messages that pass through the grid quorum.

To decrease the number of message transferring when a grid head seeks for mobile sensors in other grid quorum, each grid head sends issue messages containing the number of mobile sensors in its grid quorum to the same column of units. Furthermore, each grid head has the information of mobile sensors in other grid quorum placed in the same column. When a grid head wants to search mobile sensors in other grid quorum, it sends a *demand* message to the grid head in the same row. Because of the grid structure, it must be a grid head receiving both issue and demand messages.

This framework avoids both unnecessarily removing the sensing data and flooding control messages throughout WSN. Surely, this scheme introduces additional overhead for maintaining index nodes. However, as displayed by the analysis and evaluation results, it can still enhance the end-to-end performance.

## 4. Dynamic Load Balancing Algorithm Design

In this paper, a hybrid WSN consisting of static and mobile sensors is considered. Static sensors form a connected network and fully cover the area of interest to continuously monitor the environment. Table 1 lists all

**Table 1. Notation.**

| Symbol | Definition |
|---|---|
| $G$ | $=(V, E)$, the sensor network |
| $V$ | the set of sensors (nodes) |
| $E$ | the set of edges representing the communication links between pairs of sensors |
| $T$, | the Scheduling Task |
| $D$ | the Sensing Data |
| $s_m$ | $s_m \in T$, message |
| $w$ | the workload to be moved |
| $LB_t$ | the Load Balancing tag |
| $n$ | the iteration number |
| $j$ | the interrupting time, which is a integer |
| $\mathbb{V}$ | the number of sensors |
| $\mathbb{V}'$ | the physical connection number of the given sensor |
| $m$ | the round number |
| $\mathbb{E}_m$ , $\mathbb{E}_{max}$ | every sensor's independent load function and maximal load |
| $\mathbb{E}_m(k)$ | the workload of sensor $m$ in $k$th stationary load time interval |
| $B$ | networks bandwidth |
| $\varepsilon$ | workload for computing new distribution |
| $\gamma_m(n)$ | the iteration distribution |
| $\chi_m(n)$ | the no-executing iteration number for sensor $m$ after the time of the synchronization $n$ |
| $t_n$ | the time cost in the synchronization $n$ |
| $\tau$ | the iteration's time cost can be depicted in agreement loop |
| $f$ | the first sensor which has finished the task allocation |
| $\xi$ | the synchronization workload |
| $\eta$ | the demanding synchronization number |
| $\mathbb{E}(n)$ | the data transmission workload |
| $\psi(n)$ | the workload in the synchronization $n$'s communication |
| $\varsigma(n)$ | the number of transmission task and array's message amount |
| $TW$ | the total workload |
| $\Omega_g(n)$ | balancing factor based on the synchronization time of the different grids |
| $v(j)$ | the number of the grids in the waited queue of the centralized load balancing manager |

notation in this paper, and the problem of dispatching mobile sensors is modeled in Table 1.

The sensor network is modeled as a graph $G=(V, E)$, where $V$ denotes the set of sensors (nodes) and $E$ the set of edges representing the communication links between pairs of sensors. The dynamic load balancing architecture is executed on $G$ to construct a load-balanced tree. The load associated with a given sensor node represents the amount of data periodically generated by that sensor node. Actually, the above framework absorbs the nodes generating the greatest load to the lightest branches to achieve balance.

The study identifies the Dynamic policy in DLBO, it is composed of a *Scheduling Server's Algorithm* and a *Task Counter's Algorithm*. Firstly, *Scheduling Server's Algorithm* derives a task and divides sensing data to every task equally. The algorithms are detailed below for the sake of completeness:

**Epoch 1:** *Scheduling Server's Algorithm*

**Initialization**:
1. **derive** $T$ to every sensor node, and equally **allocate** $D$ into the set of $T$, where $T$ is Scheduling Task, $D$ is Sensing Data

**Iteration:**
2. **receive** any message from $s_m$, $s_m \in T$;
3. **if** (message∈workload's neighboring information) **then fill** in corresponding overlay;
4. **if**($w_m = 0$) **then**
5. **search** hotspot task $s_m$ by $MAX\{t_1, t_2, \ldots\}$;
6. get the sensing data's size
7. **if** ($w$ match the tradeoff condition) **then**
8. **mount** $LB_t$
9. **send(** $LB_t$, $s_n$, $w$**) to** $s_n$;r
10. **interrupt** $s_n$, **send(**$s_n$, $w$**) to** $s_n$;
11. **if** ($T$ is Null) **then exit**, all task has been finished.

Secondly, while one task has been finished, *Task Counter's Algorithm* can get the condition and granularity of load balancing from the Task's optimal size.

**Epoch 2:** *Task Counter's Algorithm*

1. **interrupt** initialization, **set** interrupting time, $j \leftarrow 0$;
2. **if**($j$<=*uptoround*−1) **then**
3. sensors nodes' data transferring part
4. **if**($j$= =*uptoround*−1) **then**
5. **send** workload information to Scheduler, **return** load balancing parameters;
6. **receive** $LB_t$, $s_n$ and $w$;
7. **if** ($LB_t$ trigger scheduling) **then**
8. **receive** $w$ packets from $s_n$;
9. *uptoround* ← $w$;
10. **update** node's data distribution;
11. $i \leftarrow i+1$;
12. **goto** (2);
13. **send** task's finishing information to the scheduler.

Finally, the *Interrupting Procedure* is as follows:

**Epoch 3:** *Interrupting Procedure*

1. **receive** message $w$ from scheduler;
2. **send** $w$ packets to task $n_i$;
3. *uptoround* ← *uptoround*−1;
4. **update** the sensor node's data distribution.

## 5. Workload Analysis with Data Communication

In this paper, the workload model is developed in order to examine the effect for load balancing strategy. One strategy's workload is primarily composed of these four parts:

- Workload for Computing New Distribution
- Synchronization Workload
- Workload for Data communication
- Workload for Dispatching Sensors

Let $\mathbb{V}$ denote the number of sensors, where $\mathbb{V}'$ is the physical connection number of the given sensor, and $m$ is round number, $n$ is iteration number. Let $\mathbb{E}_m$ be every sensor's independent load function, $\mathbb{E}_{max}$ is every sensor's maximal load, $\mathbb{E}_m(k)$ is the workload of sensor $m$ in $k$th stationary load time interval, and $B$ is networks bandwidth.

Firstly, as far as *Workload for Computing New Distribution* is concerned, workload can be defined by $\varepsilon$, and normally it is a small quantum. In this distributed strategy, every sensor node has the cost. Moreover, the workload is relatively smaller in the local strategy because of $\mathbb{V}'$ nodes in the grid event locations, and this difference can be ignored.

Secondly, *Synchronization Workload* is the hotspot node forwards interrupting request to the neighboring nodes in the synchronization process, which consequently reflects their performance and energy parameters to the *reconfiguration tracker* for load balancing. The workload can be calculated in terms of the category of data aggregation.

### 5.1. Definition 1: Workload for Dispatching Sensors

The demanding message number can be analyzed according to data transmission and sensors rescheduling. The iteration distribution can be identified as $\gamma_m(n)$, and $\chi_m(n)$ defines the no-executing iteration number for sensor $m$ after the time of the synchronization $n$. $\Phi_{(n)} = \sum_{m=1}^{\mathbb{V}} \chi_m(n)$, and $t_n$ depicts the time cost in the synchronization $n$.

