ISSN: 1945-3078 (Print), 1945-3086 (Online) Volume 1, Number 4, November 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, 233-364**
Published Online November 2009 in SciRes (http://www.SciRP.org/journal/wsn/).

TABLE OF CONTENTS

Volume 1 Number 4 **November 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, 233-244**
doi:10.4236/wsn.2009.14030 Published Online November 2009 (http://www.scirp.org/journal/wsn).

LDAP Injection Techniques

Jose Maria ALONSO[1], Antonio GUZMAN[2], Marta BELTRAN[2], Rodolfo BORDON
[1]*Informatica* 64, *S. L., Madrid, Spain*
[2]*Rey Juan Carlos University, Madrid, Spain*
Email: *chema@informatica*64.*com*, {*antonio.guzman, marta.beltran* }*@urjc.es*
Received March 12, 2009; *revised July* 3, 2009; *accepted July* 5, 2009

Abstract

The increase in the number of databases accessed only by some applications has made code injection attacks an important threat to almost any current system. If one of these applications accepts inputs from a client and executes these inputs without first validating them, the attackers are free to execute their own queries and therefore, to extract, modify or delete the content of the database associated to the application. In this paper a deep analysis of the LDAP injection techniques is presented. Furthermore, a clear distinction between classic and blind injection techniques is made.

Keywords: Web Applications Security, Code Injection Techniques, LDAP

1. Introduction

The amount of data stored in organizational databases has increased very fast in last years due to the rapid advancement of information technologies. A high percentage of these data are sensitive, private and critical to the organizations, their clients and partners.

Therefore, the databases are usually installed behind internal firewalls, protected with intrusion detection mechanisms and accessed only by applications. To access a database, users have to connect to one of these applications and to submit queries trough them to the database. The threat to databases arises when these applications do not behave properly and construct these queries without sanitizing user inputs first.

Over a 50% of web applications vulnerabilities are input validation related [1], which allows the exploitation of code injection techniques.

These attacks have proliferated in recent years causing severe damages in several systems and applications. The SQL injection techniques are the most widely used and studied [2–5] but there are other injection techniques associated to other languages or protocols such as XPath [6,7] or LDAP (Light Directory Access Protocol) [8,9].

Preventing the consequences of these kinds of attacks, lies in studying the different code injection possibilities and in making them public and well known for all programmers and administrators [10–12]. In this paper the LDAP injection techniques are analyzed in depth, because all the web applications based on LDAP trees might be vulnerable to this kind of attacks.

The key to exploiting injection techniques with LDAP is to manipulate the filters used to search in the directory services. Using these techniques, an attacker may obtain direct access to the database underlying an LDAP tree, and thereby to important corporate information. This can be even more critical because the security of many applications and services are based on LDAP directories in current single sign-on environments [13,14].

Although the vulnerabilities that lead to these consequences are easy to understand and to solve, they persist due to the lack of information about these attacks and their effects.

Although the vulnerabilities that lead to these consequences are easy to understand and fix, they persist because of the lack of information about these attacks and their effects. Though previous references to the exploitation of this kind of vulnerability exist the presented techniques don't apply to the vast majority of modern LDAP service implementations. The main contribution of this paper is the presentation and deep analysis of new LDAP injection techniques which can be used to exploit these vulnerabilities. Furthermore, a real environment has been implemented to perform different experiments in typical LDAP scenarios and to evaluate the possible danger of this kind of attacks.

It is important to note that the use of filters to limit the information that is showed to a client sending an LDAP search to the server does not increase the security of the applications, because these filters does not prevent the use of blind code injection techniques, capable of exploiting injection techniques without having detailed error messages from the server. Therefore, both, the

classic and the blind code injection techniques will be studied in depth in this paper.

This paper is organized as follows: sections 2 and 3 explain the LDAP fundamentals needed to understand the techniques presented in the following sections. Section 4 presents the two typical environments where LDAP injection techniques can be used and exemplify these techniques with illustrative cases. Section 5 describes how BLIND LDAP Injection attacks can be done with more examples. In Sections 6 and 7, some recommendations for securing systems against this kind of attack are given and, finally, Section 7 presents conclusions and future work.

2. LDAP Overview

Lightweight Directory Access Protocol is a protocol for querying and modifying directory services running over TCP/IP [15,16]. The most widely used implementations of LDAP services are Microsoft ADAM (Active Directory Application Mode, [17]) and OpenLDAP [18]. LDAP directory services are software applications that store and organize information sharing certain common attributes; the information is structured based on a tree of directory entries, and the server provides powerful browsing and search capabilities, etcetera.

LDAP is object oriented, therefore every entry in an LDAP directory services is an instance of an object and must correspond to the rules fixed for the attributes of that object. Due to the hierarchical nature of LDAP directory services read-based queries are optimized to the detriment of write-based queries. LDAP is also based on the client/server model.

The most frequent operation is to search for directory entries using filters. Clients send queries to the server and the server responds with the directory entries matching these filters. LDAP filters are defined in the RFC 4515. The structure of these filters can be summarized as:

- Filter = (filtercomp)
- Filtercomp = and / or / not / item
- And = & filterlist
- Or = | filterlist
- Not = ! filter
- Filterlist = 1*filter
- Item= simple / present / substring
- Simple = attr filtertype assertionvalue
- Filtertype = ”=” /” =”/ ”¿=” / ”¡=”
- Present = attr = *
- Substring = attr ”=” [initial] * [final]
- Initial = assertionvalue

Final = assertionvalue All the filters must be in brackets, only a reduced set of logical (AND, OR and NOT) and relational (:, ~, =, ∗) operators is available to construct them. The special character “*” can be used to replace one or more characters in the construction of the filters. Apart from being logic operators, RFC 4256 allows the use of the following standalone symbols as two special constants:

- (&) Absolute TRUE
- (|) Absolute FALSE

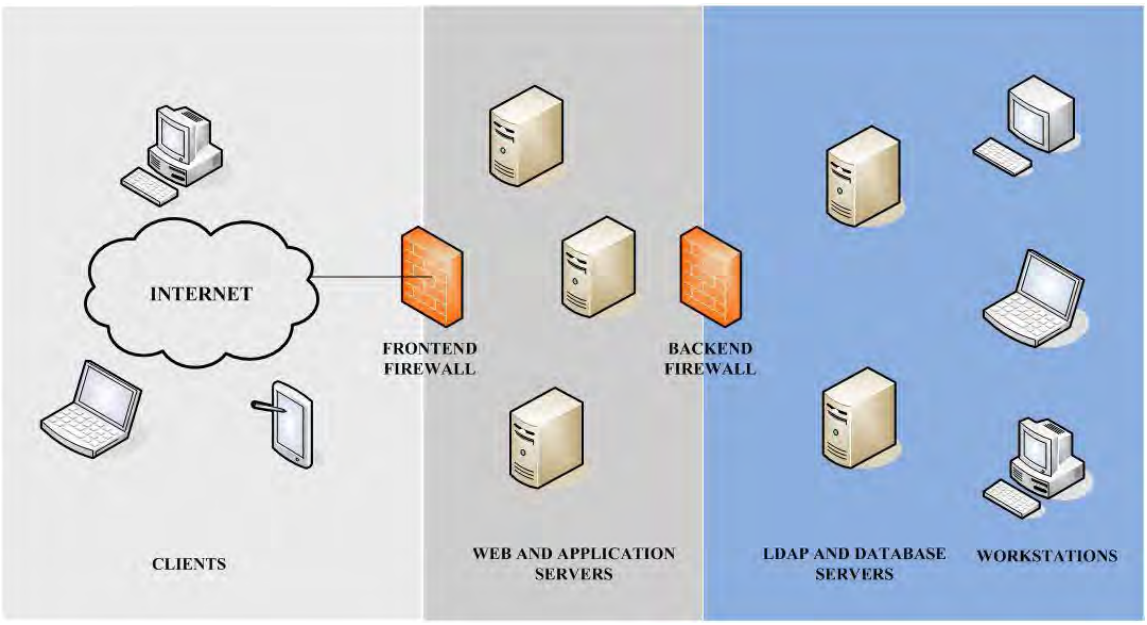


Figure 1. Typical LDAP scenario.

3. Common LDAP Environments

LDAP services are a key component for the daily operation in many companies and institutions. Directory Services such as Microsoft Active Directory, Novell E-Directory and RedHat Directory Services are based on the LDAP protocol. But there are other applications and services taking advantage of the LDAP services.

These applications and services used to require different directories (with separate authentication) to work. For example, a directory was required for the domain, a separate directory for mailboxes and distribution lists, and more directories for remote access, databases or web applications. New directories based on LDAP services are multi-purpose, working as centralized information repositories for user authentication and enabling single sign-on environments.

This new scenario increases the productivity by reducing the administration complexity and by improving security and fault tolerance. In almost every environment, the applications based on LDAP services use the directory for one of the following purposes:

- Access control (user/password pair verification, users certificates management).
- Privileges management.
- Resources management.

Due to the importance of the LDAP services for the corporate networks, the LDAP servers are usually placed in the backend with the rest of the database servers. Figure 1 shows the typical scenario deployed for corporate networks, and it is important to keep this scenario in mind in order to understand the implications of the injection techniques exposed in following sections.

4. LDAP Injection in Web Applications

LDAP injection attacks are based on similar techniques to SQL injection attacks. Therefore, the underlying concept is to take advantage of the parameters introduced by the user to generate the LDAP query. A secure Web application should sanitize the parameters introduced by the user before constructing and sending the query to the server. In a vulnerable environment these parameters are not properly filtered and the attacker can inject malicious code.

Taking into consideration the structure of the LDAP filters explained in section II and the implementations of the most widely used LDAP implementations, ADAM and OpenLDAP, the following conclusions can be drawn about the code injection. (The following filters are crafted using as value a non sanitized input from the user):

- **(attribute=value):** If the filter used to construct the query lacks a logic operator (OR or AND), an injection like value) (injected filter will result in two filter: *(attribute*=value) (injected filter). In the OpenLDAP (Figure 2) implementations the second filter will be ignored, only the first one being executed.

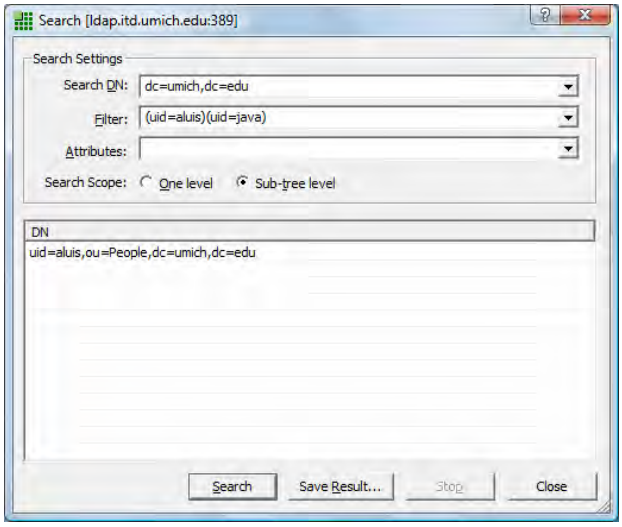


Figure 2. OpenLDAP processes only the first complete LDAP search filters. Data obtained with LDAP browser.

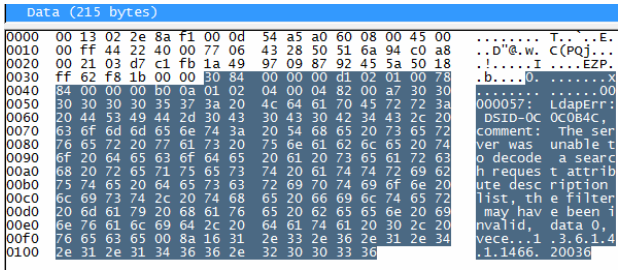


Figure 3. ADAM responses with a disconnection message in case of more than one filter are received in only one query. Data analyzed with wireshark.

In ADAM, a query with two filters isn't allowed (Figure 3). Therefore, the injection is useless.

(|(attribute=value) second filter)) or (& attribute value)(second filter)): If the filter used to construct the query has a logic operator (OR or AND), an injection like "value)(injected filter)" will result in the following filter: (&(*attribute*=***value*****)(*injected filter*)**) (*second filter*)). Though the filter is not even syntactically correct, OpenLDAP will start processing it left to right ignoring any character after the first filter is closed. Some LDAP Client web components will ignore the second filter, sending to ADAM and OpenLDAP only the first complete one, therefore allowing the injection (Figure 4).

Some application frameworks will check the filter for correctness before sending it to the LDAP server. Should this be the case, the filter has to be syntactically correct, which can be achieved with an injection like:

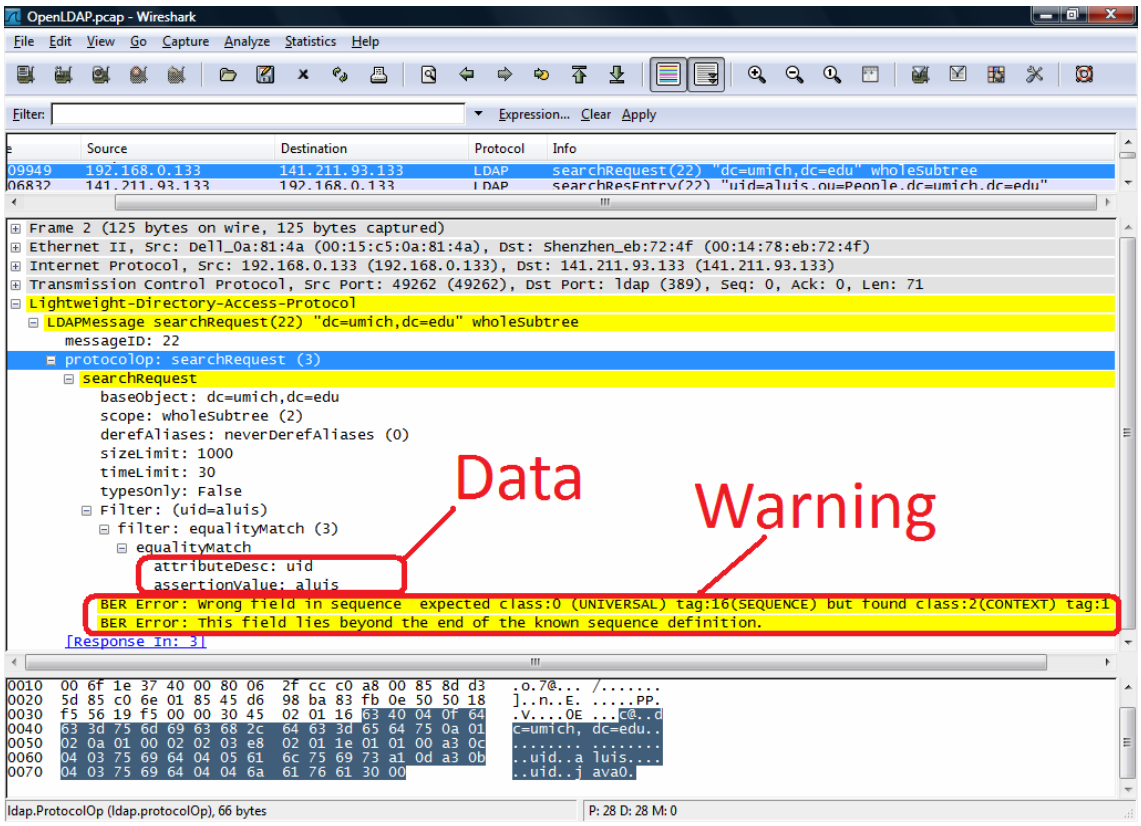


Figure 4. This is just because it tries to match the second filter to a list of attributes required. If this can be done then OpenLDAP only response with those attributes, else OpenLDAP will ignore the second filter responding with data obtained after first filter is executed and a warning message. Data analyzed with WireShark.

value)(*injected filter*))(&(1=0. This will result in two different filters, the second being ignored: (&(*attribute* = ***value)(injected filter)***)(&(1=0)(*second filter*)).

As the second filter is going to be ignored by the LDAP Server, some components won't allow an LDAP query with two filters. In these cases a special injection must be crafted in order to obtain a single-filter LDAP query. An injection like: value) (injected filter will result in the following filter: (& (attribute=value) (injected filter)(second filter)).

The typical test to know if an application is vulnerable to code injection consists of sending to the server a query that generates an invalid input. Therefore, if the server returns an error message, it is clear for the attacker that the server has executed his query and that he can exploit the code injection techniques. Taking into account the previous discussion, two kinds of environments can be distinguished: AND injection environments and OR injection environments.

4.1. AND LDAP Injection

In this case the application constructs the normal query to search in the LDAP directory with the "&" operator

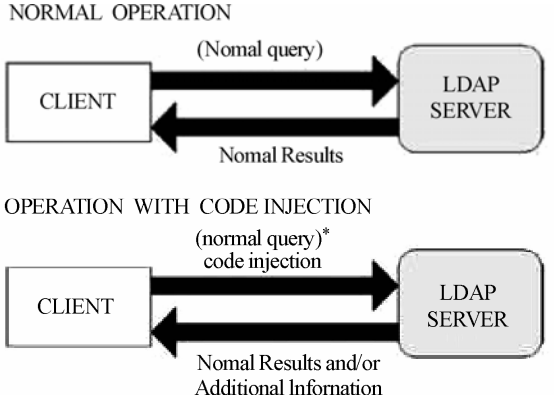


Figure 5. Injection LDAP.

and one or more parameters introduced by the user. For example:

(&(*parameter* 1= *value*1)(*parameter* 2= *value* 2))

Where *value* 1 and *value* 2 are the values used to perform the search in the LDAP directory. The attacker can inject code, maintaining a correct filter construction but using the query to achieve his own objectives.

1) *Example* 1: *Access Control Bypass:* A login page has two text box fields for entering user name and

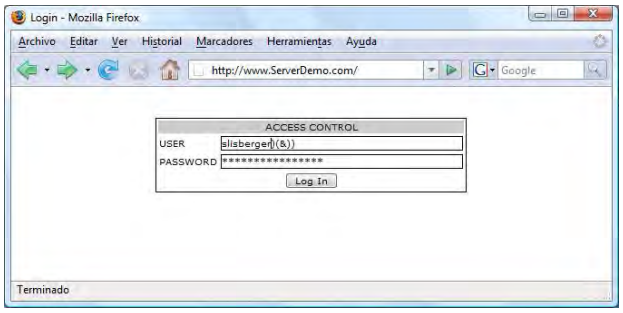


Figure 6. Loginpage with LDAP Injection.

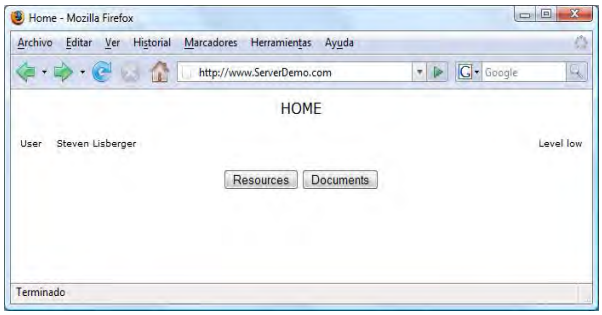


Figure 7. Home page shown to the attacker after avoiding the access control.

password (Figure 6). *Uname* and *Pwd* are the user inputs for USER and PASWORD. To verify the existence of the user/password pair supplied by a client, an LDAP search filter is constructed and sent to the LDAP server:

(&(*USER* = *Uname*) (*PASSWORD* = *Pwd*))

If an attacker enters a valid username, for example, *slisberger*, and injects the appropriate sequence following this name, the password check can be bypassed. Making *Uname*=***slisberger*)(&))** and introducing any string as the *Pwd* value, the following query is constructed and sent to the server:

(&(*USER* = *slisberger*)(&))(*PASSWORD* = *Pwd*))

Only the first filter is processed by the LDAP server, that is, only the query (&(USER=***slisberger)(&)***) is processed. This query is always true, so the attacker gains access to the system without having a valid password (Figure 7).

In case of being working with ADAM Microsoft implementation, this injection can be done just in order to obtain only one filter at the end:

USER=admin)(!(&(|PASSWORD=any))(&(USER= admin)(!(&(|)(PASSWORD = any))))

As can be seen, in this example, it is necessary to inject code in the user and password fields but it will work out not only with Microsoft implementations but with any other LDAP engine.

2) *Example* 2: *Elevation of Privileges:* For example, suppose that the following query lists all the documents visible for the users with a low security level (Figure 8):

(&(directory = documents)(security_level = low))

Where *documents* is the user entry for the first parameter and low is the value for the second (Figure 9). If the attacker wants to list all the documents visible for the high security level, he can use an injection like

*documents)(security level = *))(&(directory = documents*

resulting in the following filter:

(& (directory = documents)(security level = *))(&(directory = documents)(security level = low))

The LDAP server will only process the first filter ignoring the second one, therefore, only the following query will be processed: (&(directory = documents) (security_level=*)), while (& (directory = documents) (security_level = low)) will be ignored. As a result, a list with all the documents available for the users with all security levels will be displayed for the attacker although he doesn't have privileges to see them.

4.2. OR LDAP Injection

In this case the application constructs the normal query to search in the LDAP directory with the "|" operator and one or more parameters introduced by the user. For example:

(| (parameter 1= value1)(parameter2= value2))

Where *value1* and *value2* are the values used to perform the search in the LDAP directory. The attacker can inject

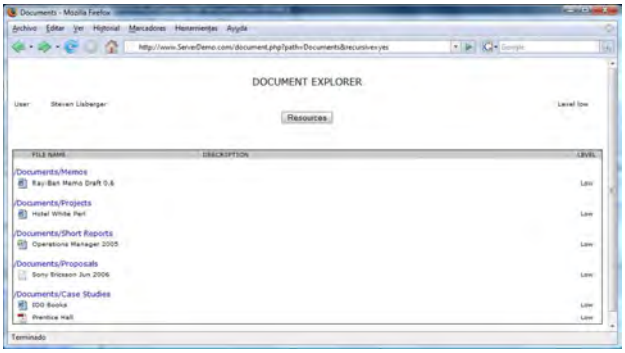


Figure 8. Low security documents.