Based on the influence of discrete workload, the sensor's *effective speed* is in the inverse proportion to the sensor's workload, thus the computation expression is

$S_m/(\mathbb{E}_m(k)+1)$, and $\mathbb{E}_m(k)\in\{0,...,\mathbb{E}_{\max}\}$. For the anterior synchronization sensor states the performance measurement in different mechanism, the sensor's performance can be determined by the average effective speed actually. As for the synchronization $n-1$ happens among the workload $x$ steady time, take $x=\lceil t_{n-1}/l_{\max}\rceil$ for example, and $y=\lceil t_n/l_{\max}\rceil$ analogously. $\chi_m(n)$ is the effective workload of sensor $m$ between the synchronization $n$ and the synchronization $n-1$, and the average effective speed of sensor $m$ between these two synchronization is:

$$\begin{aligned}\varepsilon_m(n)&=\frac{\sum_k^b=aS_m/(\mathbb{E}_m(k)+1)}{b-a+1}\\&=S_m/\left(\frac{b-a+1}{\sum_{k=a}^b 1/(\mathbb{E}_m(k)+1)}\right)=S_m/\theta_m(n)\end{aligned}\tag{1}$$

As for the finished iteration, the synchronization $n$'s influence can be analyzed in the agreement loop, and the iteration's time cost is the same in the loop. To be more specific, the iteration's time cost can be depicted as $\tau$ in agreement loop. After synchronization $n-1th$ finishing, the iteration $\gamma_n(n-1)$ can be allocated to every sensor. Furthermore, $f$ defines the first sensor which has finished the task allocation, then the time cost in sensor $f$ is:

$$t=t_n-t_{n-1}=\frac{\gamma_f(n-1)*\tau}{\varepsilon_f(n)}\tag{2}$$

The unfinished iteration number in sensor $m$ can be obtained from the old iteration distribution subtracting the iteration number of $t$ time interval:

$$\chi_m(n)=\gamma_m(n-1)-\left\lceil\frac{t*\varepsilon_m(n)}{T}\right\rceil=\gamma_m(n-1)-\gamma_f(n-1)\left(\frac{\varepsilon_m(n)}{\varepsilon_f(n)}\right)\tag{3}$$

The disagreement loop can be deal with according to disagreement loop as follows, and the time interval of sensor $f$ having finished its allocated task is:

$$t=t_n-t_{n-1}=\sum_{k=1}^{\gamma_f(n-1)}\frac{\tau_k}{\varepsilon_f(n)}\tag{4}$$

Thus $k$ belongs to the set of allocating iteration for sensor $f$. The finished iteration of sensor $m$ in time $t$ can be defined as $\gamma\ll\gamma_m(n-1)$, it can be denoted as $\sum_{k'=1}^{\gamma}\frac{\tau_{k'}}{\varepsilon_m(n)}\geq t$.

Take $t$ into this expression, and shift $\varepsilon_m(n)$ into the other side:

$$\sum_{k'=1}^{\gamma}\tau_{k'}\geq\left(\frac{\varepsilon_m(n)}{\varepsilon_f(n)}\right)\sum_{k=1}^{\gamma_f(n-1)}\tau_k\tag{5}$$

So the unfinished iteration number in sensor $i$ is $\chi_m(n)=\gamma_m(n-1)-\gamma$. In the new distribution, the unfinished task number in all sensors is $\Phi(n)=\sum\chi_m(n)$, and it is in the direct proportion to sensor's average effective speed:

$$\gamma_m(n)=\left(\frac{\varepsilon_m(n)}{\sum_{k=1}^{\mathbb{V}}\varepsilon_k(n)}\right)*\Phi(n)\tag{6}$$

The former allocated task quota in every sensor is the same, thus: $\theta_m(0)=1$, $\gamma_m(0)=I(N_{ad})/\mathbb{V}$, and $\chi_m(0)=\gamma_m(0),\forall m\in V$. Supposing that information can be seized in advance, $\theta_m(0)$ will be in the direct proportion to the asynchronous sensors' initial speed or sensor's initial workload. The *recurrence function* can be proposed from the above deduction, while the new distribution and the entire iteration number for every unfinished synchronization can be got in terms of their solution. As all the tasks have been finished, the terminating condition $\Phi(\eta)=0$ happens.

Let $\xi$ denote the synchronization workload, $\eta$ depicts the demanding synchronization number, $\varepsilon$ is the computation workload in the redistribution, $\mathbb{E}(n)$ is the data transmission workload, and $\psi(n)$ defines the workload in the synchronization $n$'s communication. Based on transferred task amount, the transferred basic task unit(normally it is iteration relation) in one synchronization process:

$$\rho(n)=\frac{1}{2}\left(\sum_{m=1}^{\mathbb{V}}\left|\chi_m(n)-\gamma_m(n)\right|\right)\tag{7}$$

In terms of data transmission workload, the transmission iteration brings about array's removing. $\varsigma(n)$ defines the number of transmission task and array's message amount, and it can be obtained from the computation of the old distribution value and the new distribution value. $\rho$ belongs to the set of distribution array to be renewed. And the total workload in data transmission can be given from the expression:

$$\mathbb{E}(n)=\varsigma(n)*\mathbb{E}+\rho(j)*\sum_{\rho}\left|D_\rho/B\right|\tag{8}$$

### 5.2. Definition 2: Workload for Data Communication

Because load balancing manager has to forward the task and data transmission message to sensor node $\varsigma(n)$, this workload can be got from the centralized mechanism. The number of instructions and of messages in transmission data is the same, for these instructions are forwarded to the sensors which need to forward data. As a result, the workload for data communication is $\psi(n)=\varsigma(n)\mathbb{E}$, in terms of the centralized mechanism. And the workload for data communication $\psi(n)=0$, it is according to the distributed mechanism.

### 5.3. Definition 3: Total Workload

In the global strategy, with the solution of the above *recurrence relation* set, the data transmission workload and the synchronization amount (Expression (7)) can be got, and it can conclude the overall workload in the total

strategy:

$$TW = \eta(\xi + \varepsilon) + \sum_{n=1}^{\eta}[\mathbb{E}(n) + \psi(n)] \quad (9)$$

In the local strategy, even if the load balancing manager is asynchronous, it is not true that the grid's handling is independent with the others. This is the reason why the load balancing manager can only handle the other grid after finishing the computation of redistribution and communication.

## 6. Performance Analysis

### 6.1. Methodology

In this section we present the performance analysis of DLBO by using simulation programs in C++ programming language. The simulator is set up as follows: It is supposed that the surveillance has a square region of size $50m \times 50m$. We assume that each node produces one unit data (400 bytes) and sends it to the sink located at the bottom-right corner. All sensors act as both sources and routers. We also performed a set of experiments with different numbers of sensors and different sizes of sensing data.

#### 6.1.1. Definition 4: Balancing Factor

This influence can be modeled as every grid's balancing factor, it is dependent on the synchronization time of the different grids:

$$\Omega_g(n) = \sum_{k=1}^{v(n)}[\varepsilon + \psi_k(n)] \quad (10)$$

$v(j)$ is the number of the grids in the waited queue of the centralized load balancing manager. In the local distributed mechanism, for the inexistence of centralized load balancing, this influence does not happen ($\Omega_g(n) = 0$). Maybe it has the influence for the overlap synchronization communication, but it can also be neglected.

For the local mechanism, every grid's workload in different mechanism is not the same, and the total workload in every grid is:

$$TW_g = \eta_g(\xi + \varepsilon) + \sum_{n=1}^{\eta_g}[\mathbb{E}_g(n) + \psi_g(n) + \Omega_g(n)] \quad (11)$$

The total workload in the local strategy is the time of the last grid finishing its computation:

$$TW = MAX\{TW_1, TW_2, ..., TW_{\lceil \mathbb{V}/\mathbb{V}_n \rceil}\} \quad (12)$$

### 6.2. Simulation Results

Next, the DLBO scheme's capability to achieve load balancing has been evaluated. In this simulation, the workload is measured by the size of packets transmitted. The overall message complexity and the hotspot message complexity are identified when the load balancing overlay-based strategy is turned on (*i.e.*, the load balancing tag is 1) or off (*i.e.*, the load balancing tag is set to 0). Similarly, the same topology is used for all simulated algorithms LEACH, Reference [13], and DLBO. Communication and balancing factor simulation are two major patterns in this procedure.

Figure 3 shows the overall task execution time. As the number of nodes increases, the execution time in DLBO is decreased in terms of a change from 0.32 to 0.74. This is due to the reason that the overlay-based grid quorum has been dynamically constructed when the load balancing tracker works. With the dynamic reconfigurations, as shown in Figure 4, the balancing factor in DLBO is comparatively steady.