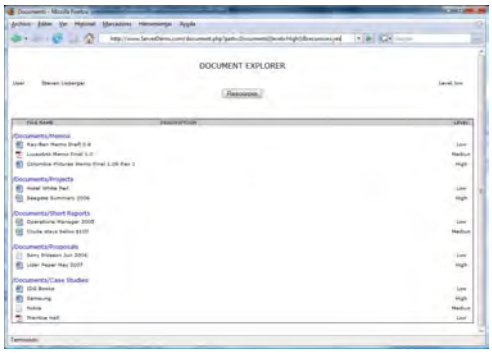


Figure 9. All security levels documents.

code, maintaining a correct filter construction but using the query to achieve his own objectives.

1) *Example* 1: *Information Disclosure:* Suppose a resources explorer allows users to know the resources available in the system (printers, scanners, storage systems, etc.). This is a typical OR LDAP Injection case, because the query used to show the available resources is:

(| (type = Rsc1)(type = Rsc2))

Rsc 1 and *Rsc 2* represent the different kinds of resources in the system. In Figure 10, *Rsc1=printer* and *Rsc 2=scanner* to show all the available printers and scanners in the system.

If the attacker enters *Rsc1= **printer)(uid=*)***, the following query is sent to the server:

*(|(type = printer)(uid = *))(type = scanner))*

The LDAP server responds with all the printer and user objects (Figure 11).

5. Blind Ldap Injection

Suppose that an attacker can infer from the server responses, although the application does not show error messages, the code injected in the LDAP filter generates a valid response (true result) or an error (false result). The attacker could use this behavior to ask the server true or false questions. These types of attacks are named "Blind Attacks". Blind LDAP Injection attacks are slower than classic ones but they can be easily implemented, since they are based on binary logic, and they let the attacker extract information from the LDAP Directory.

5.1. AND Blind LDAP Injection

Suppose a web application wants to list all available Epson printers from an LDP directory where error messages are not returned. The application sends the following LDAP search filter: *(& (object-Class=printer) (type= Epson*))* With this query, if there are any Epson printers available, icons are shown to the client, otherwise no icon is shown. If the attacker performs a Blind LDAP injection attack *injecting *) (objectClass = *)) (& (objectClass = void*, the web application will construct the following LDAP query:

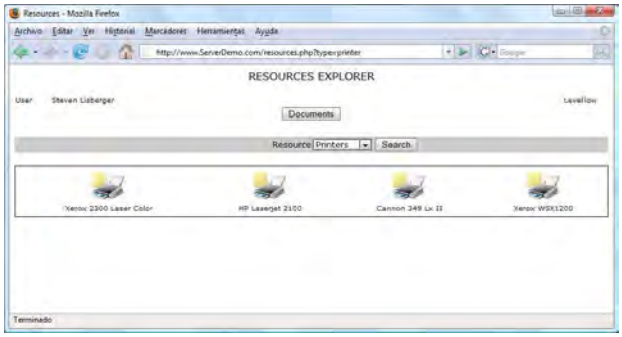


Figure 10. Resources available to the user from the resources consoles management.

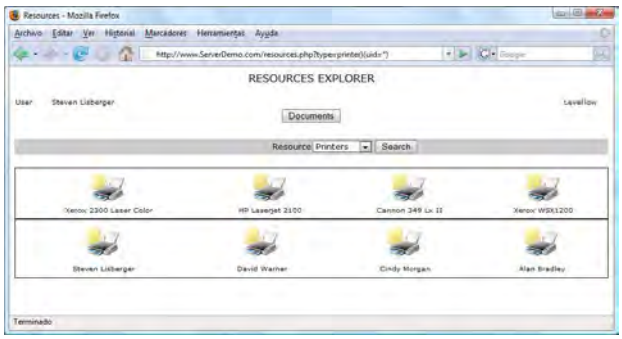


Figure 11. Resources available to the user from the resources consoles management.

*(&(objectClass = *)(objectClass=*))*

(&(objectClass=void)(type = Epson))*

Only the first complete LDAP filter will process:

*(&(objectClass = *)(objectClass = *))*

As a result, the printer icon must be shown to the client, because this query always obtains results: the filter *objectClass=** always returns an object. When an icon is shown the response is true, otherwise the response is false. From this point, it is easy to use blind injection techniques. For example, the following injections can be constructed:

*(&(objectClass=**)(objectClass=users))***

***(&(objectClass=foo)**(type=Epson*))*

*(&(objectClass=**)(objectClass=resources))***

***(&(objectClass=foo)**(type=Epson*))*

This set of code injections allows the attacker to infer the different objectClass values possible in the LDAP directory service. When the response web page contains at least one printer icon, the objectClass value exists (TRUE), on the other hand the objectClass value does not exist or there is no access to it, and so no icon, the objectclass value does not exist(FALSE). Blind LDAP injection techniques allow the attacker access to all information using TRUE/FALSE questions.

5.2. OR Blind LDAP Injection

In this case, the logic used to infer the desired information is the opposite, due to the presence of the OR logical operator. Following with the same example, the injection in an OR environment should be:

*(|(objectClass=**void)(objectClass=void))*

***(&(objectClass=void)**(type=Epson*))*

This LDAP query obtains no objects from the LDAP directory service, therefore the printer icon is not shown to the client (FALSE). If any icon is shown in the response web page then, it is a TRUE *response. Thus, an attacker could inject the following LDAP filters for gathering information:*

(|(objectClass=void)(objectClass=users))

(&(objectClass=void)(type=Epson))*

(|(objectClass=void)(objectClass=resources))
(&(objectClass=void)(type=Epson))*

5.3. Exploitation Example

In this section, an LDAP environment has been implemented to show the use of the injection techniques explained above and also to describe the possible effects of the exploitation of these vulnerabilities and the important impact of these attacks in current systems security. In this example the page printerstatus.php receives a parameter *idprinter* to construct the following LDAP search filter:

(&(idprinter=Value1)(objectclass=printer))

1) *Discovering Attributes:* Blind LDAP Injection techniques can be used to obtain sensitive information from the LDAP directory services by taking advantage of the AND operator at the beginning of the LDAP search filter built into the web application. For example, given the attributes defined for the printer object shown in Figure 12 and the response web page of this LDAP query in Figure 13 for *Value* 1=*HPLaserJet* 2100, an attribute discovering attack can be performed by making these following LDAP injections:

*(&(idprinter=**HPLaserJet2100)(ipaddress=***))*
(objectclass=printer))
*(&(idprinter=**HPLaserJet2100)(department=***))*
(objectclass=printer))
*(&(idprinter=**HPLaserJet2100)(department=***))*
(objectclass=printer))

Obviously, the attacker can infer from these results which attributes exist and which do not. In the first case, the information about the printer is not given by the application because the attribute ipaddress does not exist or it is not accessible (FALSE), as is shown in Figure 14.

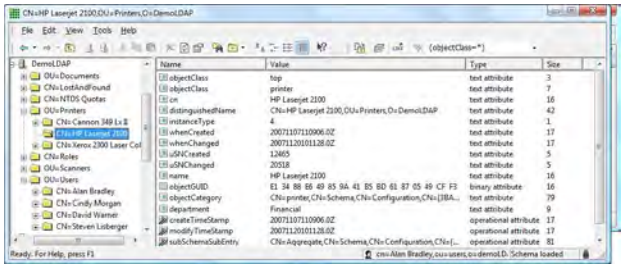


Figure 12. Attributes defined for the printer object.

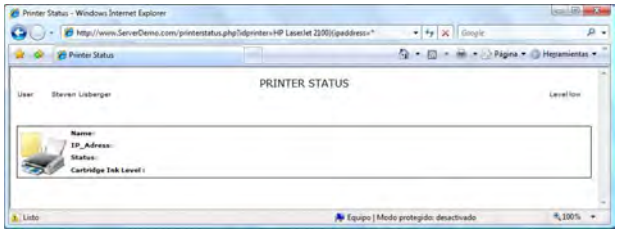


Figure 13. Normal behavior of the application.

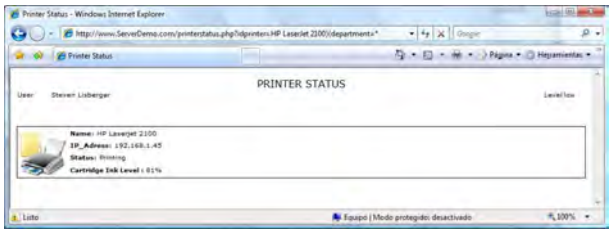


Figure 14. Response web page when the attribute does not exist.

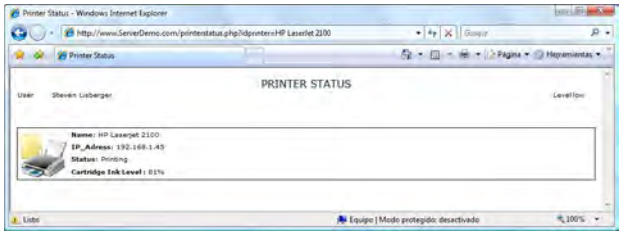


Figure 15. Response web page when the attribute exists.

On the other hand, in the second case, the response web page shows the printer status and therefore, the attribute department exists in the LDAP directory and it is possible access to it (Figure 15). Furthermore, with blind LDAP injection attacks the values of some of these attributes can be obtained. For example, suppose that the attacker wants to know the value of the department attribute: he can use booleanization and charset reduction techniques, explained in the next sections, to infer it.

2) *Booleanization:* An attacker can extract the value from attributes using alphabetic or numeric search. The crux of the idea is to transform a complex value (e.g. a string or a date) into a list of TRUE/FALSE questions. This mechanism, usually called booleanization, is summarized in Fgure 16 and can be applied in many different ways.

Suppose that the attacker wants to know the value of the department attribute. The process would be the following:

*(&(idprinter=**HPLaserJet2100)(department=a***))*
(objectclass=printer))
*(&(idprinter=**HPLaserJet2100)(department=f***))*
(objectclass=printer))

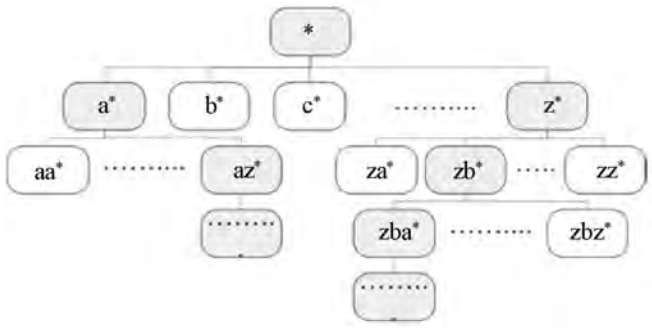


Figure 16. Booleanization.

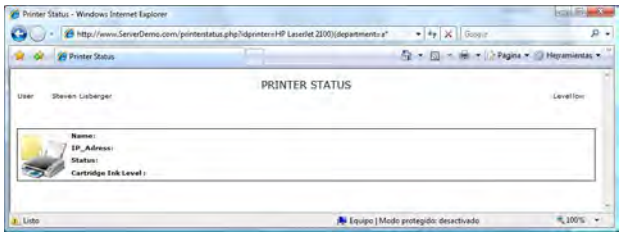


Figure 17. FALSE. Value does not start with 'a'.

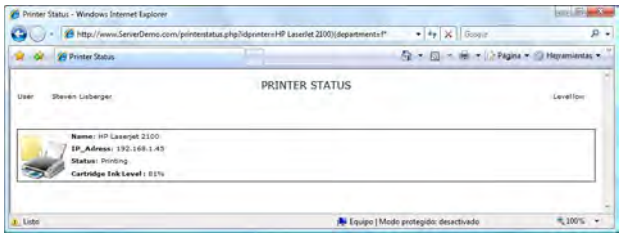


Figure 18. TRUE. Value starts with 'f'.

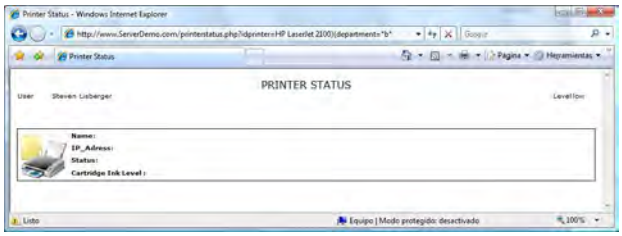


Figure 19. FALSE. Value doesn't start with 'fa'.

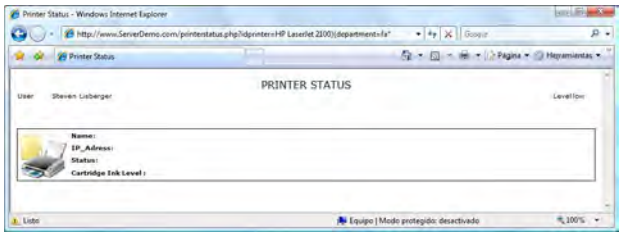


Figure 20. TRUE. Value starts with 'fi'.

*(&(idprinter=**HPLaserJet2100)(department=fa*)**)*
(objectclass=printer))
*(&(idprinter=**HPLaserJet2100)(department=fi*)**)*
(objectclass=printer))

As shown in Figure 12, the department value in this example is financial. The first try with the character "a" does not obtain any printer information (Figure 17) therefore, the first character is not an "a". After testing with the rest of the characters, the only one that obtains the normal behavior from the application is "f" (Figure18).

Regarding the second character, the only one that results in the normal operation of the application is 'i' (Figure 20) and so on. Following the process, the department value can be obtained. This algorithm can be also used for numeric values. In order to perform this, the booleanization process should use 'greater than or equal to' (≥) and 'less than or equal to' (≤) operators.

3) *Charset Reduction:* An attacker can use charset reduction to decrease the number of requests needed for obtain the information. In order to accomplish this, he uses wildcards to test if the given character is present *anywhere* in the value, e.g.:

*(&(idprinter=**HPLaserJet2100)(department=*n*)**)*
(objectclass=printer))

The Figure 21 shows the response web page when the character 'b' is tested: no results are sent from the LDAP directory service so no letter 'b' is present, but in Figure 22 a normal response web page is shown, meaning that the character 'n' is in the department value. Through this process, the set of characters comprising the department value can be obtained. Once the charset reduction is done, only the characters discovered will be used in the booleanization process, thereby decreasing the number of requests needed.

All these techniques can be easily performed with automated tools in order to extract all the information. Just as a proof of concept we developed LDAP Injector showed in Figure 23.

6. A Practical Proposal to Discover LDAP Vulnerabilities in Web Applications

In this section a practical proposal is described to recognize bugs in web applications vulnerable to LDAP injection attacks. This proposal is as general as needed to work with any LDAP directory the application might is using. It is based in black box techniques meaning no knowledge about the source code is needed. The core of this practical approach consists in try out different LDAP injections against every parameter and then to analyze the web application responses in order to recognize the vulnerability.

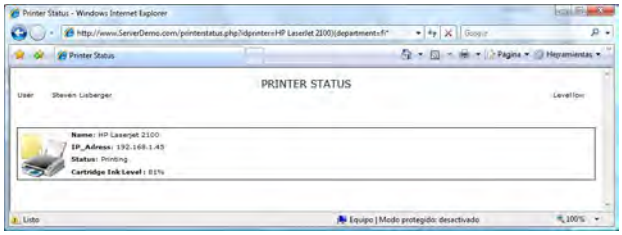


Figure 21. FALSE. Character 'b' is not in the department value.

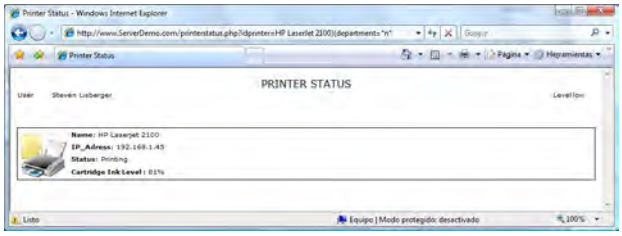


Figure 22. TRUE. Character 'n' is in the department value.

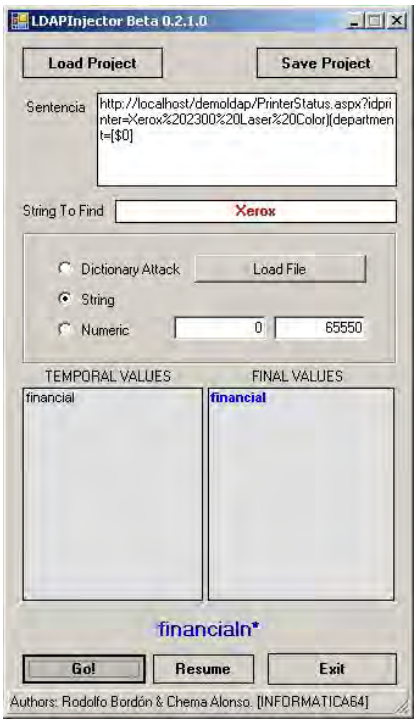


Figure 23. LDAP Injector performing a booleanization attack.

6.1. Definitions

Before start to describe the method some definitions are required to understand the basic principles in which relay on. These are the following:

Expected values. Set of characters forming the system's expecting input. These values generate a correct and normal result and behavior in the web application. These results are not an empty set of records. This means that after introduce an expected value web application retrieve any data from the LDAP directory.

Empty LDAP query (LDAP(Void)). LDAP query executed using expected values. This means no LDAP injection has been done.

Injection string (ILDAP). Set of characters not in the expected values. It is possible that the system is ready for any input character but it is assumed that LDAP special characters are those involved in LDAP Injection queries. Injections can be classified in:

- *Positive behavior change injection (ILDAP+).* It is an injection string to produce different number of retrieved records from the LDAP directory. It means the generated object list changes.
- *Negative behavior change injection (ILDAP-).* It is an injection string to produce fewer objects than the original one.

Zero behavior change injection (ILDAP0). It is an injection string to produce no change in the response object lists generated by the LDAP directory.

Injected LDAP query LDAP (ILDAP)). It is an LDAP query in which an injection string has been introduced. This injection should generate a syntax error or not. It this injection should not result in a syntax error then it is called Valid Injection (VI), otherwise it's called (Not Valid Injection). It is important to notice the use of *should* verb. This is because the injection should be correct in an injectable environment but security mechanisms in a web application could make it Not Valid. All NVI are also ILDAP-because no one object will be retrieved.

Minimum Valid Injection (MVI). It is an injection string which introduces no logic operators. It means the injections are constructed using the minimum number of parenthesis and operators without change the logic.

Complex Valid Injection (CVI). It is an injection string which introduces changes in the logic. The query should has a correct syntax and add new logic. Complex injections are necessaries to evaluate if the parameter is vulnerable to blind LDAP injections. In order to find out the correct syntax, the simplest Complex Valid Injection should be construct and this will only be possible if the web application is using and AND or an OR query, just as seen in the first part of this article. Table VI-A shows some examples.

Not Complex Valid Injection (NCVI). It is a correct injection with no syntax errors which injects new logic but changing the object list to retrieve none objects. It is a key to construct Boolean logic in Blind LDAP injection attacks. Table VI-A shows some examples.

Res(void). Object list retrieved after sending LDAP(void) to the web application. This is the result set sent from the LDAP engine to the web applications after the LDAP query is executed. Res(void) is constructed by the objects retrieved when no injection has been done and hence it is the normal result set.

Res(ILDAP). Object list retrieved after injecting and ILDAP. This result set obtained might has more or less objects than Resultset(void) depending on the ILAP. In each case will be known as Res(ILDAP+) or Res (ILDAP-). The results set obtained depend on several environmental aspects such as the normal query, the container in which is sent through, if it is recursive query, etc. If A is supposed to be injection string it will be an *ILDAP0 if RES(void)= RES(A),* it will be an *ILDAP+ if RES(void) < RES(A) and a ILDAP− if RES(void) > RES(A).*

HTMLRES (ILDAP). It will represent the response page obtained after sending the ILDAP to the web application. It is the data which methodology has to work with because is the info that web application sends back to the client as response to the test tried out. As it is working in a web environment this will be, normally, an HTML page.

6.2. Creating Valid LDAP Injections

Using as reference terminology defined in the previous section two rules can be settled up as:

1) If it is possible to construct a MVI for a parameter then it will be vulnerable against LDAP Injection attacks.

2) If it is possible to construct a CVI with AND/OR logic operators for a parameter then it will be vulnerable against Blind LDAP injection attacks.

As a general rule, in a black box pen testing audit, MVI should be constructed to test the parameter strength against LDAP Injection attacks. This is just because a Blind LDAP injection attack only can be conducted in parameters previously vulnerable against LDAP injection attacks. Let's suppose a web application retrieving a GET parameter as following:

http://www.myweb.com/prog.php?id=1.

It will be used to query an LDAP directory to obtain objects from a container matching filters as in this example:

(login_operator(atributte1=value1) (atributte2=value2)) or (atributte=value)

The query above will be known as LDAP(void) and the goal is to find out an MVI which guarantee no more records will be obtained.

As there is not a universal MVI which works in all the cases will be necessary to try out different ILDAPs. One ready for OR queries, another ready for AND ones and the last prepared to work in simple filters, it means with only one comparison and no one logic operator. In order to do this will be necessary to use as reference RES(void) supposing this is a normal behavior in the web application and that RES(void) is not null. This is mandatory in order to accomplish Res(void) > Res(ILDAP −).

Taking into consideration that:

- *Res(void) >=0 [not null].*
- *Res(void)= RES(LDAP 0).*
- *MV I are LDAP 0.*
- *Res(void) > RES (LDAP −).*
- *NCV I are LDAP −.*

Therefore is possible to conclude that if HTMLRES (void) = HTMLRES(MVI) and HTMLRES(VOID) != HTMLRES(NCVI) then the parameter is vulnerable against LDAP Injection attacks. The first condition proves LDAP directory is responding to LDAP injected queries correctly and second one proves which this is true, and not a web application behavior, by generating an empty object list and obtaining a different web application behavior.

It important to keep in mind that in blind environments, it means in web application in which data is never printed in the response web page or in the error messages, to extract all the data is necessary to find out not only a MVI which complaints the Vulnerable Rule but a CVI.

Vulnerable Rule against Blind LDAP Injection attacks: If HTMLRES (void) = HTMLRES(CVI) and HTMLRES (void)!= HTMLRES(NCVI) then the parameter is vulnerable against LDAP Injection attacks.

So, at the end, to find out if a parameter is vulnerable to LDAP Injection attacks or Blind LDAP Injection attacks, it is mandatory to recognize a response (HTMLRES) as a normal behavior or a response as a behavior when an LDAP or an empty object list has been retrieved. The first behavior will be referenced as a TRUE behavior and the other will be referenced as a FALSE behavior, allowing both to construct a binary logic.

6.3. Web Responses Analysis

Once a valid injection is constructed, it is necessary to analyze the response given by the web application in order to define the logic that is behind the booleanization. There are several behaviors that the system might has when it receives an injection. In fact these behaviors correspond to the treatment of errors implemented in the web server. The methodology has to deal with all the possibilities to be able to propose an effective criterion. This criterion determines if the response given by de the system for an CVI is a true response or a false one. The most important kinds of system responses when it faces an CVI are the following:

Table 1. Some examples of complex valid injections (Cvi).

Example	Original LDAP Query	injection String	Results
1 2 3	(attribute=value) (&(attribute1=value1)(attribute2=value2)) (|(attribute1=value1)(attribute2=value2))	Id=value)(Id=valu e1)(Id=value1)(|	(attribute=value)() (&(attribute1=value1)(&)(attribute2=value2)) (|(attribute1=value1)(|)(attribute2=value2))

Table 2. Some examples of not complex valid injections (Ncvi)).

	Original LDAP Query	injection String	Results
1 2 3	(attribute=value) (&(attribute1=value1)(attribute2=value2)) (|(attribute1=value1)(attribute2=value2))	Id=value** Id=value1)(| Id=value1)(&	(attribute=value**)() (&(attribute1= value1)(|)(attribute2=value2)) (|(attribute1= value1)(&)(attribute2=value2))

Web server error. These responses are predefined in the server configurations (p. e. http code 500).

Generic error. These responses are programmed by the application designer.

Correct results webpage. The response contains the expected values.

Last webpage displayed. The web application has implemented the errors treatment as a mechanism that proceed to send the last webpage displayed when an error occur.

For the first three alternatives it is easy to design a function to analyze the server response. Different techniques can be developed. For this work the following techniques have been evaluated:

HASH file signatures evaluation. Two different sets have to be define in order to classify which responses are false and which are true. This technique does not work with websites with dynamic content in its pages.

HTML tree evaluation. To deal with the problem exposed in the last point the focus of the evaluation is fixed on the tree structure of the HTML document not on the contents. This technique presents some limitations with websites where the error treatment maintains the same HTML structure that the normal documents.

Key words searching. This technique is oriented to define two distinguishing patterns: one for the false responses and another one for the true ones.

However, when error treatment mechanism uses the last webpage displayed to deal with a not expected input there is not any technique to define an effective error function at least for the time of being.

6.4. The Analysis of the Vulnerability of Web Application Parameter

In response to the descriptions given in sections above, it is possible to propose the steps that are necessary to determine the weakness of a parameter defined for a web application when it is faced an injection attack.

1) To find out the application's input parameters.

2) To try to construct an IMV.

3) If one IMV exists then

a) To try to construct at least one ICV

b) If this valid ICV exists with the AND or OR operators, then the parameter is vulnerable to Blind Injection attacks. At this point, it is necessary to determine the error treatment mechanism implemented in order to propose an efficient error function. If this mechanism is based on the last response given, today, the parameter can be considered as secure.

c) If is not possible construct a valid ICV the parameter can be consider as secure.

4) If no IMV exits then the parameter can be consider as secure.

7. Securing Applications against Blind LDAP Injection & LDAP Injection Attacks

The attacks presented in the previous sections are performed on the application layer, therefore firewalls and intrusion detection mechanisms on the network layer have no effect on preventing any of these LDAP injections. However, general security recommendations for LDAP directory services can mitigate these vulnerabilities or minimize their impact by applying minimum exposure point and minimum privileges principles.

Mechanisms used to prevent code injection techniques include defensive programming, sophisticated input validation, dynamic checks and static source code analysis. The work on mitigating LDAP injections must involve similar techniques.

It has been demonstrated in the previous sections that LDAP injection attacks are performed by including special characters in the parameters sent from the client to the server. It is clear therefore that it is very important to check and sanitize the variables used to construct the LDAP filters before sending the queries to the server.

However, developer communities are not widely aware of this kind of injections because there is no so much information about LDAP Injection and Blind LDAP Injection techniques, hence developers don't sanitize correctly their queries against LDAP directories. A quick search for "LDAP" in websites hosting open source projects retrieves a lot of projects with LDAP Injection vulnerabilities. On the other hand, static code analysis tools are not ready yet to discover LDAP injection vulnerabilities in the code. So it is easy, for a developer not strongly formed in security best practices, to create a vulnerable code just relaying in security post-analysis. Microsoft Code Analysis, a tool forming part of Microsoft Visual Studio Team System or Microsoft FXCop, two of the most used code analysis tools don't have any rule to detect LDAP injection vulnerabilities.

In order to sanitize correctly web application inputs which are going to be used in LDAP search filters, developers must only pay attention to ten special characters: |, &, (,), *, <, >, =, ~, !. If the developer sanitizes in a secure way the input to forbid those characters LDAP Injection attacks won't work.

8. Conclusions and Future Work

LDAP services facilitate access to networks information organizing it in a hierarchical database that allows authorized users and applications to find information related to people, resources and applications.

This protocol is simple to install, maintain, replicate and use, and it can be highly distributed. And it allows an easy implementation of the widely used single sign-on

environments. Therefore, given the increasing need for information in current systems, it is an essential service in almost all networks.

LDAP injection techniques are an important threat for these environments, specially, for the control access and privileges and resources management.

These attacks modify the correct LDAP queries, altering their behavior for the attacker benefit. And the consequences of these attacks can be very severe.

Our work is unique in providing a rigorous analysis of LDAP injection techniques and in showing representative examples of the possible effects of these attacks.

Even more, recommendations to secure applications against these techniques have been proposed. It has been showed that filtering the error messages produced by the server only fortifies the system but does not secure it against blind injection techniques. A more in depth protection is needed to avoid this kind of injection vulnerabilities too. It has been demonstrated with the presented examples, that it is essential to filter the client inputs used to construct the LDAP queries before sending them to the server. And that the AND and OR filter constructions should be avoided.

Finally, a very interesting line for future research is working on analyzing injection techniques with other protocols used to access databases and directories. And to study the possible utilization of mechanisms booleanization techniques such as character displaying or charset reduction in other environments.

9. References

[1] S. Barnum and G. McGraw, "Knowledge for software security," IEEE Security and Privacy Magazine, Vol. 3, No. 2, pp. 74–78, 2005.

[2] E. Bertino, A. Kamra, and J. Early, "Profiling database application to detect SQL injection attacks," in Proceedings of the IEEE International Performance, Computing, and Communications Conference, pp. 449–458. 2007.

[3] X. Fug, X. Lu, B. Peltsverger, S. Chen, K. Qian, and L. Tao, "A static analysis framework for detecting SQL injection vulnerabilities," in Proceedings of the 31st Annual International Computer Software and Applications Conference, pp. 87–96, 2007.

[4] E. Merlo, D. Letarte, and G. Antoniol, "SQL-injection security evolution analysis in PHP," in Proceedings of the 9th IEEE International Workshop on Web Site Evolution, pp. 45–49, 2007.

[5] S. Thomas and L. Williams, "Using automated fix generation to secure SQL statements," in Proceedings of the 3rd International Workshop on Software Engineering for Secure Systems, pp. 9–19, 2007.

[6] "XPath 1.0 specification," 1999, http://www.w3.org/TR/xpath.

[7] "XPath 2.0 specification," 2007, http://www.w3.org/TR/xpath20/.

[8] "RFC 1777: Lightweight Directory Access Protocol v2," 1995, http://www.faqs.org/rfcs/rfc1777.html.

[9] "RFC 2251: Lightweight Directory Access Protocol v3," 1997, http://www.faqs.org/rfcs/rfc2251.html.

[10] T. Holz, S. Marechal, and F. Raynal, "New threats and attacks on the world wide web," IEEE Security and Privacy Magazine, Vol. 4, No. 2, 2006.

[11] G. Hermosillo, R. Gomez, L. Seinturier, and L. Duchien, "AProSec: An aspect for programming secure web applications," in Proceedings of the Second International Conference on Availability, Reliability and Security, pp. 1026–1033, 2007.

[12] N. Jovanovic, C. Kruegel, and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," in Proceedings of the IEEE Symposium on Security and Privacy, pp. 6–15, 2006.

[13] E. Jamhour, "Distributed security management using LDAP directories," in Proceedings of the XXI Internatinal Conference of the Chilean Computer Science Society, pp. 144–153, 2001

[14] R. Sari and S. Hidayat, "Integrating web server applications with LDAP authentication: Case study on human resources information system of ui," in Proceedings of the International Symposium on Communications and Information Technologies, pp. 307–312, 2006.

[15] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)," 1997, http://www.ietf.org/rfc/rfc2251.

[16] V. Koutsonikola and A. Vakali, "LDAP: Framework, practices, and trends," IEEE Internet Computing, Vol. 8, No. 5, pp. 66–72, 2004.

[17] M. Russinovich and D. Solomon, Microsoft Windows Internals, Microsoft Press, 2004.

[18] "OpenLDAP main page," http://www.openldap.org.

Wireless Sensor Network, **2009, 1, 245-256**
doi:10.4236/wsn.2009. 14031 Published Online November 2009 (http://www.scirp.org/journal/wsn).

On the Implementation of a Probabilistic Equalizer for Low-Cost Impulse Radio UWB in High Data Rate Transmission

Sami MEKKI[1], Jean-Luc DANGER[1], Benoit MISCOPEIN[2]
[1]*Institut Telecom/Telecom Paris Tech (ENST), Paris, France*
[2]*France Télécom R & D, Meylan Cédex, France*
Email: {*mekki, danger*}*@enst.fr, benoit.miscopein@orange-ftgroup.com*
Received April 11, 2009; *revised June* 9, 2009; *accepted June* 10, 2009

Abstract

This paper treats the digital design of a probabilistic energy equalizer for impulse radio (IR) UWB receiver in high data rate (100 *Mbps*). The aim of this study is to bypass certain complex mathematical function as a *chi-squared* distribution and reduce the computational complexity of the equalizer for a low cost hardware implementation. As in Sub-MAP algorithm, the $\max^*$ operation is investigated for complexity reduction and tested by computer simulation with fixed point data types under 802.15.3a channel models. The obtained results prove that the complexity reduction involves a very slight algorithm deterioration and still meet the low-cost constraint of the implementation.

Keywords: Impulse Radio Ultra-Wideband, Probabilistic Energy Equalizer, Inter-Symbol Interference, Chi-Squared

1. Introduction

Ultra-wideband impulse radio is considered as a promising candidate for indoor communications and wireless sensor networks, as described in [1]. Despite the numerous advantages afforded by the ultra-wideband (UWB) [1], this system faces the technological limits which brake the development of impulse radio (IR) UWB. Coherent IR-UWB reception, based on Rake receiver is limited in number of implementable Rake fingers [2]. An alternative is given by the transmitter reference (TR) method [3], however the electronic architecture is more complex as it needs analog delay lines and mixers. Non-coherent energy detection receiver is far less complex as a few components like shottky diodes and capacitors suffice. Though, the energy detection is simple to implement, transmitting impulses at high data rate leads to inter-symbol interference (ISI) which decreases the performance of the receiver [4–6]. An efficient scheme is necessary to improve the system performance.

A probabilistic energy equalizer is proposed in [7], which handles different types of interference. Besides the ISI, the proposed equalizer could manage the intra-symbol interference, called also inter-slot interference (IStI) in an $M-array$ pulse position modulation. Nevertheless, equalization process is mathematically complex to implement. The problem is mainly located on the energy distribution which follows a *chi-squared* distribution [8] and on the number of multiplications required by the equalizer.

In this paper, the probabilistic equalizer defined in [7] is simplified by applying the Jacobi logarithm [9] where addition become $\max^*$ operation (using Viterbi's notation [10]) and multiplications become additions. In order to make this possible, an approximation of the chi-squared distribution is considered and rewritten in the logarithmic domain as the probabilistic equalizer. The simplified equalizer is embedded into the iterative loop of a channel decoder which applies the Sub-MAP algorithm in the decoding process.

This article is organized as follows: Section 2 defines the system model under consideration, where energy distribution is established. Equalization principle is reviewed in Section 3. In Section 4, the energy distribution is approximated by a simple function for hardware implementation. Results with the approximated distribution are compared to the *chi-squared* distribution in Section 5. In the same Section, the hardware implementation results in fixed point precision data types are also depicted and compared to the theoretical results in floating point precision. In Section 6 we rewrite the probabilistic equalizer in the logarithmic domain to base 10 with respect to $\max_{10}^*$ operation and to the approximated distribution. In Section 7, the complexity

and the performance of the logarithmic equalizer is studied and compared to the complexity of a linear equalizer. Finally, conclusion and forthcoming work in the field are given in Section 8.

2. System Design{TC "1 Transmitter and Receiver Design."\f f}

We consider an IR-UWB receiver based on energy detection. Data transmission is ensured via the M-array pulse position modulation (M-PPM) over a bandwidth W. Transmitting pulses over a high dispersive channel causes inter-symbol interference (ISI) and intra-symbol interference denoted as inter-slot interference (IStI). The received signal over a time symbol T_s has the following expression

$$y_n(t) = \sum_{k=0}^{\infty} x_{n-k}(t) + z_n(t) \tag{1}$$

where $z_n(t)$ is an additive white Gaussian noise with variance σ^2 and mean zero, and $x_{n-k}(t)$ is the channel response of the $(n-k)^{th}$ transmitted symbol defined by:

$$x_{n-k}(t) = p(t - A_{n-k}T_{slot}) \otimes h(t) \tag{2}$$

where $h(t)$ is the impulse channel response, $\otimes$ denotes the convolution product, $p(t)$ is the pulse shape, T_{slot} is the time slot duration for an M-PPM modulation,*i.e.* $T_s = MT_{slot}$, and $A_{n,k}$ takes value in $\{0,1,\ldots,M-1\}$ according to transmitted symbol.

Let K denotes the number of interfering symbol assumed by the receiver, even though the real number of interfering symbol is greater. Thus for digital treatment the received signal (1) becomes a finite sum defined as:

$$y_n(t) = \sum_{k=0}^{K-1} x_{n-k}(t) + z_n(t) \tag{3}$$

The received energy per time slot T_{slot} in the n^{th} received symbol is given by

$$\mathrm{E}_{n,m} = \int_{nT_s+(m-1)T_{slot}}^{nT_s+(m)T_{slot}} \left(s_n(t) + z_n(t)\right)^2 dt \tag{4}$$

where $s_n(t) = \sum_{k=0}^{K-1} x_{n-k}(t)$.

Following the approach of Urkowitz [11], it was shown that the energy of a signal of duration T_{slots} can be represented as a sum of $2T_{slot}W$ samples in number which is know as the degrees of freedom (DoF). Let $2L$ stands for the DoF during a time slot T_{slot}. Thus, the energy in the m^{th} slot of n^{th} symbol is given by

$$\mathrm{E}_{n,m} = \sum_{\ell=1}^{2L} (s_{n,m}^{\ell} + z_{n,m}^{\ell})^2 \tag{5}$$

where $s_{n,m}^{\ell}$ and $z_{n,m}^{\ell}$ are respectively the ℓ^{th} sample of $s_n(t)$ and $z_n(t)$ in m^{th} slot of n^{th} symbol.

Assuming $\sum_{\ell=1}^{2L} (s_{n,m}^{\ell})^2 \neq 0$, then the received energy follows a non-central *chi-squared* distribution

$$p(\mathrm{E}_{n,m} \mid B_{n,m}) = \frac{1}{2\sigma^2}\left(\frac{\mathrm{E}_{n,m}}{B_{n,m}}\right)^{\frac{L-1}{2}} e^{-\frac{(\mathrm{E}_{n,m}+B_{n,m})}{2\sigma^2}} I_{L-1}\left(\frac{\sqrt{B_{n,m}\mathrm{E}_{n,m}}}{\sigma^2}\right) \tag{6}$$

with $2L$ DoF and noncentrality parameter defined as $B_{n,m} \triangleq \sum_{\ell=1}^{2L} (s_{n,m}^{\ell})^2$. The function $I_{L-1}(u)$ is the $(L-1)^{th}$-order modified Bessel function of the first kind [8]. If the noncentrality parameter is equal to zero; *i.e.* $B_{n,m} = 0$; the received energy follows a central *chi-squared* distribution defined as

$$p(\mathrm{E}_{n,m} \mid 0) = \frac{1}{\sigma^{2L} 2^L \Gamma(L)} \mathrm{E}_{n,m}^{L-1} e^{\frac{-\mathrm{E}_{n,m}}{2\sigma^2}} \tag{7}$$

where $\Gamma(z)$ is the gamma function [8].

The energy distribution is studied in next sections and simplified for hardware implementation.

3. Energy Equalization Principle

To benefit from the iterative process of a communication system, we consider a probabilistic equalizer that can be embedded into the iterative loop of a channel decoder based on SISO (Soft-Input/Soft-Output) decoding.

Thus, the considered equalizer takes the accumulated energy per slot (i.e. $\mathrm{E}_{n,m}$) and per symbol (i.e. $E_n = (\mathrm{E}_{n,1}, \mathrm{E}_{n,2}, \ldots, \mathrm{E}_{n,M})$) as reference, in order to retrieve the transmitted symbol x_n. So the detector computes a conditioned probability $p(E_n \mid x_n)$ regarding the interfering symbols on x_n. It has been shown in [7] that the equalization is performed by computing

$$p(E_n \mid x_n) = \sum_{x_{n-1}} \cdots \sum_{x_{n-K+1}} \left(\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m}) \prod_{k=1}^{K-1} \pi(x_{n-k}) \right) \tag{8}$$

where $\pi(x_{n-k})$ is the *a priori* probability provided by the SISO decoder and $p(\mathrm{E}_{n,m} \mid B_{n,m})$ is defined in Section 2. It was also established that the set of all the

possible values that $B_{n,m}$ could take, has a finite cardinal. Figure 1 summarize the transmission and the receiver design under consideration.

In order to reduce the complexity and make the equalizer feasible, we investigate the implementation in finite precision.

Moreover the probability given by Equation (8) needs some mathematical simplifications and approximations of the probability density function (pdf) $p(\mathrm{E}_{n,m} \mid B_{n,m})$, corresponding either to the *central* (7) or *non-central chi-squared* (6) distribution. This will be investigated in the following section.

4. Chi-Squared Distribution Approximation for Hardware Implementation

The chi-squared distribution defined by (7) and (6) is a three variable function ($\mathrm{E}_{n,m}$, $B_{n,m}$ and σ^2). Thus, building a look-up table according to these parameters would occupy a great memory. For instance, if the energy distribution is coded in 7 bits and $\mathrm{E}_{n,m}$, $B_{n,m}$ and σ^2 are coded respectively in 14-bit, 6-bit and 6-bit long, the space memory allocated to this look-up table would occupy 448 Mbits (or 56 Mbytes). This corresponds to a costly silicon area in a FPGA or ASIC technology and thus incompatible with low-cost constraints.

An approximation for the *chi-squared* distribution is thus necessary. In the literature, there are some proposals for the calculation of the non-central *chi-squared* distribution [12] and the use of the normal approximation to the *chi-squared* distribution [13,14], but those approximations require high bit precision and are therefore too complex for digital design.

An intuitive approximation can be found by considering the *Remark* in [15] which stands that when a variable Σ is used to approximate a variable Ω, it is equivalent to match the mean and variance of Σ and Ω.

It is notably shown in [15], that a *chi-squared* distribution can be approximated by a Gaussian distribution. However the smaller the number of DoF $2L$, the larger the approximation error. Due to the large bandwidth W in UWB-IR, the number DoF could be big enough [16] to consider the Gaussian distribution as an approximation to the *chi-squared* density. For instance $2L$ is around 30 for $W = 3GHz$ and $T_{slot} = 5ns$. According to the previous *Remark*, the Gaussian approximation has the same mean and variance as the *non-central chi-squared* distribution, *i.e.* $\mathrm{E}_{n,m} : N(m_{\chi^2}, \sigma^2_{\chi^2})$, given by [17]:

$$m_{\chi^2} = 2L\sigma^2 + B_{n,m} \tag{9}$$

$$\sigma^2_{\chi^2} = 4L\sigma^4 + 4\sigma^2 B_{n,m} \tag{10}$$

This can be extended to the *central chi-squared* distribution by considering $B_{n,m} = 0$.

Using these results and the aforementioned assumptions, we obtain the approximation for the energy distribution (noticed $\tilde{p}$) per slot, $\forall B_{n,m} \geq 0$ and $2L >> 2$ as

$$p(\mathrm{E}_{n,m} \mid B_{n,m}) \approx \tilde{p}(\mathrm{E}_{n,m} \mid B_{n,m}) = \frac{\exp\left(-\dfrac{(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2\sigma^2_{\chi^2}}\right)}{\sqrt{2\pi\sigma^2_{\chi^2}}} \tag{11}$$

Figure 2(a) shows the error measured by $|\, p(\mathrm{E}_{n,m} \mid B_{n,m}) - \tilde{p}(\mathrm{E}_{n,m} \mid B_{n,m})\,|$ for $\mathrm{E}_{n,m} \geq 0$, $B_{n,m} > 0$ and $\sigma^2 = 1$.