Figure 5 and Figure 6 compare the transmission cost of load balancing produced by the three algorithms as a result of the grid quorum on a side for a uniform load. For different data size ($K$) in average effective time, the experiment is executed 20 times. From Figure 5, the unbalancing algorithm produces the most unbalanced trees, while our basic algorithm is slightly better balanced on average than LEACH. In the best, our basic algorithm considerably outperforms both LEACH and node-centric method. Worst cases occur when the grid head is located near the edge or corner of grid quorum, so that both LEACH and node-centric method produce unbalanced sensor nodes. In contrast, our basic algorithm attempts to

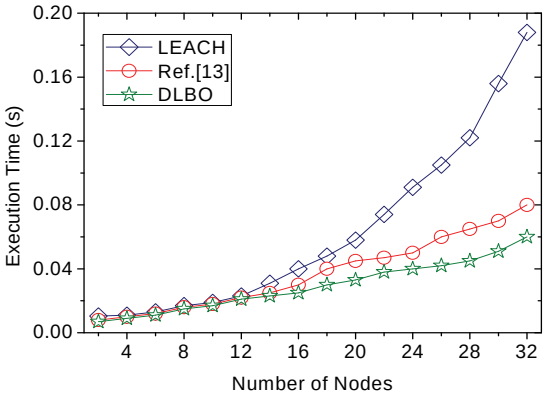



**Figure 3. Execution time comparison.**

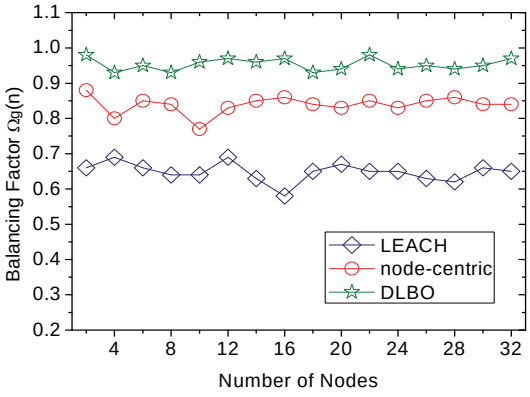



**Figure 4. Balancing factor comparison.**

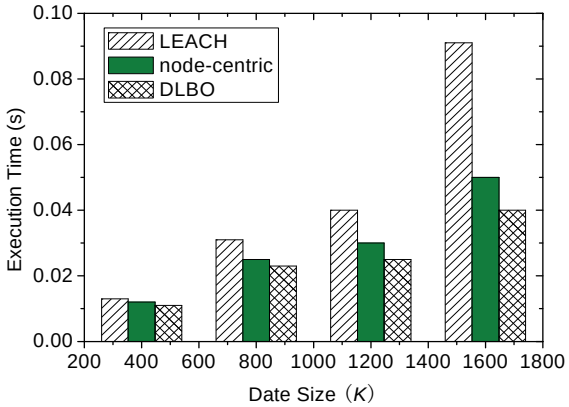



**Figure 5. Transmission cost and execution time.**

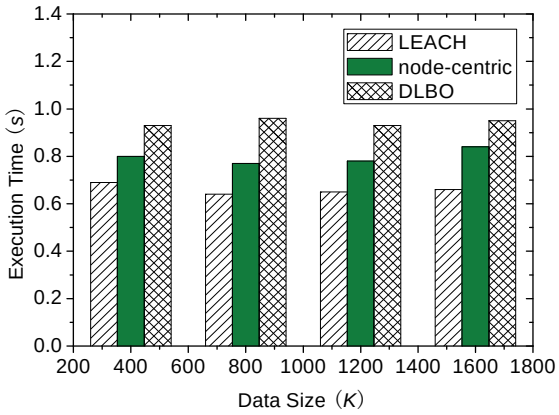



**Figure 6. Transmission cost and balancing factor.**

expand the lightest branches into open space, to avoid confining the growing scale of sensor nodes.

## 6. Conclusions

This paper addresses the load balancing in wireless sensor networks. The contributions include: 1) A dynamic nodes workload infrastructure, which is fundamental to this analysis and is believed to be a useful tool in other contexts of overlay-based WSN applications; 2) A discrete model for balancing factor, which provides a worst-case estimation for the steady data delivery between sensor nodes; 3) The optimization schemes and analysis of their reliability; and 4) simulation experiments which provide insight into the performance of dynamic load balancing from a number of major respects. In the future, we will consider how the model and the optimization schemes can be applied to wireless cognitive applications such as habitat monitoring and tracking of office equipment.

## 7. Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive suggestions to improve the quality and presentation of this paper. This paper is significantly extended from the conference version submitted to the IEEE CIT 2008.

## 8. References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communication Magazine, Vol. 40, No. 8, August 2002, pp. 102–114.

[2] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: Research challenges," Elsevier Ad Hoc Networks, Vol. 2, No. 4, pp. 351–367, 2004.

[3] Y. Law, J. Doumen, and P. Hartel, "Survey and benchmark of block ciphers for wireless sensor networks," ACM Transactions on Sensor Networks, Vol. 2, No. 1, pp. 65–93, 2006.

[4] W. Zhang, G. Cao, and T. L. Porta, "Data dissemination with ring-based index for wireless sensor networks," IEEE transactions on mobile computing, Vol. 6, No. 7, July 2007.

[5] G. Noubir and G. Lin, "Low-power DoS attacks in data wireless LANs and countermeasures," SIGMOBILE Mobile Computer Communication Review, Vol. 7, No. 3, pp. 29–30, 2003.

[6] W. Zhang, G. Cao, and T. La Porta, "Data dissemination with ring-based index for wireless sensor networks," Proceedings of IEEE International Conference on Network Protocols, pp. 305–314, November 2003.

[7] M. Khan, G. Pandurangan, and V. S. Anil Kumar, "Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 1, pp. 124–139, January 2009.

[8] X. Li, Y. Wang, P. Wan, W. Song, and O. Frieder, "Localized low-weight graph and its applications in wireless ad hoc networks," Proceedings of IEEE INFOCOM, 2004.

[9] G. D. Fatta and M. R. Berthold, "Dynamic load balancing for the distributed mining of molecular structures," IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 8, August 2006.

[10] S. Dhakal and M. M. Hayat, "Dynamic load balancing in distributed systems in the presence of delays: A regeneration-theory approach", IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 4, April 2007.

[11] F. Guo, J. Chen, W. Li, and T. Chiueh, "Experiences in building a multihoming load balancing system", INFOCOM'04, Vol. 2, pp.1241–1251, 2004.

[12] J. Gao and L. Zhang, "Load-balanced short-path routing in wireless networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 17, No. 4, pp. 377–388, April 2006.

[13] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," Global Telecommunications Conference, IEEE GLOBECOM'03, Vol. 1, pp. 548–552, December 2003.

[14] Y. Wang, W. Peng, M. Chang, and Y. Tseng, "Exploring load-balance to dispatch mobile sensors in wireless sensor networks", Computer Communications and Networks, ICCCN'07, Proceedings of 16th International Conference, pp. 669–674, August 2007.

*Wireless Sensor Network*, 2009, 1, 489-495
doi:10.4236/wsn.2009.15059 Published Online December 2009 (http://www.scirp.org/journal/wsn).