Table 1. Look-up table Input/Output size with x^2 distribution.

Parameters	Quantization size
$\mathrm{E}_{n,m}$	14 bits
$B_{n,m}$	6 bits
σ^2	6 bits
$p(\mathrm{E}_{n,m}\|B_{n,m})$	7 bits
x^2 Table size	448 Mbits (56 Mbytes)

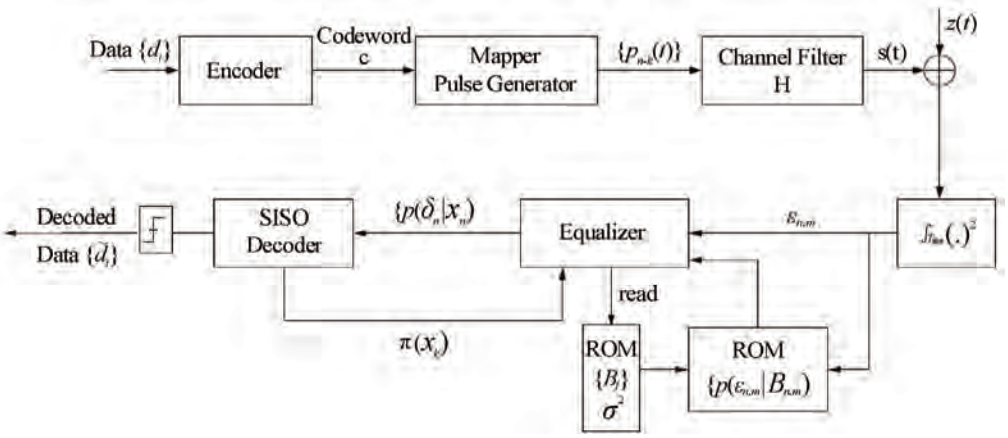
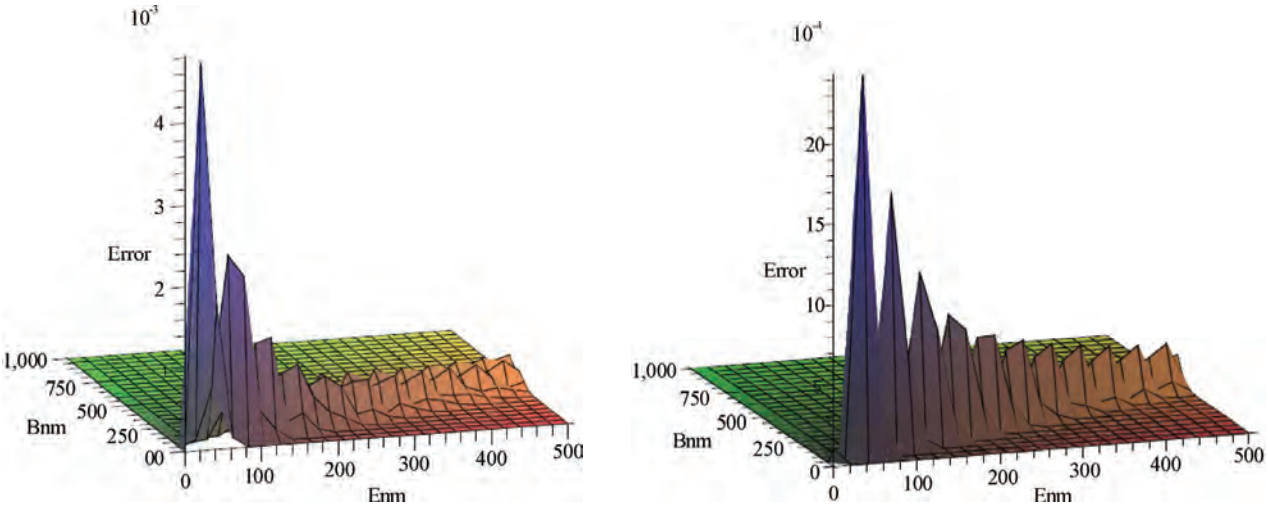


Figure 1. Transmitter and receiver design.



(a) Error for $\sigma^2 = 1$ **(b) Error for** $\sigma^2 = 0.5$

Figure 2. Error measured by $|p(E_{n,m} | B_{n,m}) - \hat{p}(E_{n,m} | B_{n,m})|$ **for** $\forall E_{n,m} \geq 0$, $B_{n,m} > 0$.

It is noticed that the error tends to zero as σ^2 decreases (Figure 2 (b)). According to [7], the energy equalizer operates at $\sigma^2 < 1$; i.e. $\sigma^2 = 1$ corresponds to $SNR = -3dB$ for a pulse energy equals to unity in coded system. So, the maximum error, considered between the *chi-squared* and Gaussian distributions, is $\varepsilon = 5 \times 10^{-3}$ as shown in Figure 2(a). We denote the normal function by

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \tag{12}$$

Using (9), (10) and (12), equation (11) can be rewritten as follows

$$\hat{p}(E_{n,m} | B_{n,m}) = \frac{1}{\sqrt{\sigma^2_{\chi^2}}} \phi\left(\frac{E_{n,m} - m_{\chi^2}}{\sqrt{\sigma^2_{\chi^2}}} \right) \tag{13}$$

As the energy distribution is simply deduced from the normal function $\phi(t)$, the digital implementation can only use two look-up tables. The first one contains the values of the normal function $\phi(t), \forall t \geq 0$. The second one contains the values of the ratio $1/\sqrt{x}, \forall x > 0$. The input/output precision of the look-up tables will be analyzed in the simulation Section according to the hardware constraints.

5. Performance of the Approximated Linear Equalizer

In this section, computer simulations have been run to assess the performance of the linear energy equalizer with the approximated Gaussien distribution defined by (13). The BER computation has been performed via simulations in both floating point precision and fixed point precision data types. In the firsts part of simulations, we compare the performance of the receiver with the Gaussian approximation (11) and with the exact calculation of the *chi-squared* distribution in floating point precision. Second part of simulations has been run in fixed point data types with the approximated distribution.

The block fading multipath channel is generated randomly according to IEEE 802.15.3a UWB channel models [18]. Channel estimation is out of the scope of this paper. The channel state information (CSI) is assumed perfectly known at the receiver side. Nevertheless, channel parameters can be approached by the mean of the expectation-maximization (EM) algorithm as studied in [19] or by a set of a specific training sequence.

5.1. Chi-Squared Versus Gaussian Approximation Simulations in Double Precision

We consider an UWB-IR system as defined in Figure 1. Transmission is ensured by a 4-PPM modulation at $100 Mbit/s$. Thus we get 2 bits per transmitted symbol. We have implemented a duo-binary turbo code as it is defined in the standards [20,21]. This channel coder is chosen because it is suited to QPSK (quadratic phase shift keying) and 4-PPM modulations. The encoded data, at the input of the encoder, are 864-bit long blocks. The turbo encoder rate is 1/2 and 10 iterations of the SISO decoder are performed at the receiver side. The equalizer is jointly implemented into the iterative loop of the decoder to benefit from the iterative process of the decoder. The efficiency of the energy equalizer will not

be treated in this paper, the reader should refer to [7] for more details concerning the equalizer performances.

The receiver assumes that there are only two interfering symbols, *i.e.* $K = 2$ and $P = 5$, but the real number of interfering symbols could be more. The CSI is assumed over P time slots duration and not otherwise. In our case, for a data rate of 100 Mbps, the time slot duration is $5ns$, so the receiver has a perfect CSI only over $25ns$. This duration is sufficient for channel models as CM1 and CM2, although their respective maximum excess delay are $80ns$ and $115ns$ as it is studied in [7]. However for highly dispersive channel as CM3 and CM4 with maximum excess delay of $140ns$ and $200ns$ respectively, channel knowledge should be extended to $K = 3$. Nevertheless, we consider only simulations with $K = 2$ for the Gaussian approximation performance comparison.

It is noticed that the results with Gaussian approximation match the *chi-squared* performances in floating point precision even for highly dispersive channel such CM3 and CM4 with a slight degradation of performance.

5.2. Fixed Point Precision Simulations

The fixed point precision is subject to hardware constraints. The duo-binary turbo coder hardware implementation is out of the scope of this paper. The digital design of the channel coder is furnished by *Turbo-Concept* for an optimum efficiency [22]. The energy detector of UWB platform is a logarithmic one [23]. To guarantee the scalar value of the energy $E_{n,m}$ for equalization, a look-up table of the function 10^x is required. Computer simulations in fixed point precision are achieved by means of the SystemC class *sc_fix* [24]. The Gaussian approximation for energy equalization are computed through the look-up tables of the functions: $\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$, $g(x) = 1/\sqrt{x}$ and $h(x) = 10^x$. Figure 4 shows the Gaussian approximation computation architecture for the chi-squared distribution.

According to the class *sc_fix* of SystemC, a signed or an unsigned object are defined by two parameters: the total word length noted as wl, *i.e.* the total number of bits used in the type, and the integer word length noted as iwl, *i.e.* the number of bits that are on the left of the binary point (.) in a fixed point number. The remaining bits stand for the fractional part of the object. Hence each object is represented by a pair of parameters noted $< wl, iwl >$.

Simulations have been carried out with different parameter sizes. Table 2 shows the word sizes of the parameters considered for the fixed point simulations. Reducing the number of bits for each variable involves a significant performance decrease.

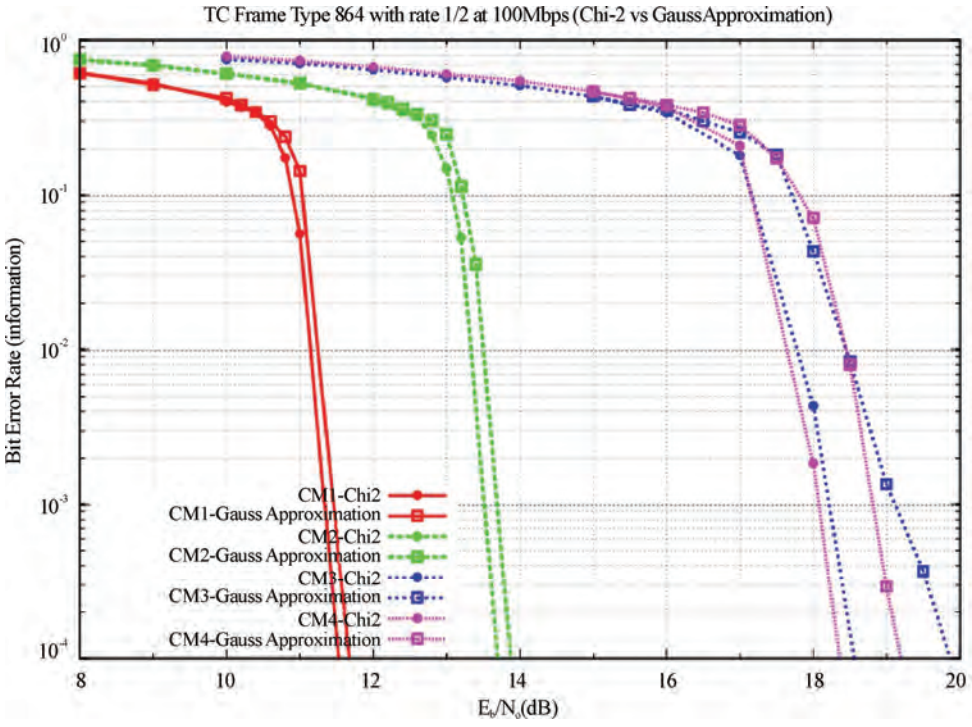


Figure 3. ***Chi-squared*** **vs gaussian approximation in float precision using duo-binary turbo code at rate 1/2 with** ***K*****=2.**

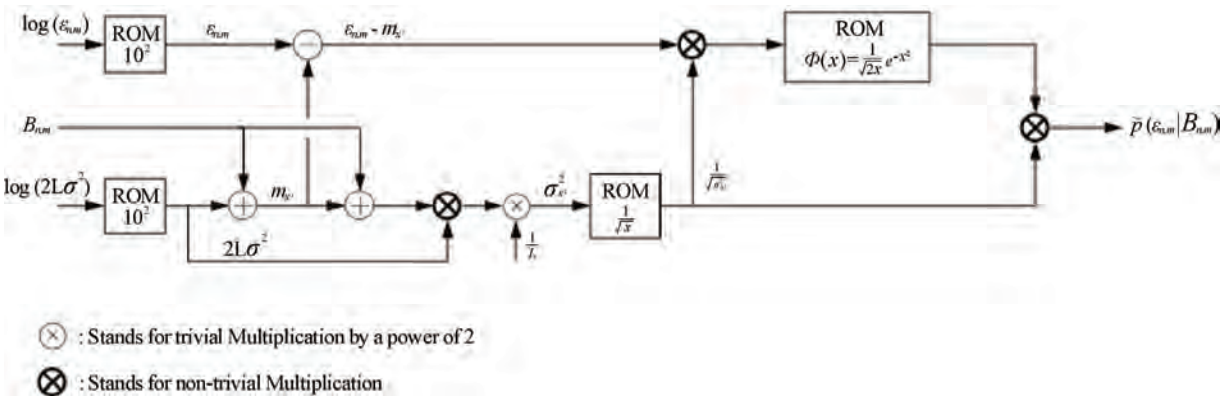


Figure 4. Approximated energy distribution architecture for the linear equalizer with x^2 approximation.

Table 3 lists the Input/Output size look–up table necessary for density computation.

Table 2. Parameters size definition.

Parameters	Quantization size $< wl, iwl >$
$\log E_{n,m}$	< 6,2 >
$E_{n,m}$	< 14,1 >
$B_{n,m}$	< 6,2 >
σ^2	< 6,1 >
m_{χ^2}	< 12,2 >
$\sigma^2_{\chi^2}$	< 12,2 >
$p(E_{n,m} \mid B_{n,m})$	< 13,3 >
$p(E_n \mid x_n)$	< 13,6 >
$\pi(x_k)$	< 4,1 >

We notice that the total memory occupied by the look-up tables is lower than the *chi-squared* look-up table as it is described in Table 1.

Simulations according to Table 2 and 3 under the same conditions as for double precision lead to the results depicted in Figure 5.

Table 3. Look-up table input/output size.

Parameters	Input size	Output size	Table size (Kbits)
$\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$	< 8,2 >	< 18,0 >	4.5
$g(x) = 1/\sqrt{x}$	< 12,2 >	< 6,4 >	24
$h(x) = 10^x$	< 6,2 >	< 14,1 >	0.875

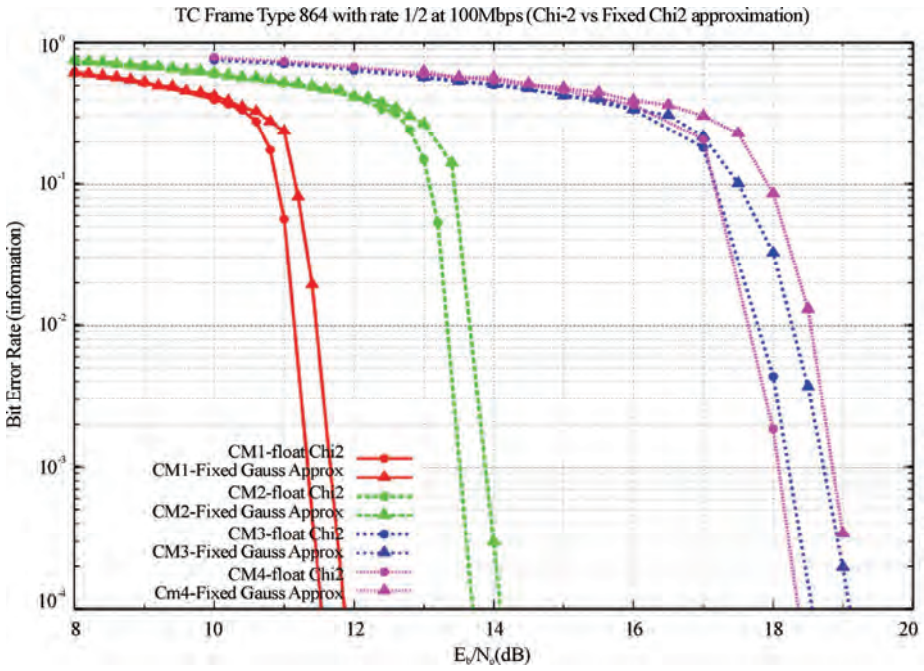


Figure 5. *chi-squared* float precision versus the Gaussian approximation in fixed point precision for K=2.

Results in fixed point precision data types are close to those obtained in double precision with *chi-squared* distribution. We notice, that even the quantization error of the energy distribution is around $1/2^{13}$, which is lower than the considered maximum error $\varepsilon = 5\times10^{-3}$, the receiver performances are slightly degraded compared to double precision simulations.

5.3. Complexity of the Linear Equalizer

The linear equalizer with a chi-squared distribution has an expensive lookup table (Table 1). Due to the size of the ROM and its cost, the *chi-squared* distribution is approximated by a simple implementable function (13) with some memories.

According to Figure 4 and equation (8), the amount of non trivial multiplications in the linear domain is about $(4M+K-2)M^K$ in a symbol period. We denote by trivial multiplication the multiplication by a power of 2 which is equivalent to a shift in hardware implementation. The detail of multiplications is as follows[1]

$$\overset{Mtimes}{\downarrow} p(E_n \mid x_n) = \underbrace{\sum_{x_{n-1}} \cdots \sum_{x_{n-K+1}}}_{M^{K-1}\,cases} \left(\underbrace{\underbrace{\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m})}_{(M-1)\,Mul} \underbrace{\times}_{1\,Mul} \underbrace{\prod_{k=1}^{K-1} \pi(x_{n-k})}_{(K-2)\,Mul}}_{(M+K-2)\,Mul} \right) \tag{14}$$

According to the decomposition in (14), $p(E_n \mid x_n)$ requires $(M+K-2)M^{K-1}$ multiplication. We should notice that $p(E_n \mid x_n)$ is computed M times per symbol duration. Thus, in a symbol period, the total amount of multiplication is equal to $(M+K-2)M^K$ multiplications. Once the number of multiplications is established for (14) per symbol period, we divide up each terms. The *a priori probability*, i.e. $\pi(x_{n-k})$, will not be discussed since it is provided by the SISO decoder. So our analysis is focused rather on the energy distribution. In the term $\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m})$, we calculate M times $p(\mathrm{E}_{n,m} \mid B_{n,m})$. With respect to the architecture in Figure 4, $p(\mathrm{E}_{n,m} \mid B_{n,m})$ requires 3 non-trivial multiplication. Hence, per symbol period there is $3M^{K+1}$ additional multiplications on the back of $(M+K-2)M^K$ This leads to $(4M+K-2)M^K$ multiplications calculated by the equalizer per time symbol.

As example, we consider a 4-PPM modulation at $100Mbps$ and $K = 2$ at the receiver side, this leads to $12.8Gmultiplication/s$ and a total memory of $30.25Kbits$ according to Table 3 for the energy distribution computing, *i.e.* $p(\mathrm{E}_{n,m} \mid B_{n,m})$. We should notice that the equalizer computes the energy distribution M^{K+1} times per time symbol. So, at $100Mbps$ for $K = 2$ with a 4-PPM the frequency of table access is about $3.2GHz$. If we consider a hardware that runs at $400MHz$, the level of parallelism to achieve the energy distribution is equal to 8. Thus, the total amount of memory is a factor of 8, *i.e.* $30.25Kbits \times 8 = 242Kbits$.

The next par of this paper will be focused on complexity reduction of the probabilistic equalizer by the mean of Sub-MAP algorithm known also as Jacobi algorithm [9].

6. Complexity Reduction of the Probabilistic Energy Equalizer

Even though the defined equalizer is implementable in hardware devices, the expensive area required by the equalizer could be decreased by simple computational method. This is made possible by computing in the logarithmic domain where only additions and comparisons operations with small memories are required. The channel decoder should operate also in the logarithmic domain, in order to get the better performance of the receiver. The decoder properties are not discussed in this paper. However, Sub-MAP algorithm, also called Max-Log-MAP or Dual Viterbi [21,10] based decoder is a good candidate for joint decoder equalizer receiver in the logarithmic domain.

As in the Sub-MAP decoding [10,25], we consider the max^*_{10} function which operates in a logarithm to base 10 defined as

$$\begin{aligned} max^*_{10}(a,b) &\overset{\Delta}{=} \log(10^a + 10^b) \\ &\overset{\Delta}{=} \max(a,b) + \log(1+10^{-|a-b|}) \end{aligned} \tag{15}$$

the max^*_{10} operation is essentially a max operation adjusted by a correction factor carried out by a lookup table; *i.e* a read-only memory (ROM); which outputs the correction term $\log(1+10^{-|a-b|})$ given the input $(a-b)$ in hardware implementation. As the max property, max^*_{10} is an associative operator (see APPENDIX 1 for the proof):

$$max^*_{10}(a,b,c) = max^*_{10}[max^*_{10}(a,b),c] \tag{16}$$

For notation simplicity we consider

$$max^*_{10}(a_1,a_2,\ldots,a_N) = max^*_{10 i\in\{1,\ldots,N\}}(a_i) \tag{17}$$

Using the max^*_{10}operation defined in (15) , the output of the probabilistic equalizer in the logarithmic domain is

[1]Mul stands for Multiplication in (14).

given by:

$$\log p(E_n \mid x_n) = max^*_{10 x_{n-1},\ldots,x_{n-K+1}} \left(\sum_{m=1}^{M} \log p(\mathrm{E}_{n,m} \mid B_{n,m}) + \sum_{k=1}^{K-1} \log \pi(x_{n-k}) \right) \quad (18)$$

Considering the result of equation (18), it is noticed that the multiplication operations are replaced by comparators and adders which are costless and easy to implement. Since the equalizer output should be a probability, i.e. $p(E_n \mid x_n) \in [0,1]$, a normalization process is applied as follow:

$$\overline{p}(E_n \mid x_n) = \frac{p(E_n \mid x_n)}{\sum_{x_n} p(E_n \mid x_n)} \quad (19)$$

where $\overline{p}$ is the normalized probability at the output of the equalizer. In the logarithmic domain normalization becomes

$$\log \overline{p}(E_n \mid x_n) = \log p(E_n \mid x_n) - \log \left(\sum_{x_n} p(E_n \mid x_n) \right) \quad (20)$$

$$= \log p(E_n \mid x_n) - max^*_{10 x_n} \log p(E_n \mid x_n) \quad (21)$$

Gaussian approximation studied in Section 4 is assumed for equalization in the logarithmic domain so that the energy distribution is feasible or implementable.

Thus, the approximated energy distribution in logarithmic domain is equal to:

$$\log \overline{p}\ (\mathrm{E}_{n,m} \mid B_{n,m}) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log \sigma^2_{\chi^2} - \frac{(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2\sigma^2_{\chi^2} \ln 10} \quad (22)$$

One should notice that with normalization process at the output of the equalizer, the redundant constants are removed. In addition, due to hardware restraint, the energy detector is a logarithm to base 10 detector as in [23] which provides logarithmic energies per time slot T_{slot} . So, the only available data is $\log \mathrm{E}_{n,m}$, $\log B_{n,m}$ and $\log(2L\sigma^2)$. Expending $\sigma^2_{\chi^2}$ in (22), we get

$$\log \overline{p}\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2}\log[2\sigma^2(2L\sigma^2 + 2B_{n,m})] - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{4L\sigma^2(2L\sigma^2 + 2B_{n,m})\ln 10} \quad (23)$$

$$\propto -\frac{1}{2}\log(2\sigma^2) - \frac{1}{2}\log(2L\sigma^2 + 2B_{n,m}) - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2.10^{\log(2L\sigma^2(2L\sigma^2+2B_{n,m}))} \ln 10} \quad (24)$$

where the symbol $\propto$ means "*proportional to*" and the function log stands for the logarithmic to base 10. Rewriting (24) taking into account $\max^*_{10}$and removing the redundant constant such as $-\frac{\log(2\sigma^2)}{2}$, leads to

$$\log \overline{p}\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2} max^*_{10}\left[\log(2L\sigma^2), \log 2 + \log B_{n,m}\right] - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2.10^{\log(2L\sigma^2)+\log(2L\sigma^2+2B_{n,m})} \ln 10} \quad (25)$$

the devision part in (25) can be transformed into multiplicationas follows:

$$\log \overline{p}\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2} max^*_{10}\left[\log(2L\sigma^2), \log 2 + \log B_{n,m}\right] - \frac{L}{2\ln 10}(\mathrm{E}_{n,m} - m_{\chi^2})^2 10^{-\gamma} \quad (26)$$

where γ is defined as

$$\gamma = \log(2L\sigma^2) + max^*_{10}[\log(2L\sigma^2), \log 2 + \log B_{n,m}] \quad (27)$$

and $\mathrm{E}_{n,m} - m_{\chi^2}$ is easily calculated as follows

$$\mathrm{E}_{n,m} - m_{\chi^2} = 10^{\log \mathrm{E}_{n,m}} - \left(10^{\log(2L\sigma^2)} + 10^{\log B_{n,m}}\right) \quad (28)$$

It is noticed that the energy distribution in logarithmic domain is achieved by the mean of two lookup tables. A first ROM for the $\max^*_{10}$ function and a second ROM for 10^x function. The memories size will be treated in the simulation Section.

The advantage of working in logarithmic domain is that the amount of multiplications is confined only on the energy distribution. Figure 6 depicts the new energy distribution architecture implemented in digital design. This architecture is used for the probabilistic equalizer complexity study.

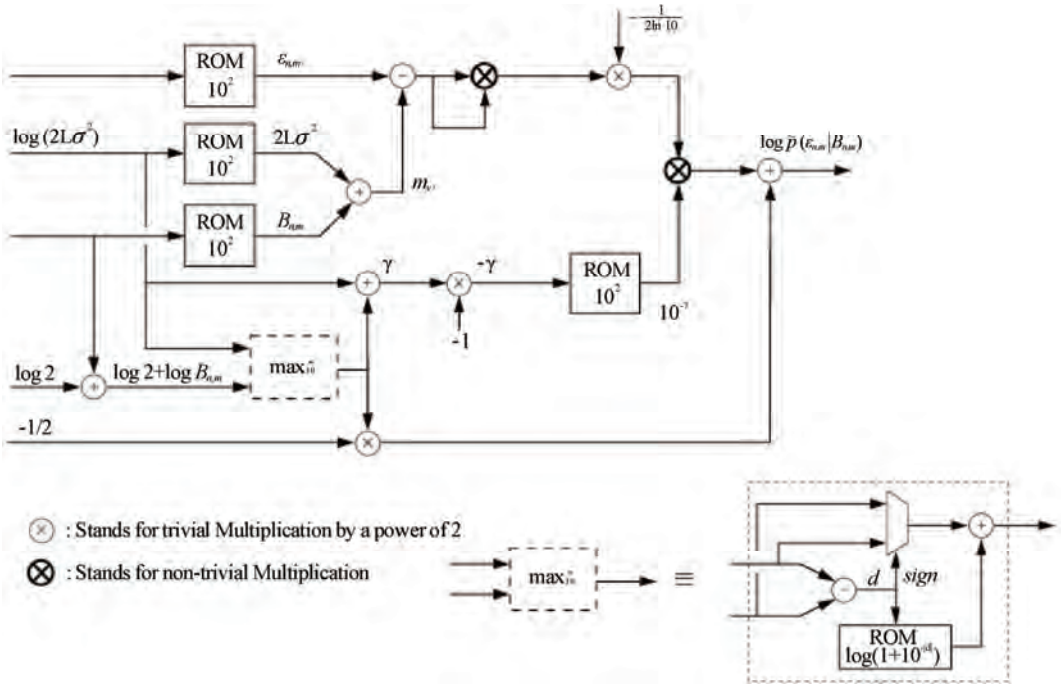


Figure 6. Energy distribution architecture for logarithmic equalizer.

7. Performance of the Logarithmic Equalizer with the Approximated Distribution

7.1. Fixed Point DataTypes Simulation in Logarithm Domain

The performances of the logarithmic equalizer with the approximated distribution are simulated in fixed point precision and compared to the chi-squared distribution in double precision. Reception is ensured by a logarithmic energy detector [23]. Simulations in fixed point precision are carried out by the mean of the class *sc_fix* of SystemC as in 5.2. Table 4 shows the word sizes of parameters considered for the fixed point simulations.

With respect to the equalizer expression (18) and to the approximated energy distribution in logarithmic domain (26), we consider two ROM types whose sizes are defined in Table 5.

Figure 7 shows the results obtained if the receiver assumes that there are only 2 interfering symbols, i.e. CSI is known only over $P = 5$ slots, however the real number of interfering symbols could be more.

We notice that for less dispersive channel such as CM1 and CM2, the results in fixed point precision data types are close to those obtained in double precision with *chi-squared* distribution. Regarding the results for highly dispersive channel (CM3 and CM4), we get a loss of $1dB$ at $BER=10^{-4}$.According to [7], the receiver could be improved if the supposed number of interfering symbols are bigger than 2 , especially in highly dispersive channel. It has been proven that for channel models CM3 and CM4, the optimal compromise is to consider $K = 3$ [7].

Simulations run with $K = 3$ for CM3 and CM4 in fixed point data types are depicted in Figure 8. Although the complexity is slightly increased due to cardinal of the set $\{B_{n,m}\}$, *i.e.* $|\{B_{n,m}\}|= 88$ for $K = 3$ [7], the receiver is improved of $2dB$ for CM3 at $BER = 10^{-4}$.

Table 4. Parameters size definition.

Parameters	Quantization size $< wl, iwl >$
$\log E_{n,m}$	<6,2>
$\log B_{n,m}$	<8,2>
$\log(2L\sigma^2)$	<7,4>
m_{χ^2}	<8,4>
$\log p(E_{n,m} \mid B_{n,m})$	<7,4>
$\log p(E_n \mid x_n)$	<6,4>
$\log \pi(x_k)$	<6,4>

Table 5. ROM input/output size.

Parameters	Input size	Output size	Table size (Kbits)
$g(x) = \log(1+10^{-\lvert d \rvert})$	<6,2>	<4,0>	0.25
$h(x) = 10^x$	<9,5>	<8,3>	4

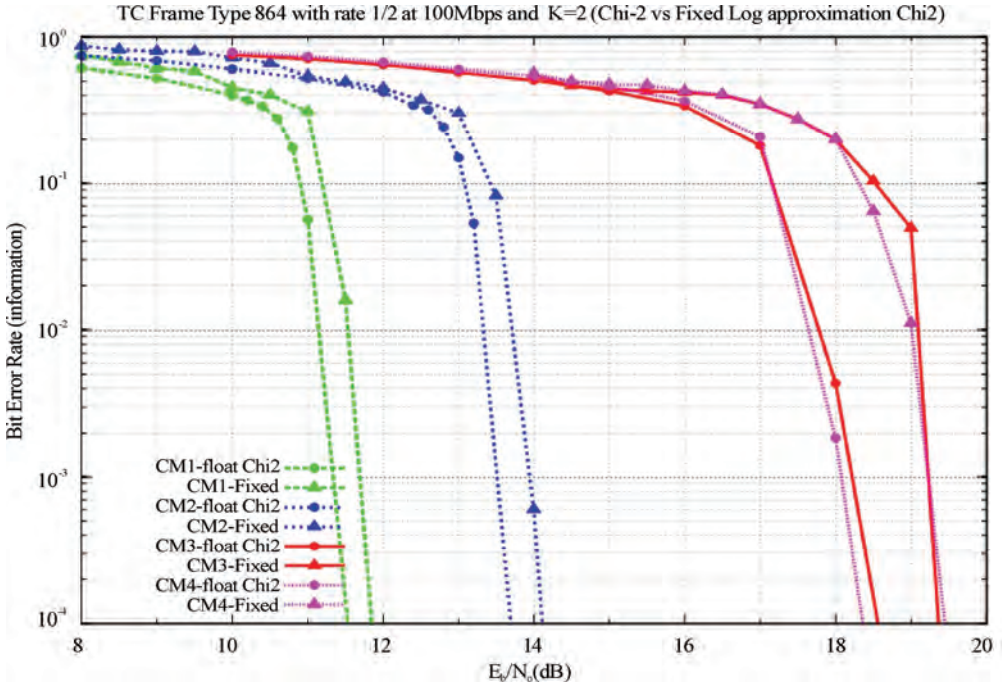


Figure 7. ***Chi-squared*** **float precision versus the logarithmic Gaussian approximation in fixed point precision for** ***K*****=2.**

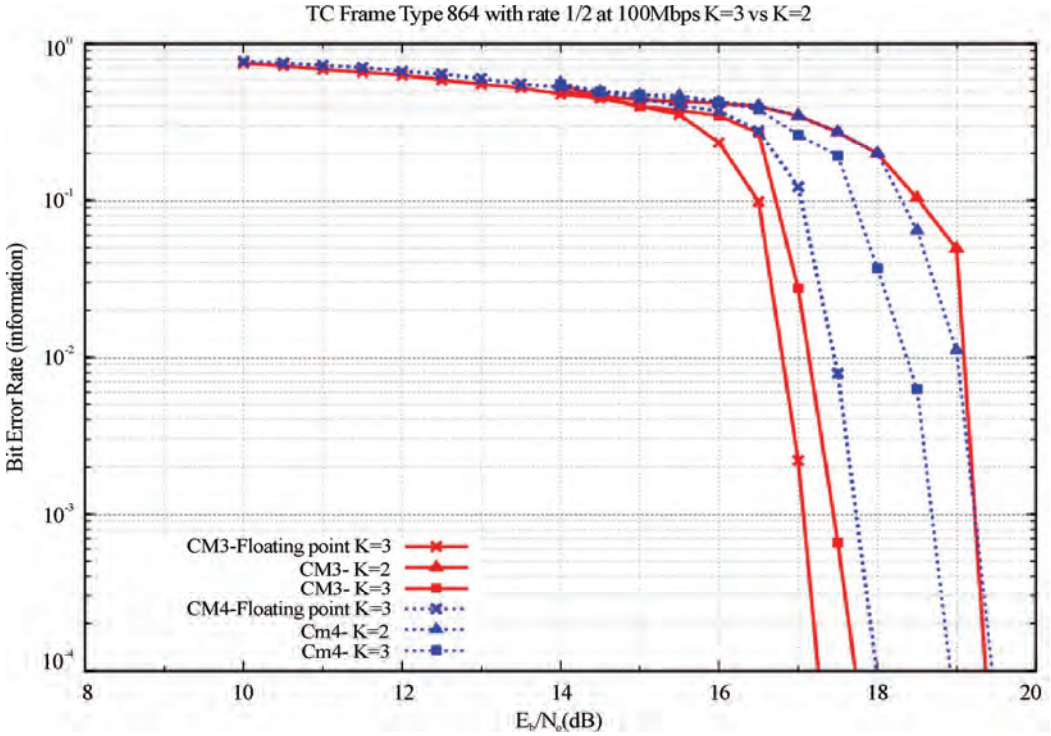


Figure 8. Simulation in fixed point data types for CM3 and CM4.

7.2. Complexity of the Logarithmic Equalizer

According to the equalizer expression (18) and Figure 9, the computational complexity of the equalizer in terms of non-trivial multiplication is equal to $2M^{K+1}$ multiplication per symbol. Thus, with 4-PPM modulation at $100Mbps$ and $K=2$, we get $6.4Gmultiplication/s$. Regarding the memory size, the logarithmic approximati-

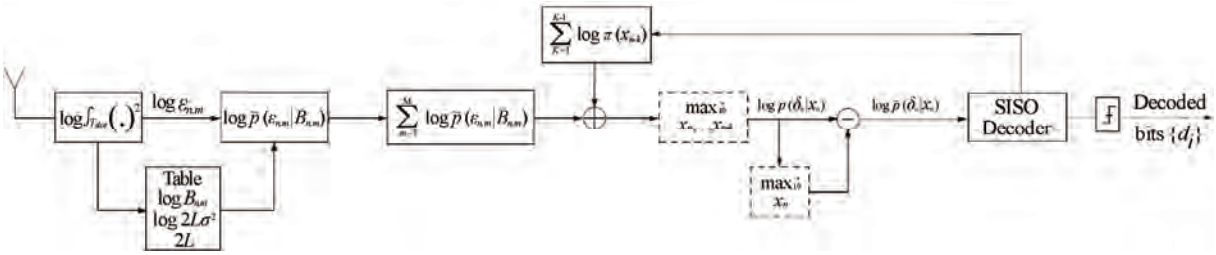


Figure 9. Equalizer architecture in the logarithmic domain.

Table 6. Complexity requirement with 4-PPM and *K*=2.

	Number of Multiplications		
Function	Linear x^2	Linear x^2 approximated	Logarithmic x^2 approximated
$p(E_n \mid x_n)$	$(M+K-2)M^K$ 3.2^0 *GMultip/s*	$(M+K-2)M^K$ 3.2 *GMultip/s*	0
$p(E_{n,m} \mid B_{n,m})$	0	$3M^{K+1}$ 9.6 *GMultip/s*	$2M^{K+1}$ 6.4 *GMultip/s*
Total Equalizer Multiplications	3. 2 *GMultip/s*	12.8 *GMultip/s*	6.4 *GMultip/s*
Total required memory per level of parallelism	448 *Mbit*	30.25 *Kbit*	16.25 *Kbit*

on requires $16.25 Kbits$. Comparing to the linear equalizer the complexity in the logarithm domain, *i.e.* number of multiplications and the memory size, is promising for a low cost hardware implementation. For instance, if the hardware runs at $400 Mhz$ and the frequency of table access is $3.2 Ghz$ for the same example quoted before, we require 8 level of parallelism to achieve the energy distribution. So the total required memory is a factor of 8; *i.e.* $16.25 Kbits \times 8 = 130 Kbits$) which is lower than the required memory in linear domain ($242 Kbits$). Moreover, the required bits for each parameters in Table 4 are shorter than in Table 2.

7.3. Complexity Summary

Table 6 is the synthesize of the complexity requirement for the linear and the logarithmic equalizer with the chi-squared distribution approximation. We notice that the logarithmic equalizer with the approximated energy distribution is far the less complex for hardware implementation, since it allows a compromise between the number of multiplications and the size of the required memory for equalization calculation.

8. Conclusion

In this paper, we a have shown how a complex and costly probabilistic equalizer is simplified for digital design by using the logarithmic domain. A first simplification concerns the energy distribution which is approximated by a Gaussian distribution instead of a chi-squared. This leads to reduce significantly the required memory for distribution computation. The second simplification is to calculate all the probabilities in the logarithmic domain by the mean of the max^*_{10} operation. Hence, the computational complexity of the equalizer is highly reduced compared to the linear equalizer. Moreover, only two lookup table types are required for equalizer calculation in logarithmic domain. Computer simulations demonstrated the performance of the receiver in finite precision. It showed, that for highly dispersive channels such as CM3 and CM4, the receiver is still able to equalize and decode the transmitted informations with a slight increase in complexity.

As perspective, some operations or memories could even be simplified or reduced by the mean of polynomial approximations with a negligible loss on the receiver performance. This could be a subject of investigation for future research.

9. References

[1] L. Yang and G. B. Giannakis, "Ultra-wideband communications: an idea whose time has come," IEEE Signal Processing Magazine, Vol. 21, No. 6, pp. 26–54, November 2004.

[2] J. D. Choi and W. E. Stark, "Performance of ultra-wideband communications with suboptimal receivers in multipath channels," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 9, pp. 1754–1766, December 2002.

[3] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted reference RF communications," IEEE Conference on Ultra-Wideband Systems and Technologies, pp. 265–269, May 2002.

[4] V. Lottici, L. Wu, and Z. Tian, "Inter-symbol interference mitigation in high-data-rate uwb systems," IEEE International Conference on Communications, pp. 4299–4304, June 2007.

[5] Y. Zhang, H. Wu, Q. Zhang, and P. Zhang, "Interference mitigation for coexistence of heterogeneous ultra-wideband systems", EURASIP Journal on Wireless Communications and Networking, pp. 1–13, 2006.

[6] M. E. Sahin and H. Arslan, "Inter-symbol interfrence in high data rate uwb communications using energy detector receivers," IEEE International Conference on UWB, ICU, pp. 176–179, September 2005.

[7] S. Mekki, J. L. Danger, B. Miscopein, J. Schwoerer and J. J. Boutros, "Probabilistic equalizer for ultra-wideband energy detection," IEEE 67th Vehicular Technology Conference (VTC), pp. 1108–1112, May 2008.

[8] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, December 1972.

[9] J. A. Erfanian and S. Pasupathy, "Low-complexity parallel-structure symbol-by-symbol detection for ISI channels," IEEE Pacific Rim Conf. Communications, Computers and Signal Processing, pp. 350–353, 1989.

[10] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," IEEE Journal On Selected Areas In Communications, Vol. 16, No. 2, pp. 260–264, February 1998.

[11] H. Urkowitz, "Energy detection of unknown deterministic signals," Proceedings of the IEEE, Vol. 55, No. 4, pp. 523–531, April 1967.

[12] A. H. M. Ross, "Algorithm for calculating the noncentral chisquare distribution," IEEE Transactions on Information Theory, Vol. 45, No. 4, pp. 1327–1333, May 1999.

[13] N. C. Severo and M. Zelen, "Normal approximation to the chisquared and non-central F probability functions," Biometrika, Vol. 47, No. 3/4, pp. 411–416, December 1960.

[14] L. Canal, "A normal approximation for the chi-square distribution," Computational Statistics & Data Analysis, Vol. 48, No. 4, pp. 803–808, April 2005.

[15] J.-T. Zhang, "Approximate and asymptotic distributions of chi-squared-type mixtures with applications," Journal of the American Statistical Association, Vol. 100, pp. 273–285, March 2005.

[16] R. Saadane, D. Aboutajdine, A. M. Hayar, and R. Knopp, "On the estimation of the degrees of freedom of in-door UWB channel," IEEE 61st, Vehicular Technology Conference (VTC), Vol. 5, pp. 3147–3151, May 2005.

[17] J. G. Proakis, Digital Communications, Second Edition, New York, McGraw Hill, 1989.

[18] J. Foerster, "Channel modeling sub-committee report final," IEEE P802.15-02/368r5-SG3a, Tech. Rep., 18 November 2002.

[19] S. Mekki, J. L. Danger, B. Miscopein, and J. J. Boutros, "EM channel estimation in a low-cost UWB receiver based on energy detection," IEEE International Symposium on Wireless Communication Systems 2008 (ISWCS 08), pp. 214–218, October 2008. [Online]. Available: http://samimekki.free.fr/.

[20] "Digital video broadcasting (DVB); interaction channel for satellite distribution systems," ETSI EN 301 790 V.1.3.1, Tech. Rep., March 2004.

[21] "Digital video broadcasting (DVB); interaction channel for satellite distribution systems; guidelines for the use of en 301 790," ETSI TR 101 790 V.1.2.1, Tech. Rep., January 2003.

[22] "TC1000-xX DVB-RCS Turbo Decoder v2.1," Turbo-Concept, Tech. Rep., February 2005.

[23] AD8318 1 MHz to 8 GHz, 70 dB Logarithmic Detector/Controller. [Online]. Available: http://www.analog.com/en/prod/0%2C2877%2CAD8318%2C00.html.

[24] SystemC User's Guide, version 2.0. [Online]. Available: http://www.systemc.org.

[25] W. J. Gross and P. G. Gulak, "Simplified MAP algorithm suitable for implementation of turbo decoders," Electron ics Letters, Vol. 34, No. 16, pp. 1577–1578, August 1998.

Appendix

max^*_{10} Proprieties

$$max^*_{10}(a,b,c) = max^*_{10}[max^*_{10}(a,b),c] \tag{29}$$

Proof.