# Robust Speech Endpoint Detection in Airplane Cockpit Voice Background

**Hongbing CHENG[1], Ming LEI[2], Guorong HUANG[1], Yan XIA[3]**
[1]*College of Engineering, Air Force Engineering University, Xi'an, China*
[2]*People's Liberation Army* 95340 *Unit, Tianyang, China*
[3]*Air Force Equipment Research Academy, Beijing, China*
*Email*: *newcheng*2008@*yahoo.com.cn*, *newcheng*2008@163.*com*
*Received July* 5, 2009; *revised June* 7, 2009; *accepted June* 24, 2009

## Abstract

A method of robust speech endpoint detection in airplane cockpit voice background is presented. Based on the analysis of background noise character, a complex Laplacian distribution model directly aiming at noisy speech is established. Then the likelihood ratio test based on binary hypothesis test is carried out. The decision criterion of conventional maximum a posterior incorporating the inter-frame correlation leads to two separate thresholds. Speech endpoint detection decision is finally made depend on the previous frame and the observed spectrum, and the speech endpoint is searched based on the decision. Compared with the typical algorithms, the proposed method operates robust in the airplane cockpit voice background.

**Keywords:** Complex Laplacian Model, Maximum A Posterior Criterion, Likelihood Ratio Test, Speech Endpoint Detection, Airplane Cockpit Voice

## 1. Introduction

The information recorded by airplane cockpit voice recorder is called cockpit voice for short. Cockpit voice background is non-human voice in cockpit voice. It will take significant effect to pick-up voice information of cockpit voice in understanding the station of pilot, investigating the fly accident and finding out causes of accident. Speech endpoint detection is the base of speech tone, and its purpose is to distinguish speech segment and non-speech segment in speech signal [1]. In the airplane communication system, voice background has many characteristics: excessive kinds, complex, non-calm, transient and broad frequency. It makes up of engine noise, air current voice when it is flying, activity voice of manipulated component, diversified switch voice, alarm voice and so on. Especially prophase of airplane wrecking, noise background energy is very strong. The signal-to-noise falls obviously [2]. How to distinguish speech signal and noise signal in cockpit voice background is still a difficulty. Many researchers put forward various algorithms, such as based on entropy [3–5], cepstral feature [6–7], higher-order statistics [8], signal recursion analysis [9] etc., which are not ideal in the circumstance.

Recently years, speech endpoint detection based on statistical model get effective evolvement [10–11], especially the method based on Gaussian mixture model (GMM) [12], which establishes models of pure speech and noise respectively, and makes use of likelihood ratio test (LRT) and maximum probability criterion to judge the station of current frame, and exhibits preferable veracity. Because cockpit voice background has traits of abnormality and complexity, and has no prior information, it is impossible to establish statistical model of noise. Goodness-of-test (GOF) in literature [13] checkout that complex Laplacian model is better than traditional Gaussian model in any noise environment.

This paper imports complex Laplacian distribute model to describe the whole speech which include noise. Aiming at the defect that traditional statistical model analysis every frame signal station distribution absolutely, it thought about interframe relativity sufficiently. Then, it gained two kinds of thresholds of speech station and non-speech station respectively. In the judge criterion, it will adjust threshold automatically depending on previous frame and the observed spectrum to judge the appear or non-appear speech station. So, it achieved cockpit voice background robust speech endpoint detection.

## 2. Speech Endpoint Detection Based on GMM and LRT

Recently years, speech endpoint detection based on GMM [12] gets effective evolvement [14], which establishes

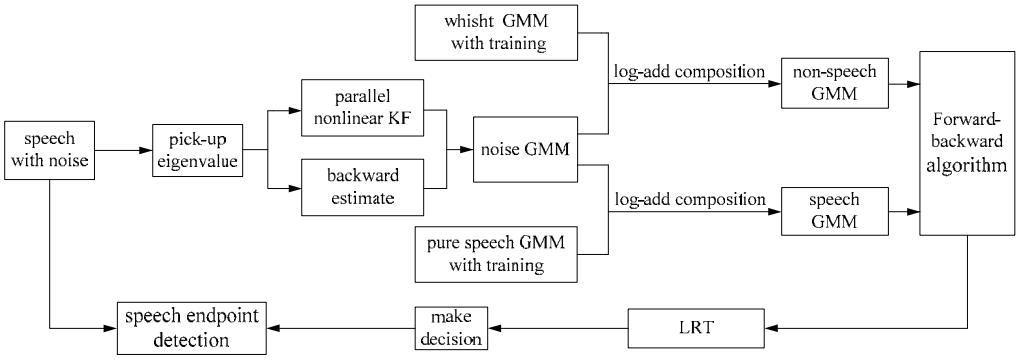



**Figure 1. The speech endpoint detection algorithm flow chart based on GMM and LRT.**

models of pure speech and noise respectively, and makes use of LRT and maximum probability criterion to judge the station of current frame, and exhibits preferable veracity. The algorithm flow chart based on GMM and LRT is showed in Figure 1.

## 2.1. Mathematical Describing of Statistical Models

Hidden Markov models (HMM), as a statistical model of speech signal, can describe the produce process of speech signal accurately. The method of speech endpoint detection based on statical models makes use of LRT to differentiate the speech frame and non-speech frame. Figure 2 shows the analysis platform of speech endpoint detection based on speech or non-speech transfer model [10] of every station.

where,

$H_0$: non-speech station in cockpit voice;

$H_1$: speech station in cockpit voice;

$a_{i,j}$: transfer probability from i to j,

$a_{i,j} = p(q_t = H_j \mid q_{t-1} = H_i)$, i,j=0 or 1;

$b_j(\boldsymbol{O}_t)$: the probability when the output of t frame cockpit voice is j station, $b_j(\boldsymbol{O}_t) = p(\boldsymbol{O}_t \mid q_t = H_j)$;

$\boldsymbol{O}_t$: the L dimension station vector of the t short time amplitude.

The way of distinguish speech frame and non-speech is to estimate the station $q_t$ of t frame short time amplitude on the condition of $\boldsymbol{O}_{0:t} = \{\boldsymbol{O}_0, \cdots, \boldsymbol{O}_t\}$. The compute formula of conditional probability density $p(q_t \mid \boldsymbol{O}_{0:t})$ is:

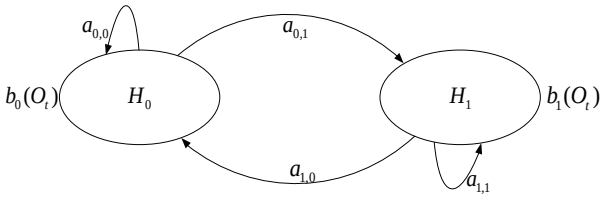



**Figure 2. Speech/non-speech transfer model.**

$$p(q_t \mid \boldsymbol{O}_{0:t}) = p(\boldsymbol{O}_{0:t}, q_t) / p(\boldsymbol{O}_{0:t}) \propto p(\boldsymbol{O}_{0:t}, q_t) \quad (1)$$

Applying one rank Markov chain recursion formula, the combine probability $p(\boldsymbol{O}_{0:t}, q_t)$ of Formula (1) can be showed as:

$$p(\boldsymbol{O}_{0:t}, q_t) = \sum_{q_{t-1}} p(q_t \mid q_{t-1}) p(\boldsymbol{O}_t \mid q_t) p(\boldsymbol{O}_{0:t-1}, q_{t-1}) \quad (2)$$

$p(\boldsymbol{O}_{0:t}, q_t)$ usually called as forward probability $a_{j,t}$, combining $a_{i,j}$ with $b_j(\boldsymbol{O}_t)$:

$$\alpha_{j,t} = a_{0,j} b_j(\boldsymbol{O}_t) \alpha_{0,t-1} + a_{1,j} b_j(\boldsymbol{O}_t) \alpha_{1,t-1} \quad (3)$$

Finally, we can get station $q_t$ through likelihood ratio threshold $R_t = \alpha_{1,t} / \alpha_{0,t}$:

$$q_t = \begin{cases} H_0 & R_t < Threshold \\ H_1 & R_t \geq Threshold \end{cases} \quad (4)$$

For example, if we can ascertain observed that signal $q_t$ is in station $H_1$, comparatively, $q_t$ is speech frame.