From the definition of max^*_{10} we have

$$max^*_{10}(a,b,c) = \log(10^a + 10^b + 10^c) \tag{30}$$

in other hand we can write

$$10^a + 10^b = 10^{\log(10^a+10^b)} = 10^{max^*_{10}(a,b)} \tag{31}$$

Rewriting (29) taking on consideration (30), we get

$$max^*_{10}(a,b,c) = \log(10^{max^*_{10}(a,b)} + 10^c) \tag{32}$$

$$= max^*_{10}[max^*_{10}(a,b),c] \tag{33}$$

Wireless Sensor Network, **2009, 1, 257-267**
doi:10.4236/wsn.2009.14032 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Wireless Sensor Network Management and Functionality: An Overview

Dimitrios GEORGOULAS, Keith BLOW
Adptive Communication Networks Research Group,
EE, Aston University, Aston Triangle, B47ET, United Kingdom
Email: *dimitriosgeorgoulas@yahoo.com*
Received March 23, 2009; *revised May* 5, 2009; *accepted June* 12, 2009

Abstract

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. They have applications in a variety of fields such as environmental monitoring; military purposes and gathering sensing information in inhospitable locations. Sensor nodes have various energy and computational constraints because of their inexpensive nature and adhoc method of deployment. Considerable research has been focused at overcoming these deficiencies through more energy efficient routing, localization algorithms and system design. Our survey presents the fundamentals of wireless sensor network, thus providing the necessary background required for understanding the organization, functionality and limitations of those networks. The middleware solution is also investigated through a critical presentation and analysis of some of the most well established approaches.

Keywords: Wireless Sensor Networks, Organization, Functionality, Middleware

1. Introduction

Wireless sensor networks have been identified as one of most important technologies for the 21st century [1]. As technologies advance and hardware prices drop, wireless sensor networks will find more prosperous ground to spread in areas where traditional networks are inadequate. The foundational concept which applies in a vast number of networks can be identified through the simple notion: Sensing Capabilities plus CPU Power plus Radio Transmission equals a powerful framework for deploying thousands of potential applications.

However, this notion is underlined by some complex and detailed understanding of each separate network components capabilities and limitations as well as understanding in areas of modern network management and distributed systems theory.

The primary goal of wireless sensor networks is to make useful measurements for as long as possible. To do this it is essential to minimize energy use by reducing the amount of communication between nodes without sacrificing useful data transmission. Each node is designed in an interconnected web that will grow upon the deployment in mind. Wireless sensor networks are highly dynamic and susceptible to network failures, mainly because of the physically harsh environments that they are deployed in and connectivity interruptions [2].

To make the wireless sensor network dream a reality, an architecture must be developed that will monitor and control the node communication in order to optimize and maintain the performance of the network, ensure that the network operates properly and control/instruct a set of cluster nodes without human intervention [3].

In order to develop a system architecture with the above characteristics, we focus explicitly on the functions and the roles of wireless network management systems. Additionally, we present the middleware concept as a novel solution to the limitations that wireless sensor networks inhabit. A number of network systems are presented, critical reviewed and categorized.

2. Network Management Systems

Around 1980s computer networks began to grow and be interconnected in a large scale. This growth produced problems in maintaining and managing those networks, thus the need of network management was realized. Today, networks are far more dynamic and interconnected

than before, especially in the area of wireless sensor networks, thus a managing infrastructure is one of the most basic requirement for monitoring and controlling such networks [4].

A network management system can be defined as a system with the ability to monitor and control a network from a central location. Ideally there are four key functional areas that this system must support [5]:

1) Fault Management: This area provides the facilities that allow the discovery of any kind of faults that the managed devices of the network will produce, determining in parallel the possible causes of such errors. Thus, the fault management function should provide mechanisms for error detection, correction, log reports and diagnostics preferably without the user interference.

2) Configuration Management: Responsible for monitoring the entire network configuration information and also having access to all the managed devices in terms of reconfigure, operate and shut down if necessary.

3) Performance Management: Responsible for measuring the network performance through analysis of statistical data about the system so that it may be maintained at an acceptable level.

4) Security Management: This area provides all those facilities that will ensure that access to network resources can not be obtained without the proper authorization. In order to do so, it provides mechanisms for limiting the access to network resources and provides the end user with notifications of security breeches and attempts.

Those four functional areas of network management are far more challenging and vital for a network that will consist of tiny sensors which can be supplied to a specific environment running applications such as habitat monitoring, microclimate research, medical care and structural monitoring [6]. For every sensor network application, the network is presented as a distributed system consisting of many autonomous nodes that cooperate and coordinate their actions based on a predefined architecture. Every node is assigned with a specific role inside the network such as data acquisition and processing. Also, nodes can be used as data aggregation and caching points in order to reduce the communication overhead [7].

3. The System Organization of a Sensor Network System

The organization of sensor network systems is based upon the approach that they will adapt in order to monitor and control the state of the wireless sensor network. There are four predominant approaches [8]:

- Passive Monitoring: The system role is to collect data during the lifetime of the network. The data will identify the state of the network in different time intervals without any action taking place during the data gathering. An analysis of the data will take place in later stages.
- Fault Detection Monitoring: The system dedicates its resources to identifying faults and errors during the lifetime of the network. All the information is gathered and reported back to the operator whose responsibility is to correct those problems in later stages. No action is taken by the system towards the resolution of those problems in real time.
- Reactive Monitoring: The system has a double role to accomplish during the lifetime of the network. Firstly, as we identified in the previous approaches, the collection of data that will provide information about the states of the network, is the main role. This time though, the system will be eligible to identify and detect any events and act upon them in real time mainly by altering the parameters of the fixed asset under its control.
- Proactive Monitoring: The system collects and analyzes all the incoming data concerned with the state of the network. Then an analysis is taking place similar to the one of the reactive monitoring with the big difference that certain events, described by the collected information, are stored. The system is then able to maintain better available network performance by predicting future events based on past ones.

Wireless sensor systems can be categorized according to their architecture which can be centralized, hierarchical or distributed [9]. The centralized one identifies the role of the base station as the most important one in the whole architecture. The base station will collect the information from all the nodes and will monitor and control the entire network. Benefits to this architecture can be found in areas of processing power and decision making. A base station with unlimited power resources can perform complex analysis of data and process a variety of information, reducing the weight of this energy consuming task from the nodes of the network.

The distributed architecture focuses on the deployment of multiple manager stations across the network usually in a web based format. Thus, each substation can coordinate its actions and co-operate based on knowledge that it can acquire from a neighboring substations. In this approach, the communication cost is less than the centralized one and more energy efficient since all the workload will be distributed evenly across the network. However, due to the scalability and complexity of wireless sensor networks it is proven quite difficult to manage and quite expensive in terms of memory cost.

The hybrid between the centralized and distributed approach is that of the hierarchical one. In this architecture we have the existence of substations in the network but this time no communication is allowed between them. The design tends to be cluster based, with the heads of the cluster be responsible for a set of network nodes in

terms of processing and transmitting information. All the cluster heads will report back to a single base station.

4. The Functionality of a Sensor Network System

The main functionality of sensor network systems is based on the theory behind network management systems, thus is focusing on two attributes those of monitor and control. In this section, we classify some well known sensor systems in terms of the functionality they provide inside the network. Figure 1 is demonstrating this classification.

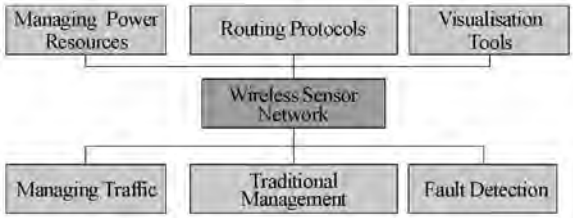


Figure 1. The functionalities of wireless sensor networks in different categories.

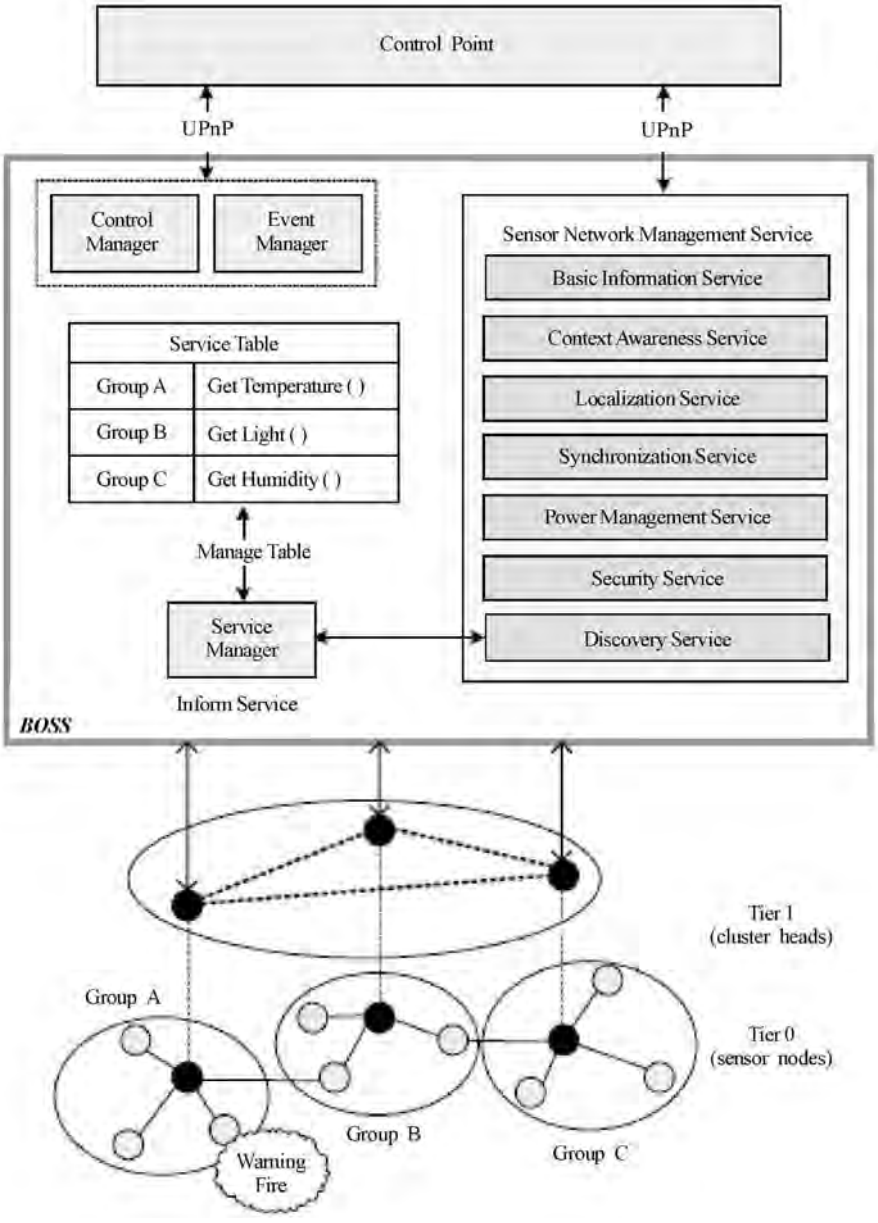


Figure 2. The BOSS architecture, song *et al* 2005.

Two characteristic examples of wireless sensor networks that are based on traditional management systems are those of MANNA [10] and BOSS [11]. MANNA provides a general architecture for managing a wireless sensor network by using a multidimensional plane for the functional, physical and the informational architecture of the network. The functional plane is responsible for the configuration of the application specific entities, the information plane is object oriented and specifies all the syntax and semantics that will be exchanged between the entities of the network and lastly the physical plane establishes, according to the available protocols profiles, the communication interfaces for the management entities that will be present inside the network.

The BOSS architecture, Figure 2, is based on the traditional method of the standard service discovery protocol, UpnP. With the UpnP protocol the user does not need to self configure the network and devices in the network automatically will be discovered. However, due to computational power consumption required by the devices and memory space allocation limitations, the protocol itself is not suitable for tiny sensor devices. BOSS architecture is overcoming this problem by acting as a mediator between UpnP networks and sensor nodes. In order to do that, it combines four different components: service manager, control manager, service table and a sensor network management service, under the same framework.

Routing protocols is another alternative way of monitoring and controlling a wireless network when they get embedded in an application with examples such as LEACH [12] and GAF [13].

LEACH is a routing protocol for users that want remotely to monitor an environment. The protocol is build upon two foundational assumptions. The first one acknowledges that the base station is at a fixed point and in a far distance from the network nodes and the second one assumes that all nodes in the network are homogeneous and energy constrained. In order to maximize the system lifetime and coverage, LEACH is using a set of methods such as distributed cluster formation with randomized selection of cluster heads and local processing. LEACH dynamic clustering method, splits time in fixed intervals with equal length. Also, it does not allow clusters and cluster heads to be at a fixed point inside the network. LEACH, dictates that once other sensors of the network receive a message they will join a cluster with the stronger signal cluster head.

GAF, which stands for geographic adaptive fidelity, focuses its architecture on the extension of the lifetime of the network by exploiting node redundancy. This node redundancy is achieved by switching off unnecessary sensor nodes in the network without any effect on the level of routing fidelity.

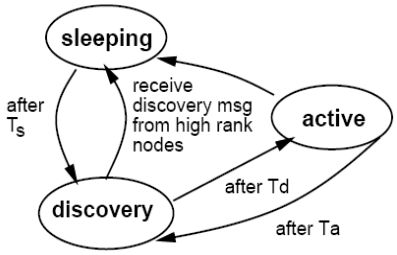


Figure 3. The GAF nodes state transitions, xu *et al* 2001.

GAF recognizes three transition states for the nodes, Figure 3, active, sleeping and discovery. Initially all nodes in the network are in a discovery state. This means that all nodes will turn their radio on and exchange discovery messages in order to identify neighbor nodes in the same grid. When a node is active it will set a timeout, Ta, that will determine for how long it will stay in that state before it returns back to the discovery state. While active, the node periodically re-broadcasts its discovery message at time intervals, Td. The sleeping state is regulated by a time interval Ts which is dependent upon the application. GAF assumes that sensor nodes can identify their location in the forming virtual grid with the use of GPS cards.

Systems, such as WinMS [14] and Sympathy [15] focus more on the importance of fault detection in a wireless sensor network. WinMS uses a novel management function, called systematic resource transfer, in order to provide automatic self-configuration and self-stabilization both locally and globally for the given wireless sensor network. This function allows the network, in case of a failure, to have a predetermined period of time where nodes will listen to their environment activities and self-configure. No prior knowledge of the topology of the network is necessary. WinMS, uses a TDMA-based MAC protocol, called FlexiMAC [14], in order to support resource transfer among nodes in the network. FlexiMac protocol provides synchronized communication between the nodes. Thus, it can adaptively adjust the network by providing local and central recovery mechanisms.

Sympathy is a tool for detecting and debugging failures in wireless sensor networks, but unlike WinMS it does not provide any automatic network reconfiguration incase of a failure. One of the main functionalities of the system is the collection and analysis of network information metrics such as nodes next hop and neighbors. By doing so, it is able to identify which of the nodes deliver insufficient data to the sink node or to the base station and locate the cause by reporting back to the end user. One of the major advantages of Sympathy is that it takes into account interactions upon multiple nodes however, by doing so it will require nodes to exchange neighborhood lists, something that has proven highly costly in terms of energy levels.

Another very useful functionality of wireless sensor systems is that of the visualization of the actual network. This ability of an end user to demonstrate graphical representations of the different states of the network at various time intervals can be found in systems such as the TinyDB [16] and MOTE-VIEW [17].

TinyDB is a distributed query processor for sensor networks. It uses an SQL like interface for collecting data from nodes in the given environment and also provides aggregation, filtering and routing of the acquired results back to the end user. With the use of a declarative language for specific user queries, TinyDB proves to be flexible in two domains. Firstly, all the queries that are generated are easy to read and understand and secondly the underlying system will be responsible for the generation and the modifications of the query without the query itself to need any modifications. In the core of the system we find a metadata catalog that identifies the commands and attributes that are available for querying.

The MOTE-VIEW system is an interface system between the end user and the deployed network of wireless sensors. Through this interface the user can make alterations to node characteristics in terms of radio frequency, sampling frequency and transmission power. The system architecture is based on four layers; data access abstraction, node abstraction, conversion abstraction and visualization abstraction layer. The data abstraction layer acts as the database interface where all the data is been stored. The node abstraction layer collects and stores all the nodes metadata which will create relational links with the database. All the raw data that is going to be collected from the nodes will be translated into understandable engineering units at the conversion abstraction layer. Finally, the visualization abstraction layer will provide to the end user displays of the data in forms of spreadsheets and charts.

MOTE-VIEW is a passive monitor system in that it does not provide any interpretation of the displayed graphical data on behalf of the user. However, in terms of network and other failures MOTE-VIEW does not provide any self-configuration scheme.

The resource management is one of the key aspects of every wireless sensor network. Systems such as the Agent Based Power Management [18] and SenOS [19] have been created with that concern in mind. The Agent Based Power Management is a system that builds its architecture upon mobile agents. These intelligent entities are set responsible for local power management processing by applying energy saving strategies to the nodes of the network. This agent-based scheme is suitable for applications where the state of the network is partial visible at a known time or location. One of the major concerns of this approach is that by minimizing the transmission power, the communication range of the nodes will be reduced accordingly, threatening the network connectivity.

SenOS is managing network power resources by instructing nodes to sleep when they are not active inside the network. To achieve this, SenOS adapts a dynamic power management algorithm known as DPM. The DPM algorithm, by observing events inside the network, can generate a policy for state transitions. Based on that, all redundant nodes are placed inside a cluster with only one node awake for a period of time per cluster while the others are in a sleep mode for conserving energy. The SenOS architecture is based on a finite state machine which consists of three components. Firstly, we have the kernel which provides a state sequencer and an event queue. The second component is a transition table and the final component is a call back library.

Siphon [20] and DSN RM [21] are two representative systems that provide traffic management functions in their architecture. Siphon, is based on a Stargate implementation of virtual sinks in order to prevent congestion at near base stations inside the network. These virtual sinks act as intermediates between the actual nodes and the base stations and they are distributed randomly inside the network. If at any point the rate of generate data increases beyond a level that exists in a predetermined threshold inside the system then the virtual sinks will redirect the traffic to other visible nodes. The visibility of the available nodes by the virtual sinks is one of the disadvantages of this approach, as there is a high probability that some nodes will be not covered by any virtual sink.

DSN RM (Distributed sensor network resource management) uses single radio nodes to evaluate each of their incoming and outgoing data rate and apply delay schemes to those nodes when necessary in order to reduce the amount of the traffic in the network. In every DSN there are a number of decision stations whose role is to act as data managers in a hierarchical format. However, the effectiveness of this technique is tightly bound on finding reliable data for every decision station inside the wireless sensor network that from its nature can provide inaccurate data during its lifetime due to connectivity and radio problems.

Table 1 presents a tabular evaluation of the currently available systems in terms of their organization and functionality.

5. The Role of Middleware in Wireless Sensor Networks

Many researchers have identified, that a middleware inside a wireless sensor network can establish a framework for bridging the gap between applications and low levels constructs such as the physical layer of the sensor nodes.

Table 1. Wireless sensor network systems evaluation based on designed criteria.

Wireless sensor Network Systems	Reactivity	Architecture	Function	Energy efficiency	Adaptability	Memory efficiency	Scalability
BOSS	Proactive	Centralised	Management System	Yes	Yes	Yes	Yes
MANNA	Proactive	Hierarchical	Management System	N/A	N/A	N/A	N/A
LEACH	Proactive	Distributed	Routing Protocol	Yes	Yes	Yes	Yes
GAF	Proactive	Distributed	Routing Protocol	Yes	Yes	Yes	Yes
WinMS	Proactive	Hierarchical	Fault Detection	Yes	Yes	Yes	Yes
Sympathy	Proactive	Centralised	Fault Detection	Yes	Yes	Yes	No
TinyDB	Passive	Centralised	Visualization Tool	Yes	No	Yes	Yes
MOTE- VIEW	Passive	Centralised	Visualization Tool	Yes	No	Yes	Yes
SensOS	Reactive	Hierarchical	Management resources	Yes	No	Yes	No
A. B. P. M	Proactive	Distributed	Management resources	Yes	Yes	Yes	No
DSNRM	Proactive	Hierarchical	Traffic Control	Yes	Yes	No	No
Siphon	Proactive	Distributed	Traffic Control	No	Yes	Yes	No

A middleware can be visualized as a network managing software mechanism that will create communication bonds with the network hardware, the operating system and the actual application. A fully implemented middleware should provide to the end user a flexible interface through which actions of coordination and support will take place for multiple applications preferably in real time.

This section identifies some of the most well known middleware approaches that have been developed in recent years and classifies them according to their programming paradigm. This classification presents middlewares as virtual machines based on modular programming, virtual database systems and adaptive message oriented systems.

The use of a virtual machine inside a middleware is a flexible approach since it can allow a programmer to partition a large application into smaller modules. The middleware will inject and distribute those modules inside the wireless sensor network with the use of a predefined protocol that will aim to reduce the overall energy and resource consumption. The main role of the virtual machine is to interpret those distributed modules. The communication protocol can be designed based on modular programming. The use of mobile code can facilitate an energy efficient framework for the injection and the transmission of the application modules inside the network.

The Mate [22] middleware is among those that use a virtual machine in order to send applications inside the wireless sensor network. The developers having identified the predominant limitations of wireless sensor networks such as energy consumption and limited bandwidth propose a new programming paradigm that is based on a tiny centric virtual machine that will allow complex programs to be very short. In order to achieve that, Mate's virtual machine acts as an abstraction layer with content specific routing.

This section identifies some of the most well known middleware approaches that have been developed in recent years and classifies them according to their programming paradigm. This classification presents middlewares as virtual machines based on modular programming, virtual database systems and adaptive message oriented systems.

The use of a virtual machine inside a middleware is a flexible approach since it can allow a programmer to partition a large application into smaller modules. The middleware will inject and distribute those modules inside the wireless sensor network with the use of a predefined protocol that will aim to reduce the overall energy and resource consumption. The main role of the virtual machine is to interpret those distributed modules. The communication protocol can be designed based on modular programming. The use of mobile code can facilitate an energy efficient framework for the injection and the transmission of the application modules inside the network.

The Mate [22] middleware is among those that use a virtual machine in order to send applications inside the wireless sensor network. The developers having identified the predominant limitations of wireless sensor networks such as energy consumption and limited bandwidth propose a new programming paradigm that is based on a tiny centric virtual machine that will allow complex programs to be very short. In order to achieve that, Mate's virtual machine acts as an abstraction layer with content specific routing.

Figure 4 presents Mate architecture and execution model. This high level architecture will enable the programming code to break up into small capsules of 24 instructions each that can self-replicate inside the network.

This architecture enables Mate to begin execution in response to a specific event such as a packet transmission or a time out. This is applicable through the three execution contexts that refer to an equal amount of events: clock timers, message receptions and message send re-

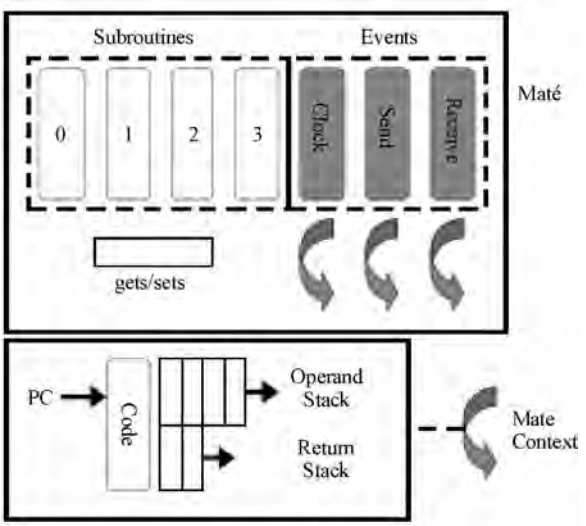


Figure 4. Mate architectural concept, P. Levis and D. Culler (2002).

quests. Each of the three contexts has an operant stack and a return address one. The operant stack will be used for instructions handling all the data while the return address stack will handle all the subroutine calls.

Every capsule that is sent inside the network includes a type and a version number. Based on that information, Mate can achieve easy version updates by adding a new number to the capsule every time a new version of the program is uploaded. However, Mate middleware suffers from the overhead that every new message introduces and also all the messages are transmitted by flooding the network in order to minimize asynchronous events notifications, raising issues with the energy consumption of every node inside the network.

Agilla [23] is a middleware that provides a mobile code paradigm for programming and making effective use of a wireless sensor network. Agilla applications consist of mobile agents that can clone or migrate across the network. The framework is based on the fact that each agent is acting as an autonomous entity inside the network allowing the developer to run parallel processes at the same time. Agilla is based on the Mate architecture in terms of the virtual machine specifications but unlike Mate, which as we described above divides an application into capsules flooding the network, Agilla uses mobile agents in order to deploy an application.

Figure 5 presents the Agilla model identifying the communication principle between two neighbor network nodes.

In every network node we can have one or more agents residing and working independently. Their intercommunication and coordination is established by local tuple spaces that are accessible by all the agents resident in that node and a neighbor list. The local tuple space is a shared memory architecture that is addressed by field-matching. A tuple can be defined as a sequence of data objects that is inserted into the tuple space of each node by every agent. These data objects will remain in the node regardless of the agent status. In due time, an agent can retrieve an old tuple by template matching. In order to do so the sending agent must generate a query for that tuple, matching the exact same sequence of fields. The neighbor list contains the addresses of neighboring nodes and is accessible for every agent in the network that wants to clone or migrate in a different location.

Based on these attributes, Agilla allows network reprogramming thereby eliminating the power consumption cost of flooding the network. However, the lack of a hierarchical communication model for the agent society and the absence of precise real location information of every node in the network can lead to deadlocks and memory management problems.

What is known as a database middleware will visualize the whole network as a virtual database system. The middleware in this case will provide the user with an interface for sending queries to the sensor nodes of the system to extract the desired data.

The Cougar middleware adopts the above approach by considering the extracted sensor data to be a virtual relational database. The developers by using an SQL-like language assume that the whole network is the database. The contents of such a database are stored data and the sensor data. The stored data is represented as a virtual relationship between the sensors that participate in the network and the physical characteristics. The sensor data which is the outcome of processing functions is represented as time series that will be adapted towards the query formulation.

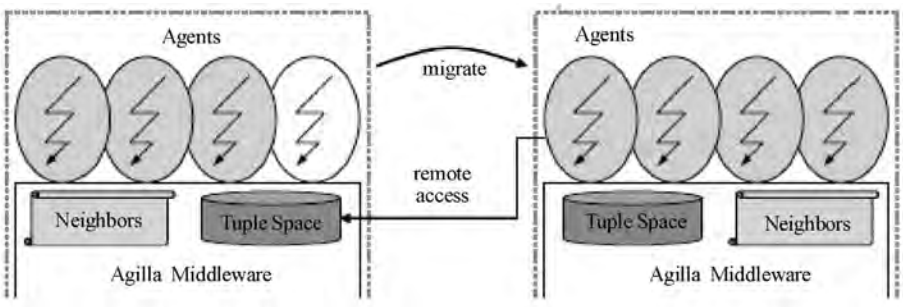


Figure 5. The Agilla architectural concept, C. Fok and G. Roman (2005).

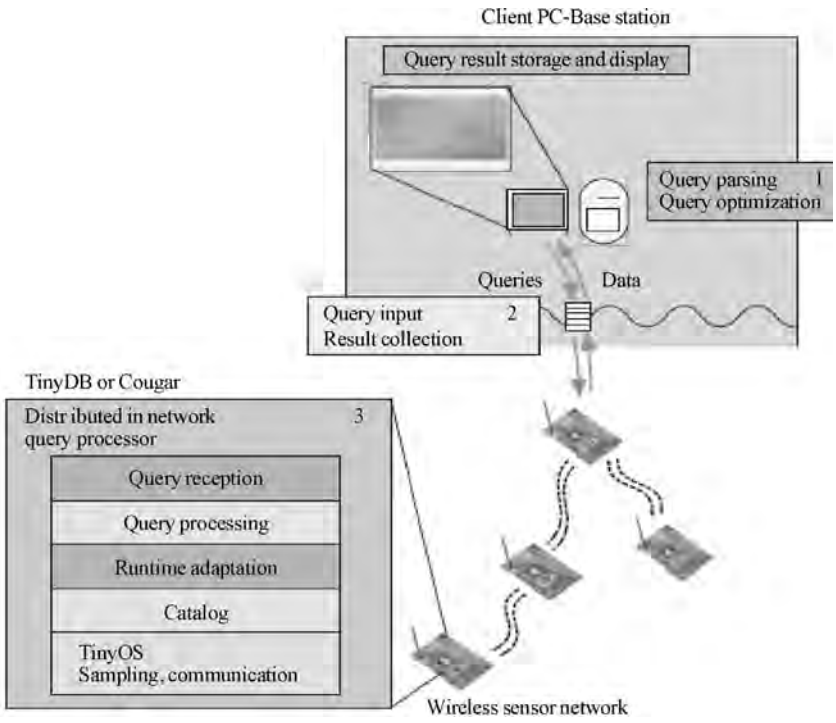


Figure 6. Cougar architecture for query processing, G. Gehrke and S. Madden (2002).

Figure 6 shows a block diagram that can explain the Cougar architecture for querying processing. One block presents the user end with the base station in the active role of transmitting and receiving queries from the wireless sensor network. The second block presents the distributed network query processor that consists of a number of abstract data types with virtual relationships with the operating system of the network. In an SQL query format of SELECT-FROM-WHERE-GROUP-INCLUDES Cougar can allow the user to access this object relational database which mirrors the actual network.

The third class of middleware is message oriented. Their core architecture is based on creating a communication model that will facilitate message exchanges between nodes and the sink nodes of the wireless sensor network. To achieve that most middlewares will adapt a publish-subscribe mechanism, an asynchronous communication paradigm which allows a loose coupling between the sender and the receiver saving precious power resources.

Mires [24] middleware provides such an asynchronous communication model. This model defines three distinct faces for the nodes resident in the network. Initially the network nodes will advertise their sensed data (topic). Using a multi-hop routing algorithm Mires will route those advertisement messages to the sink node. Lastly, a user application interface will select the desired advertised topics to be monitored.

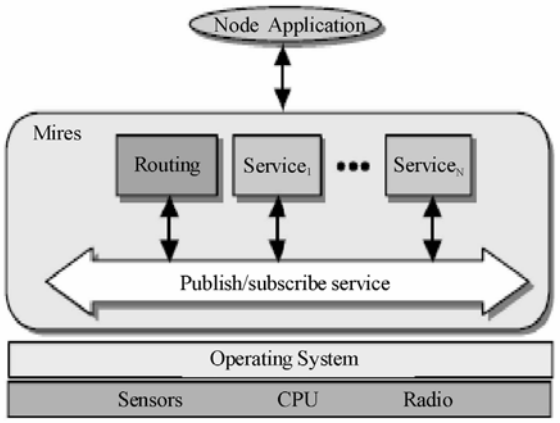


Figure 7. The Mires architecture, E. Souto and G. Vasconcelos (2004).

Figure 7 demonstrates the key characteristics of the Mires architecture. The bottom block consists of the hardware components of the node which are directly interfaced and controlled by the operating system. The middleware is placed on top of the operating system to implement its publish-subscribe communication model. This model is able to advertise the sensor data (topics) provided by the running application while it maintains a topic list provided by the node application. Mires send only messages referring to subscribed topics thus reducing like that the numbers of the transmitted packets and therefore saving energy.

6. Towards the Design of the In-Motes Middleware

In order to design and develop a successful middleware solution for a wireless sensor network that will be able to satisfy some if not all the functionalities of a network management system for monitor and control there are certain design criteria, which we described in the previous sections, and must be considered and brought forward in our design. In the following paragraphs we are going to present those design principles for a substantial middleware development.

A wireless sensor network consists of tiny devices that are battery powered and provided with a small CPU processor. Usually, and as we have already mentioned, they can be deployed in hundreds and typically in physical harsh environments. It is obvious, that after the deployment a physical contact for replacement or maintenance is highly unlikely. A middleware should be able to provide remote access to these nodes making sure that they will exhaust all their resources in terms of battery power and memory in a timely manner. Hence, one of the basic design principles for our middleware is the ability to manage limited power and resources.

Our approach [25] in order to satisfy the above design criterion is based in the creation of a flexible communication protocol between the nodes of our network and the base station. Thus, inspired from the GAF protocol that was described in the previous section, we are aiming to develop a protocol having node redundancy in mind, thus to regulate in an energy efficient manner which of the nodes of our wireless sensor network will be active and which ones will be in a sleep mode. Subject to our trials and the development of our middleware this protocol will be introduced both hand written in the middleware engine as well as it will be introduced as part of the running application.

Field trials such as the CodeBlue Project [26] and the Wireless Sensing Vineyards [27] identified that a wireless sensor network topology is subject to frequent changes due to factors such as device failure, interference, mobility and moving obstacles. Also, they proved that it is very possible that an application will grow in time, therefore mechanisms for a dynamic network topology should be available from the middleware. A middleware should be able to adapt to parameter changes caused by unexpected external factors of the environment and also provide mechanisms for fault tolerance and self configuration of the nodes inside the wireless sensor network [28].

Based on the above observations, and inspired from approaches similar to WinMS and Sympathy our middleware will incorporate some mechanisms for fault detection and prevention together with a flexible way of reconfiguring the network in real time [29]. As Sympathy is a fully automated system, we will be aiming to provide some kind of automatic mechanisms in our middleware for the above design criteria without though this to be our first priority.

Unlike traditional networks, sensor networks and their applications are real-time phenomena with dynamic resources. Upon deployment of an application, core parameters such as energy usage, bandwidth and processing power cannot be predefined due to the dynamic character of these networks. A middleware should be designed with mechanisms that should allow the network to run efficiently and as long as possible. Such mechanisms include resource discovery and location awareness for the nodes in the system. Low-level programming models must be introduced as well in order to bridge the gap between the running application and the hardware.

As we mentioned before, mechanisms that are predominant in traditional networks are not sufficient to maintain the quality of service of a wireless sensor network because of constraints such as the dynamic topology and the power limitations. A middleware should be able to provide and maintain the quality of service over a long period of time while in parallel to be able to adapt in changes based on the application and on the performance metrics of the network like these of energy consumption and data delivery delay.

Inspired both form the work of the Mate and Agilla middlewares we will introduce a mechanism that will allow mobile code to be transmitted inside our network and will be able to allow changes in network parameters such as the bandwidth of each node as well as application parameters such as switching between available sensors of each node. Thus, an architecture will be developed adapting technologies such as Linda-like tuple spaces and agents/mobile code transmission with the support of a virtual machine engine [30].

Wireless sensor networks can be widely deployed in areas such as healthcare, rescue and military, all of these are areas where information has a certain value and is very sensitive. The environments of those areas tend to be very active and harsh, increasing the chances for malicious intrusions and attacks such as denial of service. Traditional approaches and mechanisms used to secure the network cannot be applied in this kind of network since they are heavy in terms of energy consumption. A middleware must be able to provide a secure framework for deploying applications inside the network. During the life-cycle of the network the middleware should establish protective mechanisms to ensure security requirements such as authenticity, integrity and confidentiality.

Table 2 presents a tabular evaluation of the currently available middleware systems in terms of their organization and functionality.

Table 2. Wireless sensor network middleware systems evaluation based on designed criteria.

Project	Main features	Openness	Scalability	Mobility	Heterogeneity	Power Awareness	Easy of use
Mate	Mobile active Capsules, TinyQS, Byte code interpreter	Full	Full	Full	Partial	Full	Little
Agilla	Generic Agents, TinyQS	Full	Partial	Full	Partial	Little	Average
Cougar	Virtual Database, SQL like language	Partial	Partial	Partial	Partial	Partial	Full
Mires	nesC, TinyQS, message oriented	Full	Full	Partial	Full	Full	Full
In-Motes	nesC, TinyQS, agents, behavioral rules	Full	Full	Full	Full	Full	Full

7. Conclusions

This survey presents and demonstrates the wireless sensor networks as one of the predominant technologies for the 21st century. The foundational concept behind this technology is explained together with the limitations and barriers that are incorporated and need to be addressed in the process of making a wireless sensor network applicable for a number of useful applications in modern societies. The survey opens by presenting the foundational functions of a network management system. Functions that are based upon two main attributes those of monitor and control and are critical for every wireless sensor system. A number of wireless sensor network systems are critically reviewed providing an analytical explanation of their architecture and identifying pros and cons thus helping us draw some important lessons for our proposed system. The survey continues by presenting the middleware solution as a key player in overcoming the wireless sensor networks limitations and as our main methodology behind our proposed approach. A classification and analysis of the current middleware approaches for the wireless sensor networks is produced. The survey concludes by presenting the role and the key design principles our middleware approach.

8. References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," IEEE Computer, pp. 41–49, 2004.

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and D. Culler, "System architecture directions for networked sensors," Architectural Support for Programming Languages and Operating Systems, pp. 93–104, 2000.

[3] A. Boulis, C. C. Han, and M. B. Srivastava, "Design and implementation of a framework for efficient and programmable sensor networks," International Conference On Mobile Systems, Applications And Services, pp. 187–200, 2003.

[4] Cisco Systems, "Network management systems organization," 2000, http://www.ddirv.lv/doc_upl/sazonovs1.pdf.

[5] W. Stallings, "Wireless communications and networks," ISBN-10: 0131918354, pp. 45–50, 2004.

[6] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," IEEE Computer, pp. 41–49, 2004.

[7] I. F. Akyildiz, Y. Sankarasubramaniam, W. Su, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, pp. 393–422, 2002.

[8] W. L. Lee, A. Datta, and R. Cardell-Oliver, "WinMS: Wireless sensor network-management system," CSSE Technical Report, UWA-CSSE-06-001, 2006.

[9] I. F. Akyildiz, Y. Sankarasubramaniam, W. Su, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, pp. 393–422, 2002.

[10] L. B. Ruiz and J. M. Nogueira, "MANNA: Management architecture for wireless sensor networks," Communications Magazine, Vol. 41, No. 2, pp. 116–125, 2003.

[11] H. Song, D. Kim, K. Lee, and J. Sung, "UPnP-based sensor network management architecture," Proceedings of the second International Conference on Mobile Computing and Ubiquitous Networking, pp. 85–92, 2005.

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishan, "Energy efficient communication protocol for wireless microsensors networks," Proceedings of the Hawaii International Conference on System Science, pp. 10–17, 2000.

[13] Y. Xu, "Geography-informed energy conservation for ad hoc routing," Mobicom'01, pp. 203–212, 2001.

[14] W. L. Lee, A. Datta, and R. Cardell-Oliver, "WinMS: Wireless sensor network-management system," CSSE Technical Report, UWA- CSSE-06-001, 2006.

[15] N. Ramanathan, K. Chang, R. Kapur, L. Girod, and E. Kohler, "Sympathy for the sensor network debugger," Centre for Embedded Network Sensing, pp. 98, 2005.

[16] S. Madden, J. Hellerstein, and W. Hong, "TinyDB: In-network query processing in TinyOS," ACM Transactions on Database Systems, pp. 122–173, 2003.

[17] Touron, "Crossbow: moteview interface," Crossbow, 2005, http://www.xbow.com/Technology/UserInterface. aspx.

[18] R. Tynan, D. Marsh, D. O'Kane, and G. M. P. O'Hare, "Intelligent agents for wireless sensor networks," AAMAS 2005: 1179–1180.

[19] T. H. Kim and S. Hong, "Sensor network management

protocol for state-driven execution environment," Proceedings of the International Conference on Ubiquitous Computing, pp. 197–199, 2003.

[20] J. Wan, S. Eisenman, A. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multi-radio virtual sinks in sensor networks," Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, pp. 116–129, 2005.

[21] Zhang, "Network management in wireless sensor networks," 2001. http://www.csse.uwa.edu.au/~winnie/Network_Management _in_WSNs_.pdf.

[22] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, pp. 100–111, 2002.

[23] C. L. Fok and G. C. Roman, "Mobile agent middleware for sensor networks: An application case study," Proceedings of the 4th International Conference on Information Processing in Sensor Networks, pp. 382–387, 2005.

[24] S. Eduardo, G. Germano, and V. Glauco, "A message-oriented middleware for sensor networks," Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing, pp. 127–134, 2004.

[25] D. Georgoulas and K. Blow, "In-Motes: An intelligent agent based middleware for wireless sensor networks," Best Student Paper Award, Proceedings of the 5th WSEAS International Conference SEPADS 06, pp. 225–231, 2006.

[26] Welsh *et al*, "Wireless medical sensor networks in emergency response: Implementation and pilot results," IEEE International Conference on Technologies for Homeland Security.

[27] Holler, "Wireless sensing vineyards," 2008, http://camalie.com/WirelessSensing/WirelessSensors.htm.

[28] D. Georgoulas and K. Blow, "Making motes intelligent: An agent based approach to wireless sensor networks," In WSEAS on Communications Journal, pp. 515–522, 2006.

[29] D. Georgoulas and K. Blow, "In-Motes bins: A real time application for environmental monitoring in wireless sensor networks," Proceedings of the 9th IEEE/IFIP International Conference on Mobile and Wireless Communications Networks, pp. 21–26, 2007.

[30] D. Georgoulas and K. Blow, "Intelligent mobile agent middleware for wireless sensor networks: A real time application case study," AICT 2008, in Proceedings of 4th Advanced International Conference on Telecommunications: Programmable networks, active networks and mobile agents, protocol & standards, Athens, pp. 35–43, 2008.

Wireless Sensor Network, **2009, 1, 268-275**
doi:10.4236/wsn.2009.14033 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Scientific Research

Performance Improvement of the DSRC System Using a Novel S and Π-Decision Demapper

Jeich MAR, Chi-Cheng KUO
Department of Communications Engineering, Yuan Ze University, Taiwan, China
Email: *eejmar@saturn.yzu.edu.tw*
Received March 31, 2009; *revised June* 20, 2009; *accepted June* 23, 2009

Abstract

Based on the constellation diagram of the different modulations, a novel S and Π-decision rule is designed for the analog demapper of the orthogonal frequency-division multiplexing (OFDM) systems. The dedicated short-range communications (DSRC) system is chosen as an OFDM platform to compare the performance among the proposed S and Π-decision decoder, hard-decision and soft-decision decoders. Simulation results demonstrate that both the complexity and performance of S and Π-decision demapper used for M-ary quadrature amplitude modulation (QAM) OFDM system can be greatly improved. The number of decisions between the received symbol and constellation points can be simplified to look up table $\log_2 M$ times for M-ary QAM OFDM system.

Keywords: S and Π-decision Rule, Analog Demapper, DSRC System, OFDM

1. Introduction

In the traditional digital communication system where the digital Viterbi decoder is used, the maximum likelihood decision rule is applied to both the demapper and digital hard-decision or soft-decision Viterbi decoder. The soft-decision decoder is the recommended scheme to be used in the digital Viterbi decoder because it provides a coding gain over the hard-decision decoder [1,2]. A simplified algorithm of the soft-decision Viterbi decoder for the 16-quadrature amplitude modulation (QAM) and 64-QAM constellations was presented in [3], which allows the complexity of the demapper to be maintained at almost the same level for all the possible modes of HIPERLAN/2. In [4], it presented that for M-ary QAM systems the complexity of the demapper in a soft-decision Viterbi decoder used for bit-interleaved coded modulation can be significantly lowered without compromising the performance. Four types of analog-input Viterbi decoders are described and compared in [5], where the analog-to-digital converter (A/D) converter is included as part of a digital Viterbi decoder. The analog circuit flaws of the previously used add-compare-select (ACS) chips are included in the comparison. It concludes the analog Viterbi decoder is able to outperform the digital Viterbi decoder, as well as achieve 3-bit or higher decoding resolution. In this paper, we propose a new S and Π-decision decoder, where the S and Π-decision rules are designed for the analog demapper of the orthogonal frequency-division multiplexing (OFDM) systems. The ACS and path memory (PM) modules, which are parts of the analog Viterbi decoder introduced in [6], are used to perform the Viterbi decoding process. The proposed S and Π-decision demapper combined with the digital Viterbi decoder is another alternative to using the S and Π-decision decoder. Because analog Viterbi decoder outperforms digital Viterbi decoder and analog very-large-scale integration (VLSI) implementation is in general more area and power efficient than digital implementation, the performance comparison of the S and Π-decision decoder using the digital Viterbi decoder is not included in the paper.