## 2.2. The Computation of Probability Density Function Based on GMM

In Formula (3), The computation of $b_j(O_t)$ take significant effect in the precision of endpoint detection. It is more flexible and more applicable to use the method based on GMM of log-mail spectrum than to use the method based on prior and posterior signal-to-noise, so that the precision of estimate of $b_j(O_t)$ will be higher.

$$b_j(\boldsymbol{O}_t) = \sum_{k=1}^{k} \omega_{j,k} \prod_{l=0}^{L-1} \frac{1}{\sqrt{2\pi}\sigma_{j,k,l}} \exp\left\{ -\frac{(O_{t,l} - \mu_{j,k,l})^2}{2\sigma_{j,k,l}^2} \right\} \quad (5)$$

where, $\omega_{j,k}$ is the k mixture weight of gauss distribution of GMM; $O_{t,l}$ is the Lth element of $\boldsymbol{O}_t$; $\mu_{j,k,l}$ is the average of $O_{t,l}$; $\sigma_{j,k,l}^2$ is the variance of $O_{t,l}$. In this method, if we know the average vector of whisht speech GMM, pure speech GMM and noise, we can figure

out real time noise GMM and GMM with noise through log-add composition [15] (LAC), so we can gain $b_j(\boldsymbol{O}_t)$.

LAC showed as:

$$\mu_{j,k,l} = \mu_{S,j,k,l} + \log(1+\exp(\mu_{N,l} - \mu_{S,j,k,l})) \quad (6)$$

where $\mu_{S,j,k,l}$ is the average of whisht (j=0) or speech (j=1) GMM in log-mail spectrum, $\mu_{N,l}$ is the average of noise.

In the method, we can establish whisht and pure speech GMM by training pure speech. The average of noise ($\mu_{N,l}$) can be estimated one by one frame by using parallel nonlinear KF. The noise GMM and GMM with noise will update timely with $\mu_{N,l}$.

The traditional likelihood estimation is gained by forward estimating with present and past parameter. The value of t+1,…,T is still the important factor of time sequence estimate. Processing likelihood estimate with the future frame is backward estimate. The definition of backward estimate is:

$$p(\boldsymbol{O}_{0:T}, q_t) = p(\boldsymbol{O}_{0:t}, q_t)\, p(\boldsymbol{O}_{t+1:T} \mid q_t) \quad (7)$$

Similar with Formula (2), conditional probability is showed as:

$$p(\boldsymbol{O}_{t+1:T} \mid q_t) = \sum_{q_{t+1}} p(q_{t+1} \mid q_t)\, p(\boldsymbol{O}_{t+1} \mid q_{t+1})\, p(\boldsymbol{O}_{t+2:T} \mid q_{t+1}) \quad (8)$$

$p(\boldsymbol{O}_{t+1:T} \mid q_t)$ has usually called forward probability $\beta_{j,t}$, combining $a_{i,j}$ with $b_j(\boldsymbol{O}_t)$:

$$\beta_{j,t} = a_{i,0} b_0(\boldsymbol{O}_{t+1}) \beta_{0,t+1} + a_{i,1} b_1(\boldsymbol{O}_{t+1}) \beta_{1,t+1} \quad (9)$$

Usually, backward estimate begin from terminal of tested signal, but in the test of endpoint, the terminal is unknown. So we introduce back modularize estimation. It is begin from T=t+b, where b is a constant. When b=0, backward estimate equal to does not process.

We can conclude from the definition of the Forward-Backward (F-B) algorithm that: $p(\boldsymbol{O}_{0:t}, q_t = H_t) = \alpha_{j,t}\beta_{j,t}$. We can gain likelihood ratio $R_t$ by applying likelihood ratio test.

$$R_t = \frac{p(O_{0:T}, q_t = H_1)}{p(O_{0:T}, q_t = H_0)} = \frac{\alpha_{1,t}\beta_{1,t}}{\alpha_{0,t}\beta_{0,t}} \quad (10)$$

Finally, substituting $R_t$ in Formula (4), we get the station value $q_t$ of speech endpoint detection.

## 3. The Establishing of Complex Laplacian Distribution Model

Speech endpoint detection is processed one by one frame. Every frame includes M sampling. In generally, speech signal is thought as windless signal in short period (10~30ms). We can suppose that speech signal with noise is statistical irrelated complex Laplacian random course. We denote coefficient vector of discrete fourier transform (DFT) of M dimension noise speech with $\mathbf{X}(t)$:

$$\mathbf{X}(t)=[X_1(t), X_2(t), \cdots X_k(t) \cdots X_M(t)]^T$$

If $X_{\mathrm{k(R)}}$ and $X_{\mathrm{k(I)}}$ denote real part and imaginary part of $X_{\mathrm{k}}$ respectively, the probability density distribution of $X_{\mathrm{k(R)}}$ and $X_{\mathrm{k(I)}}$, according to the Laplacian probability distribution, can be written as:

$$p(X_{\mathrm{k(R)}}) = \frac{1}{\sigma_{\mathrm{x}}} \exp\{-\frac{2\left|X_{\mathrm{k(R)}}\right|}{\sigma_{\mathrm{x}}}\} \quad (11)$$

$$p(X_{\mathrm{k(I)}}) = \frac{1}{\sigma_{\mathrm{x}}} \exp\{-\frac{2\left|X_{\mathrm{k(I)}}\right|}{\sigma_{\mathrm{x}}}\} \quad (12)$$

where, $\sigma_{\mathrm{x}}^2$ is the variance of $X_{\mathrm{k}}$. If the real part and imaginary part of $X_{\mathrm{k}}$ are uncorrelated, the distribution density of $X_{\mathrm{k}}$ can be written as:

$$\begin{aligned} p(X_{\mathrm{k}}) &= p(X_{\mathrm{k(R)}}) \cdot p(X_{\mathrm{k(I)}}) \\ &= \frac{1}{\sigma_{\mathrm{x}}^2} \exp\{-\frac{2(\left|X_{\mathrm{k(R)}}\right| + \left|X_{\mathrm{k(I)}}\right|)}{\sigma_{\mathrm{x}}}\} \end{aligned} \quad (13)$$

## 4. The Likelihood Ratio Test Based on Hypothesis Test

Speech endpoint detection can be regarded as a binary hypothesis issue:

$$H_0 : \textit{speech donot appear} \quad \mathbf{X}(t) = \mathbf{N}(t)$$
$$H_1 : \textit{speech appear} \quad \mathbf{X}(t) = \mathbf{N}(t) + \mathbf{S}(t)$$

where, $H_0$ denote the situation of speech not appearing, $H_l$ denote the situation of speech appearing, $N(t)$ and $S(t)$ denote DFT coefficient vector of background noise and pure speech respectively. The conditional probability density of noise under the situation of $H$ and $H_l$ can be written as:

$$p(X_{\mathrm{k}} \left| \mathrm{H_n} = \mathrm{H_0}\right.) = \frac{1}{\lambda_{\mathrm{n,k}}} \exp\{-\frac{2(\left|X_{\mathrm{k(R)}}\right| + \left|X_{\mathrm{k(I)}}\right|)}{\sqrt{\lambda_{\mathrm{n,k}}}}\} \quad (14)$$

$$p(X_{\mathrm{k}} \left| \mathrm{H_n} = \mathrm{H_1}\right.) = \frac{1}{\lambda_{\mathrm{n,k}} + \lambda_{\mathrm{s,k}}} \exp\{-\frac{2(\left|X_{\mathrm{k(R)}}\right| + \left|X_{\mathrm{k(I)}}\right|)}{\sqrt{\lambda_{\mathrm{n,k}} + \lambda_{\mathrm{s,k}}}}\} \quad (15)$$

We can receive likelihood test of hypothesis test by Formulas (14) and (15). Likelihood ratio $\Lambda_{\mathrm{k}}$ of the kth frequency band can be denoted as:

$$\Lambda_{\mathrm{k}} \equiv \frac{p(X_{\mathrm{k}} \left| \mathrm{H_n}=\mathrm{H_1} \right.)}{p(X_{\mathrm{k}} \left| \mathrm{H_n}=\mathrm{H_0} \right.)} \tag{16}$$

Because the signal samples $\mathbf{X}_{\mathrm{k}}(k=1,2\cdots M)$ are uncorrelated and have the same distribution, the likelihood ratio of M dimension observed vector of two hypothesis is:

$$\begin{aligned}\Lambda &= \frac{p(\mathbf{X}\left|H_n=H_1\right.)}{P(\mathbf{X}\left|H_n=H_0\right.)} = \prod_{k=0}^{M-1}\frac{p(X_k\left|H_n=H_1\right.)}{p(X_k\left|H_n=H_0\right.)} \\ &= \prod_{k=1}^{M}\frac{1}{1+\xi_k}e^{\left\{2(\left|X_{k(R)}\right|+\left|X_{k(I)}\right|)((\left|X_k\right|-\sqrt{\lambda_{n,k}})/\sqrt{\left|X_k\right|\lambda_{n,k}})\right\}}\end{aligned} \tag{17}$$

where, $\xi_k$ is the forward signal-to-noise, define as $\xi_k = \frac{\lambda_{s,k}}{\lambda_{n,k}}$, we assume that all the frequency vectors are uncorrelated.

We can know from Formula (17) that $\lambda_{\mathrm{n,k}}$ and $\xi_k$ have great influence on the veracity of likelihood ratio test. The estimate of $\lambda_{\mathrm{n,k}}$ of traditional speech endpoint detection updates in speech intermission time. The power spectrum changes when speech appears in cockpit voice background, where the impulse noise does not appear in other time. So, the estimation of noise power spectrum should be updated really both when speech appear and when speech do not appear. We adopt the method of long time power spectrum smooth to compute $\lambda_{\mathrm{n,k}}$ [16]. From [16] we know that the estimation of the kth fourier transform coefficient variance is:

$$\hat{\lambda}_{n,k}(t+1) = \varsigma_{\lambda_n}\hat{\lambda}_{n,k}(t) + (1-\varsigma_{\lambda_n})E[\left|N_k(t)\right|^2\left|X_k(t)\right.] \tag{18}$$

where, $\hat{\lambda}_{n,k}(t)$ is the estimation of $\lambda_{n,k}(t)$ and $\varsigma_{\lambda_n}$ is the smooth coefficient. Considering the two situation of speech appearing and not appearing, the estimation of the noise power spectrum of current frame is:

$$\begin{aligned}&E[\left|N_k(t)\right|^2\left|X_k(t)\right.] \\ &= E[\left|N_k(t)\right|^2\left|X_k(t), H_n=H_0\right.]P(H_n=H_0\left|X_k(t)\right.) \\ &\quad + E[\left|N_k(t)\right|^2\left|X_k(t), H_n=H_1\right.]P(H_n=H_1\left|X_k(t)\right.)\end{aligned} \tag{19}$$

where: $E[\left|N_k(t)\right|^2\left|X_k(t), H_n=H_0\right.] = \left|X_k(t)\right|^2$

$$\begin{aligned}&E[\left|N_k(t)\right|^2\left|X_k(t), H_n=H_1\right.] \\ &= (\frac{\hat{\xi}_k(t)}{1+\hat{\xi}_k(t)})\hat{\lambda}_{n,k}(t) + (\frac{1}{1+\hat{\xi}_k(t)})^2\left|X_k(t)\right|^2\end{aligned}$$

The prior signal-to-noise $\xi_k$ can be estimated, following literature [17], as:

$$\hat{\xi}_k(t) = (1-\varsigma_{SNR})\frac{\left|\hat{S}_k\right|^2(t-1)}{\hat{\lambda}_{n,k}(t-1)} + \varsigma_{SNR}P[\hat{\gamma}_k(t)-1] \tag{20}$$

where, $P[x]=\begin{cases}x, & x\ge 0\\ 0, & others\end{cases}$, $\gamma_k = \frac{\left|X_k\right|^2}{\lambda_n}$ is posterior signal-to-noise, $\hat{\gamma}_k(t)$ is it's estimation, $\varsigma_{SNR}$ is the weight of direct judge estimate, $\left|\hat{S}_k\right|^2(t-1)$ is the speech amplitude breadth of pre-frame which has estimated by using MMSE.

We can gain likelihood estimate by substituting (18) and (20) in (7).We can judge whether speech appear or not based on traditional MAP criterion [18].

## 5. The Judge Criterion Based on Conditional MAP

The decision-making of speech endpoint detection based on traditional MAP criterion is:

$$\frac{p(H_n=H_1\left|\mathbf{X}\right.)}{P(H_n=H_0\left|\mathbf{X}\right.)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{21}$$

where, $H_n$ denote the nth frame right hypothesis. According to Bayesian formula, the criterion of likelihood ratio is:

$$\frac{p(\mathbf{X}\left|H_n=H_1\right.)}{P(\mathbf{X}\left|H_n=H_0\right.)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{p(H_n=H_0)}{P(H_n=H_1)} \tag{22}$$

However, the speech appear model $H_1$ include speech do not appear model $H_0$. It causes the computing of likelihood ratio partial to $H_1$ [10]. In order to make up the difference, the Formula (22) is adjusted as:

$$\frac{p(\mathbf{X}\left|H_n=H_1\right.)}{P(\mathbf{X}\left|H_n=H_0\right.)} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha\frac{p(H_n=H_0)}{P(H_n=H_1)}, \quad \alpha>1 \tag{23}$$

The speech endpoint detection of interframe has strong relativity. The probability of that speech frame's next frame turns into speech frame is very large. The relativity was validated by FSM [11].

The paper combined the relativity of interframe with MAP criterion. It is different from traditional forward probability $P(H_n\left|\mathbf{X}\right.)$. The present observed value and the decision-making of pre-frame were used for computing forward probability. It was denoted as $P(H_n\left|\mathbf{X},H_{n-1}\right.)$, and the decision-making verification of speech endpoint detection decision-making was adjusted:

$$\frac{p(H_n=H_1\left|\mathbf{X},H_{n-1}=H_i\right.)}{P(H_n=H_0\left|\mathbf{X},H_{n-1}=H_i\right.)} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \quad i=0,1 \tag{24}$$

where, $\alpha$ is threshold. The estimation of likelihood ratio becomes:

$$\frac{p(\mathbf{X}|H_n = H_1, H_{n-1} = H_i)}{P(\mathbf{X}|H_n = H_0, H_{n-1} = H_i)} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \frac{p(H_n = H_0|H_{n-1} = H_i)}{P(H_n = H_1|H_{n-1} = H_i)}, \quad i = 0,1 \tag{25}$$

In the actual cockpit voice, because of the lack of prior information, distributed parameters, $p(\mathbf{X}|H_n = H_1, H_{n-1} = H_i)$ and $P(\mathbf{X}|H_n = H_0, H_{n-1} = H_i)$, have not been estimated, and the distributed parameters of current frame were decided by the current observed value. So it was predigested as:

$$p(\mathbf{X}|H_n = H_j, H_{n-1} = H_i) = P(\mathbf{X}|H_n = H_j), \quad i = 0,1, \quad j = 0,1. \tag{26}$$

Formula (25) is changed to:

$$\frac{p(\mathbf{X}|H_n = H_1)}{P(\mathbf{X}|H_n = H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \frac{p(H_n = H_0|H_{n-1} = H_i)}{P(H_n = H_1|H_{n-1} = H_i)}, \quad i = 0,1 \tag{27}$$

Its form of log is:

$$\log \frac{p(\mathbf{X}|H_n = H_1)}{P(\mathbf{X}|H_n = H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \log\left[\alpha \frac{p(H_n = H_0|H_{n-1} = H_i)}{P(H_n = H_1|H_{n-1} = H_i)}\right] \triangleq \eta_i, \quad i = 0,1 \tag{28}$$

The Formula (27) or (28) is the judge criterion of speech endpoint detection. $\eta_i$ is the threshold. When preframe is speech frame, $\eta_1$ will be regarded as the threshold of the current frame. When preframe is nonspeech frame, $\eta_0$ will be regarded as the threshold of the current frame. Multiple thresholds can provide more freedom and can enhance the robusticity of speech endpoint detection. Considering the relativity of interframe, parameter distribution has the trait as follows:

$$\frac{p(H_n = H_0|H_{n-1} = H_0)}{P(H_n = H_1|H_{n-1} = H_0)} > \frac{p(H_n = H_0|H_{n-1} = H_1)}{P(H_n = H_1|H_{n-1} = H_1)} \tag{29}$$

It indicates that the probability of nonspeech frame's next frame become nonspeech frame is large. When the preframe is nonspeech frame, $\eta_0$ is larger than $\eta_1$. It is all the same for speech frame.

## 6. Experimentation

In order to test the validity of the paper's algorithm, the cockpit voice background sound of airplane normal station and wrecked station have been picked up respectively, and two teams experimentation of speech endpoint detection based on GMM and the paper's algorithm have been done.

### 6.1. The Establishment of Experimentation

In environment of lab, we record 200 sentences of 6 persons (3 men and 3 women) to form storage of pure speech and training GMM. The test group makes up of other 40 sentences. Because cockpit voice background sound is complex, excessive, so its bandwidth is broad (150Hz-6800Hz), and its signal is not calm and is transient. Different kind airplanes have different cockpit voice background sounds. Its characteristics are different from F16 noise provided by group NOISEX-2. So that, the cockpit voice background sound used in simulation test was recorded in the real environment. Its sample frequency is 16KHz and quantitative change bite is 16 and single channel is format wave. We can get airplane normal station and wrecked station speech with noise group by adjusting breadth of pure speech and adding it to cockpit voice background sound. The extracting of character is showed in Table 1.

When training GMM, the GMM parameter with 25 characteristic vector (12 rank mail cepstral coefficient and its differential coefficient, short time power differential coefficient) was gained by using the expectation-maximization (EM) algorithm. The smooth coefficient $\varsigma_{\lambda_n}$, the weight $\varsigma_{SNR}$ for judging forward signal-to-noise estimation and the known threshold $\eta_i$ based on preframe should be chosen carefully to ensure the robusticity.

### 6.2. The Result of Experiment

We define that $P_d$ is the ratio that the speech frame is detected as the speech frame correctly and $P_f$ is the ratio that the nonspeech frame is detected as the speech frame. The performance of the two algorithms is depicted by the ROC curve which denote the relation of $P_d$ and $P_f$. Figure 3 shows a real example of speech endpoint detection. Its last time is 1s. Figure 4 and Figure 5 show the ROC curve, which is the cockpit voice background speech endpoint detection of airplane normal station and wrecked station, of the two algorithms.

In Figure 3, the broken line of pure speech graph is the manual mark place of speech begin point. When the air-

**Table 1. the condition of character extracting.**

| | |
|---|---|
| Sample frequency | 16kHz |
| Quantitative change bite | 16bite |
| Advance add quantity | $1\text{-}0.97z^{-1}$ |
| Length of frame | 20ms |
| Moving of frame | 10ms |
| Function of window | Hamming |

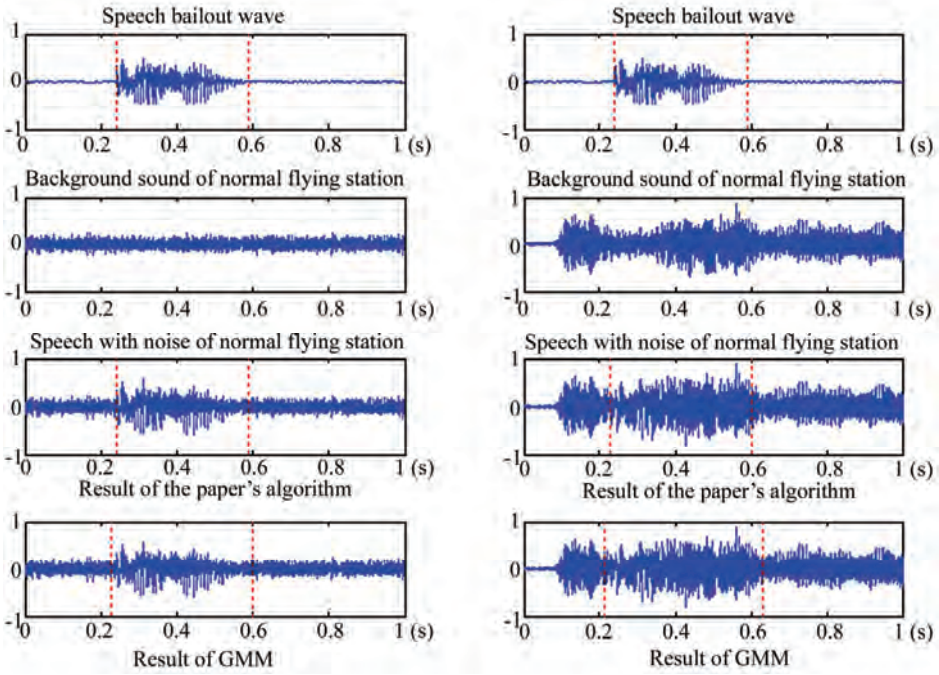



**Figure 3. A real example of speech endpoint detection.**

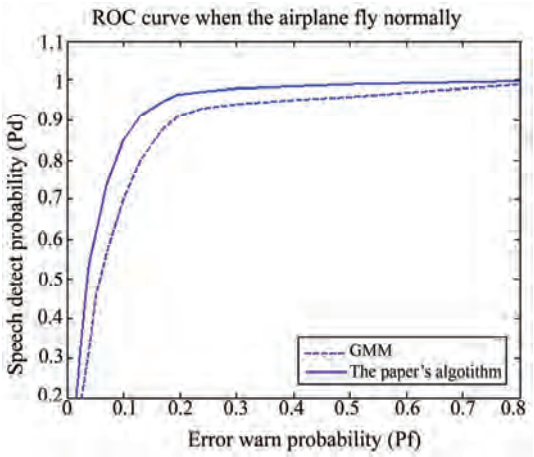



**Figure 4. The ROC curve of airplane normal station**

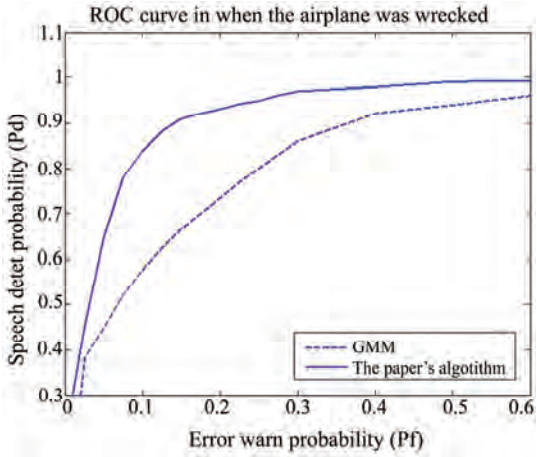



**Figure 5. The ROC curve of airplane wrecked station.**

plane fly normally, the noise in cockpit primary is smooth engine noise and the quiver noise arosed by aerosphere mussy flu. So the veracity of the speech endpoint detection result of the two algorithms almost the same. When the airplane was wrecked, the noise in cockpit is very intensive. The prior half part is the airplane speech alarm sound, the posterior half part is the alarm ring. There is strike sound of pilot pull switch in it. In the complex and nonsmooth background sound, speech was almost silenced. From Figure 3 we can see that the paper's algorithm, modelling directly for speech with noise, robuster than GMM algorithm, modelling noise and speech respectively, and gets better effect of speech endpoint detection.

From Figure 4 we can see, when the airplane fly normally, the ROC curve's best work points of GMM and the paper's algorithm are [0.180,0.885] and [0.135,0.920] respectively. Compared with GMM, The error warn probability and the detect probability of the paper's algorithm reduce 25% and increase 4% respectively. The cause of the phenomena is that the draw up precision of complex Laplacian transformation higher than that of GMM. Adding the application of the relativity of the interframe, his total precision is better than GMM. From Figure 4 we can see, when the airplane was wrecked, the speech endpoint detection algorithm of the paper is better than GMM obviously. The best work points of the two algorithms are [0.141,0.910] and [0.275,0.820] respec-

tively. Compared with GMM, The error warn probability and the detect probability of the paper's algorithm reduce 49% and increase 10% respectively. The cause of the phenomena is that GMM modeling noise and speech respectively is not applicable for the environment of wrecked station. When the airplane was wrecked, there are many kinds of noise and they are transient, which is difficult to establish a universal model. Then, the paper's algorithm models the total speech with noise directly and exhibits preferable robusticity.

## 7. Conclusions

The speech endpoint detection of airplane cockpit voice background was put forward by the paper. The two teams' experiment denotes that the algorithm can preserve preferable veracity and robusticity in the airplane normal station and wrecked station.

## 6. References

[1] Y. M. Guo, Q. Fu, and Y. H. Yan, "Speech endpoint detection in complex noise environment [J]," Journal of Acoustics, Vol. 31, No. 6, pp. 549–554, 2006.

[2] D. L. Cheng, C. J. Yi, H. Y. Yao, *et al.*, "The primary research of voice information identify methods of airplane cockpit voice recorder [J]," Control of Noise and Quiver, Vol. 3, pp. 81–84, 2006.

[3] J. L. Shen, J. W. Hung, and L. S. Lee, "Robust entropy-based endpoint detection for speech recognition in noisy environments [C]," In Proceedings of ICSLP, pp. 232–235, 1998.

[4] J. L. Shen and C. H. Yang, "A novel approach to robust speech endpoint detection in car environment [C]," In Proceedings of ICASSP, Vol. 3, pp. 1751–1754, 2000.

[5] C. Jia and B. Xu, "An improved entropy-based endpoint detection algorithm [C]," In Proceedings of ISCSLP, 2002.

[6] J. A. Haigh and J. S. Mason, "Robust voice activity detection using cepstral feature [C]," In Proceedings of IEEE TELCON'93, pp. 321–324, 1993.

[7] X. D. Wei, G. R. Hu, and X. L. Ren," Speech endpoint detection with noise using cepstral feature [J]," Journal of Shanghai Jiao Tong University, Vol. 34, No. 2, pp. 185–188, 2001.

[8] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain [J]," IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 3, pp. 217–231, 2001.

[9] R. Q. Yan and Y. S. Zhu, "Speech endpoint detection based on the analysis of signal recursion [J]," Journal of Communication, Vol. 1, pp. 35–39, 2007.

[10] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection [J]," IEEE Signal Processing Letters, Vol. 6, No. 1, pp. 1–3, 1999.

[11] A. Davis, S. Nordholm, and R. Togneri, "Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold [J]," IEEE Transactions on Audio, Speech, Language Process, Vol. 14, No. 2, pp. 412–424, 2006.

[12] M. Fujimoto, K. Ishizuka, and H. Kato, "Noise robust voice activity detection based on statistical model and parallel non-linear Kalman filter [C]," ICASSP'07, pp. 797–800, 2007.

[13] J. H. Chang, J. W. Shin, and N. S. Kim, "Likehood ratio test with complex Laplacian model for voice activity detection [C]," In Proceedings of Euro Speech, pp. 1065–1068, 2003.

[14] M. J. F. Gales, "Models based techniques for noise robust speech recognition [D]," Cambridge University, 1995.

[15] H. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition [A]," ICASSP'95 Proceedings, pp. 153–156, 1995.

[16] N. S. Kim and J. H. Chang, "Space enhancement based on global soft decision [J]," IEEE Signal Processing Letters, Vol. 7, No. 5, pp. 108–110, 2000.

[17] W. H. Shin, B. S. Lee, Y. H. Lee, *et al.*, "Speech/non-speech classification using multiple features for robust endpoint detection [C]," In Proceeding of ICAASSP, Vol. 3, pp. 1399–1402, 2000.

[18] J. J. Lei, "The research of some issues in noise robust speech identification [D]," Doctor Thesis of Beijing University of Posts and Telecommunications, 2007.

The 6th International Conference on Wireless Communications, Networking and Mobile Computing

**September 23-25, 2010, Chengdu, China**

http://www.wicom-meeting.org/2010

# Call for Papers

**WiCOM** serves as a forum for wireless communications researchers, industry professionals, and academics interested in the latest development and design of wireless systems. In 2010, WiCOM will be held in **Chengdu**, China. You are invited to submit papers in all areas of wireless communications, networking, mobile computing and applications. All papers accepted will be included in IEEE Xplore and indexed by EI Compendex and ISTP.

### Wireless Communications

- B3G and 4G Technologies
- MIMO and OFDM
- Cognitive Radio
- Coding, Detection and Modulation
- Signal Processing
- Channel Model and Characterization
- Antenna and Circuit

### Network Technologies

- Ad hoc and Mesh Networks
- Wireless Sensor Networks
- RFID, Bluetooth and 802.1x Technologies
- Network Protocol and Congestion Control
- QoS and Traffic Analysis
- Network Security
- Multimedia in Wireless Networks

### Services and Application

- Applications and Value-Added Services
- Location Based Services
- Authentication, Authorization and Billing
- Data Management
- Mobile Computing Systems

**Submission Requirement:**

The working language of the conference is English. All the papers must be submitted in IEEE electronic format. Instructions and full information on the conference are posted on the conference website. Anyone wishing to propose a special session or a tutorial should contact us: wicom@scirp.org

**Important Dates:**

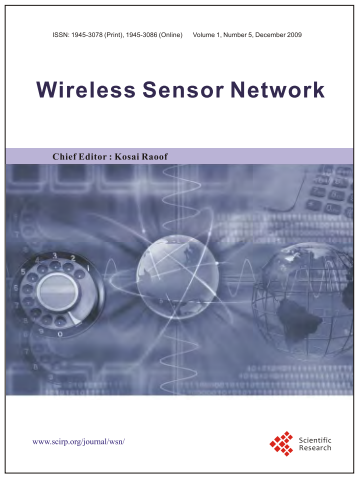
Paper Due: Feb. 28, 2010

Acceptance Notification: May. 4, 2010

**Contact Information:**

Website: http://www.wicom-meeting.org/2010

E-mail: wicom@scirp.org




# Wireless Sensor Network (WSN)

## *Call For Papers*

**http://www.scirp.org/journal/wsn**

**ISSN 1945-3078 (Print) ISSN 1945-3086 (Online)**

WSN is an international refereed journal dedicated to the latest advancement of wireless sensor network and applications. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these areas.

### Editor-in-Chief

Dr. Kosai Raoof , GIPSA LAB, University of Joseph Fourier, Grenoble, France

### Subject Coverage

This journal invites original research and review papers that address the following issues in wireless sensor networks. Topics of interest are (but not limited to):

- Network Architecture and Protocols
- Self-Organization and Synchronization
- Quality of Service
- Data Processing, Storage and Management
- Network Planning, Provisioning and Deployment
- Integration with Other System
- Software Platforms and Development Tools
- Routing and Data Dissemination
- Energy Conservation and Management
- Security and Privacy
- Developments and Applications
- Network Simulation and Platforms

We are also interested in short papers (letters) that clearly address a specific problem, and short survey or position papers that sketch the results or problems on a specific topic. Authors of selected short papers would be invited to write a regular paper on the same topic for future issues of the WSN.

### Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. Authors are responsible for having their papers checked for style and grammar prior to submission to WSN. Papers may be rejected if the language is not satisfactory. For more details about the submissions, please access the website.

### Website and E-Mail

http://www.scirp.org/journal/wsn

Email: wsn@scirp.org

*Wireless Sensor Network*, 2009, 1, 365-495
Published Online December 2009 in SciRP (http://www.SciRP.org/journal/wsn/).

# TABLE OF CONTENTS

**Volume 1 Number 5** **December 2009**



9771945307008 05