The dedicated short-range communications (DSRC) system [7], which employs OFDM technique, provides wireless communications over a short distance between the roadside and high-speed mobile radio units or between high-speed vehicles. The DSRC system will work in a mobile environment with time-varying characteristics. The time-varying fading effect of the DSRC system may not be effectively compensated by using the long symbol training and pilot-based frequency synchronizer [8]. The combined interleaving and convolution coding, which provides the time diversity function, may further improve the performance of the DSRC system. We choose DSRC system as a basis for the performance

comparison among different decoders. Based on the analog-input Viterbi decoder, the coding gain of the DSRC system achieved by replacing the hard-decision and soft-decision decoders with the proposed S and Π-decision decoder will be confirmed with the simulations.

The rest of this paper is organized as follows. In Section II, several channel decoding schemes are described. The proposed S and Π-decision decoder is depicted in detail for the binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), 16-QAM and 64-QAM OFDM signals of the DSRC systems. Section III briefly describes the base band model of the DSRC system, which is used to compare the bit error rate (BER) performance of the DSRC system for the different decoders. Simulation results are given in Section IV, which show the coding gain achieved by the proposed S and Π-decision decoder compared to the hard-decision and soft-decision decoders. Finally, conclusions are drawn in Section V.

2. S and Decision Decoder for the OFDM Systems

In the traditional digital communication system, the digital Viterbi decoding uses a maximum likelihood rule which is ideal for an additive white Gaussian noise (AWGN) channel. For a hard-decision Viterbi decoder, the samples matching to a single bit of a code word are quantized to the two levels zero and one, a decision is made as whether each transmitted bit in a code word is zero or one. The coding gain of the soft-decision Viterbi decoder for the hard-decision Viterbi decoder in Rayleigh fading channel increases to about 2dB [1]. A four-level discrete symmetric channel model [2] is used for the soft-decision decoder. The demapper assigns one of four values to each received signal. The path metrics in the Viterbi algorithm are calculated by weighting the square of the Hamming distance between the soft-decision and the reference value. The four-level soft-decision Viterbi decoder is almost exactly as shown for the hard-decision case with the only difference being the increased number of path metrics.

The block diagram of the proposed S and Π-decision decoder for the OFDM system is shown in Figure 1, where it consists of a S and Π-demapper and an analog Viterbi decoder. The S and Π-decision decoder is a non-uniform infinite-level quantization decoder. The S and Π-decision demapper assigns an analog complex value to the analog Viterbi decoder for each received signal $z(y)$ according to a combination of S and Π functions, as shown in Figure 2. The Π function [9], as shown in Figure 2(a), is defined as follows:

$$\Pi(z(y);R,M)=\begin{cases} S(z(y);M-R,M-R/2,M) & for\ z(y)\le M \\ 1-S(z(y);M,M+R/2,M+R) & for\ z(y)\ge M \end{cases} \tag{1}$$

where $z(y)$ represents either $\hat{I}^m(y)$ or $\hat{Q}^m(y)$ and y is the received signal after channel compensation. The constellation decoder estimate the m^{th} symbol gained through the received signal after channel compensation, $\hat{Y}^m(y)=\hat{I}^m(y)+j\hat{Q}^m(y)$, can be found in the signal space diagrams [10] for BPSK, QPSK, 16-QAM and 64-QAM, respectively, where the values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$ are serially decoded according to the modulation type. The Π-function goes to zero at the points

$$z(y)=M\pm R \tag{2}$$

while the Π-function goes to 0.5 at the crossover points

$$z(y)=M\pm\frac{R}{2} \tag{3}$$

Notice the parameter R is now equal to one, which is the total width at the crossover points; parameter M is now equal to zero, which is the middle point of the Π-function. The S-function, as shown in Figure 2(b), is defined as follows:

$$S(z(y);\alpha,\beta,\gamma)=\begin{cases} 0 & for\ z(y)\le\alpha \\ 2(\frac{z(y)-\alpha}{\gamma-\alpha})^2 & for\ \alpha\le z(y)\le\beta \\ 1-2(\frac{z(y)-\gamma}{\gamma-\alpha})^2 & for\ \beta\le z(y)\le\gamma \\ 1 & for\ z(y)\ge\gamma \end{cases} \tag{4}$$

For BPSK modulation, the value of $\hat{I}^m(y)$ is in the interval of (-1,1) of constellation diagram. From BPSK constellation diagram, the original one bit binary data (b_0) is decided as $\hat{b}_0$ using the S-decision rule. Using (4) for α=-1, β=0 and γ=1, the values of $\hat{b}_0$ are produced as follows: $\hat{b}_0$ =-1 for $z(y)\leqq$-1; $\hat{b}_0$ =1 for $z(y)\geqq$1; $\hat{b}_0=S(z(y))$ for -1<$z(y)$<1.

For QPSK, the values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$ in the constellation diagram are found in the intervals (-1, 1). The original two-bit vector $\vec{b}^{(2)}$ = (b_0, b_1) is also estimated using the S-function as a decision rule. The S-decision rule in (4) for α=-1, β=0 and γ=1 is used to determine $\hat{b}_0$ from the received I-channel signal part $\hat{I}^m(y)$ and determine $\hat{b}_1$ from the Q-channel signal part $\hat{Q}^m(y)$.

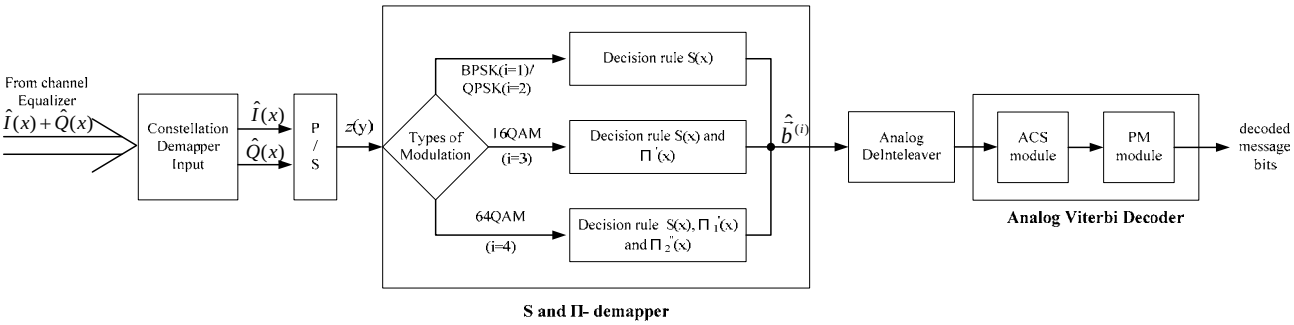


Figure 1. Block diagram of the *S* and Π-decision decoder.

The constellation diagram of 16-QAM is shown in Figure 3, where the values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$ for 16-QAM modulation are found in the intervals of (-3, -1, 1, 3), respectively. The message points in each quadrant are assigned with Gray-encoded four-bit vector $\vec{b}^{(3)} = (b_0\ b_1\ b_2\ b_3)$. The first two bits ($b_0\ b_1$) and last two bits ($b_2\ b_3$) in $\vec{b}^{(3)}$ are transmitted in I and Q-channel, respectively. Both first two bits (boldface) of the I-channel from left to right message points and last two bits (Normal) of the Q-channel from bottom to top message points have the same 8-bit pattern 00 01 11 10 in Figure 3. The first four odd bits are 0 0 1 1 and the second four even bits are 0 1 1 0 can be estimated by using the S and Π'-decision rule, respectively, as shown in Figure 4. The Π'-decision rule as shown in Figure 4(a) is defined as

$$\Pi'(z(y)) = \begin{cases} 0 & z(y) \le -3 \\ S(z(y);-3,-2,-1) & -3 \le z(y) \le -1 \\ 1 & -1 \le z(y) \le 1 \\ 1-S(z(y);1,2,3) & 1 \le z(y) \le 3 \\ 0 & z(y) \ge 3 \end{cases} \quad (5)$$

where the S-function is defined in (4). The first two bits ($\hat{b}_0\ \ \hat{b}_1$) and the last two bits ($\hat{b}_2\ \ \hat{b}_3$) of the demapper output for each message point are estimated from the values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$, respectively. The S-decision rule is used to determine $\hat{b}_0$ and $\hat{b}_2$ and the Π'-decision rule is used to determine $\hat{b}_1$ and $\hat{b}_3$. The number of decision needed to obtain S and Π-decision demapper output in 16-QAM OFDM system is four.

The same design principle of the S and Π-decision demapper used for 16-QAM is applied for 64-QAM. The values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$ in the constellation diagram for 64-QAM modulation are found in the intervals of (-7, -5,-3, -1, 1, 3, 5, 7), respectively. Similarly, the six-bit vector $\hat{\vec{b}}^{(4)} = (\hat{b}_0\ \ \hat{b}_1\ \ \hat{b}_2\ \ \hat{b}_3\ \ \hat{b}_4\ \ \hat{b}_5)$ for each message point of 64-QAM modulation is de

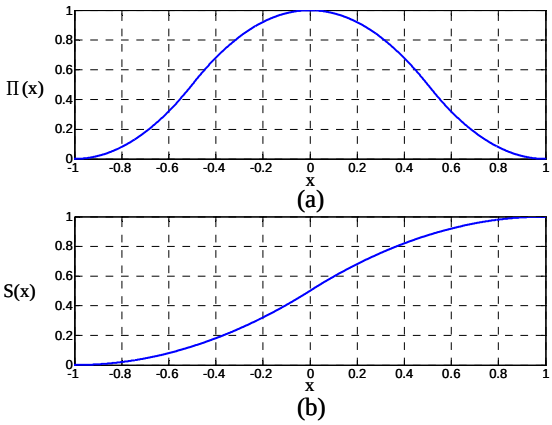


Figure 2. (a) Π-decision rule; (b) *S*-decision rule.

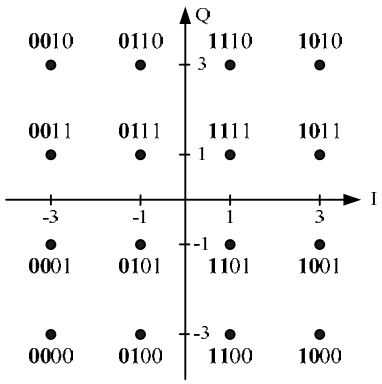


Figure 3. The constellation diagram of 16-QAM.

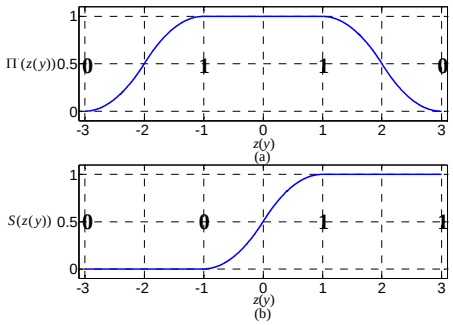


Figure 4. (a) Π'-decision rule; (b) *S*-decision rule for 16-QAM.

termined using the $\Pi_1^{"}$, $\Pi_2^{"}$ and S-decision rules, which are designed with 24-bit pattern 000 001 011 010 110 111 101 100, which is generated from first three bits from left to right message points and last three bits from bottom to top message points, for both I and Q-channels. The first 8-bit pattern 0 0 0 0 1 1 1 1 can be determined from the S-decision rules as shown in Figure 5(a). The second middle 8-bit pattern 0 0 1 1 1 1 0 0 can be estimated by using the $\Pi_1^{"}$-decision function which0 is defined as

$$\Pi_1^{"}(z(y)) = \begin{cases} 0 & z(y) \le -5 \\ S(z(y);-5,-4,-3) & -5 \le z(y) \le -3 \\ 1 & -3 \le z(y) \le 3 \\ 1-S(z(y);3,4,5) & 3 \le z(y) \le 5 \\ 0 & z(y) \ge 5 \end{cases} \quad (6)$$

The third 8-bit pattern 0 1 1 0 0 1 1 0 can be estimated from the $\Pi_2^{"}$-decision rule as shown in Figure 5(b), which is defined as

$$\Pi_2^{"}(z(y)) = \begin{cases} 0 & z(y) \le -7 \\ S(z(y);-7,-6,-5) & -7 \le z(y) \le -5 \\ 1 & -5 \le z(y) \le 3 \\ 1-S(z(y);-3,-2,-1) & -3 \le z(y) \le -1 \\ 0 & -1 \le z(y) \le 1 \\ S(z(y);1,2,3) & 1 \le z(y) \le 3 \\ 1 & 3 \le z(y) \le 5 \\ 1-S(z(y);5,6,7) & 5 \le z(y) \le 7 \\ 0 & 7 \le z(y) \end{cases} \quad (7)$$

The first three bits ($\hat{b}_0$ $\hat{b}_1$ $\hat{b}_2$) and the last three bits ($\hat{b}_3$ $\hat{b}_4$ $\hat{b}_5$) of each message point are estimated from the values of $\hat{I}^m(y)$ and $\hat{Q}^m(y)$, respectively. The S-decision rule is used to determine $\hat{b}_0$ and $\hat{b}_3$, the $\Pi_1^{"}$-decision rule is used to determine $\hat{b}_1$ and $\hat{b}_4$ and the $\Pi_2^{"}$-decision rule is used to determine $\hat{b}_2$ and $\hat{b}_5$. The number of decisions needed for S and Π–demapper in 64-QAM OFDM system is six. It is concluded that for Gray encoded M-ary QAM OFDM systems, the number of decisions for analog S and Π-demapper can be reduced to $\log_2 M$. If the S, Π, Π' and $\Pi^{"}$ decision rules are calculated and stored in the table, each demapper decision can be simplified as to look up table. The analog Viterbi decoder designed in Figure 1 is a 64-state decoder with a trace back length of 24, 48, 96 and 144 for BPSK, QPSK, 16-QAM and 64-QAM, respectively. The analog complex values of the constellation decoding vectors ($\hat{\bar{b}}^{(i)}$) gained through the combinational S and Π-decision rules are input to analog deinterleaver and analog Viterbi decoder in turn. The analog deinterleaver [11] permutes the analog demapper output according to the switching order performed in the interleaver. The analog Viterbi decoder consists of the analog ACS module and a digital PM module [6]. ACS module performs the calculation of the analog path metrics. The transmitted message bits are decoded by PM module using the trace back through the trellis architecture. The decoding algorithm and the sequence control for analog Viterbi decoder remain identical with the digital Viterbi decoder.

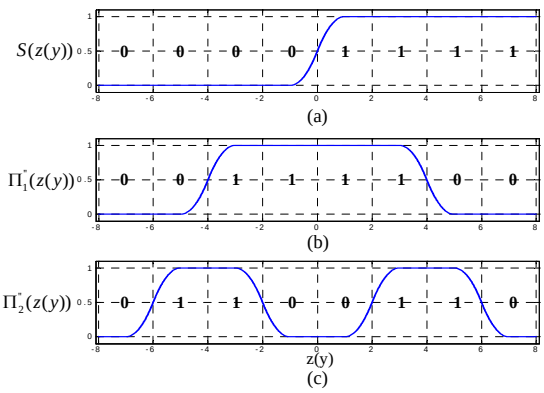


Figure 5. (a) *S*-decision rule; (b) $\Pi_1^{"}$ -decision rule (c) $\Pi_2^{"}$-decision rule for 64-QAM.

3. Base Band Model of the DSRC System

The block diagram of the DSRC system is shown in Figure 6. The protocol data unit (PDU) trains are applied to the physical layer for transmission. A 127 pseudorandom sequence is used to scramble the data out of the binary sequence before the convolutional encoding. The purpose of the scrambler is to prevent a long sequence of 1s or 0s to aid the timing recovery at the receiver. The generator polynomial [10] of the pseudorandom sequence is

$$g(D) = D + D^4 + D^7 \quad (8)$$

where D is the unit-delay. The different initialization value is decided by the first 7 bits of each PDU train. The scrambled data sequence is encoded with a rate 1/2 convolutional code with the generator polynomial $g^{(1)}(D)$ for the upper connection and $g^{(2)}(D)$ for the lower connection as follows:

$$g^{(1)}(D) = 1 + D^2 + D^3 + D^5 + D^6 \quad (9)$$

$$g^{(2)}(D) = 1 + D + D^2 + D^3 + D^6 \quad (10)$$

where D is the unit-delay for convolutional codes and the lowest-order term in the polynomial matches the input stage of the shift register. The puncturing pattern [10] is

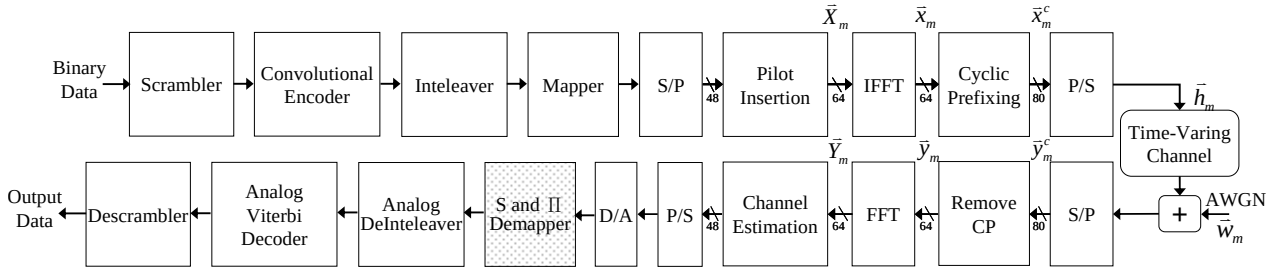


Figure 6. The block diagram of the DSRC base band model.

used to make a rate 3/4 convolutional code from the rate 1/2 convolutional code. A convolutional code may correct many well-spaced errors, while being unable to handle an error burst introduced by the fading channel. The block interleaver or deinterleaver pair [10] applied to the DSRC system can spread the burst error across onto nonadjacent subcarriers and mapped alternately onto less and more significant bits of the constellation. After processing the scrambler, convolution encoder and interleaver, followed by mapping to BPSK or QAM constellation points, the transmitting data stream is divided into several parallel bit streams. An OFDM signal is built using an 64-points inverse fast Fourier transform (IFFT). The input vector to the IFFT is given as

$$\vec{X}_m = [X_{m,0}, X_{m,1}, ..., X_{m,N-1}]^T \tag{11}$$

where $X_{m,k}$ represents the k^{th} subcarrier of the m^{th} OFDM symbol and N is 64 in the DSRC system. The IFFT output signal vector is

$$\vec{x}_m = [x_{m,0}, x_{m,1}, ..., x_{m,N-1}]^T \tag{12}$$

where $x_{m,n}$ is the n^{th} sample point of the m^{th} OFDM symbol.

$$x_{m,n} = \frac{1}{N}\sum_{k=0}^{N-1} X_{m,k} \exp(j2\pi nk) = \text{IFFT}\left\{X_{m,k}\right\} \tag{13}$$

The cyclic prefixes (CP), which are produced with the copies of the last parts of the OFDM symbol, are prepended to the front of each vector $\vec{x}_m$. The cyclic prefixing output signal vector is represented as

$$\begin{aligned}\vec{x}_m^c &= [x_{m,0}^c, x_{m,1}^c, ..., x_{m,N+q-1}^c]^T \\ &= [x_{m,N-q}, x_{m,N-q+1}, ..., x_{m,N-1}, x_{m,0}, x_{m,1}, ..., x_{m,N-1}]^T\end{aligned} \tag{14}$$

where $x_{m,n}^c$ is the n^{th} sample point of the m^{th} OFDM symbol and q is the length of the CP. Therefore, the received signal vector is given by

$$\vec{y}_m^c = \vec{x}_m^c \otimes \vec{h}_m + \vec{w}_m = [y_{m,0}^c, y_{m,1}^c, ..., y_{m,N+q-1}^c]^T \tag{15}$$

Where $\otimes$ stands for linear convolution, $\vec{h}_m$ and $\vec{w}_m$ are the channel impulse response vector and the additive white Gaussian noise (AWGN) vector for the m^{th} OFDM symbol, respectively. $y_{m,n}^c$ is the n^{th} sample point of the m^{th} OFDM symbol in the m^{th} received signal vector $\vec{y}_m^c$. The channel impulse response vector $\vec{h}_m = [h_{m,0}, h_{m,1}, ..., h_{m,N-1}]^T$ can be represented by [12]:

$$h_{m,n} = \sum_{i=0}^{\gamma-1} h_i^m e^{j\frac{2\pi}{N} f_{D_i} Tn} \delta(\lambda - \tau_i),\ 0 \le n \le N-1 \tag{16}$$

where h_i^m is the complex impulse response of the m^{th} OFDM symbol in the i^{th} path; f_{Di} is the i^{th}-path Doppler frequency shift, which may cause intercarrier interference (ICI) for the received signals; T is the sample period; λ is the delay spread index; and τ_i is the i^{th}-path delay time normalized by sampling time.

After removing the CP, the received signal vector $\vec{y}_m$ is

$$\vec{y}_m = [y_{m,0}, y_{m,1}, ..., y_{m,N-1}]^T = [y_{m,q}^c, y_{m,q+1}^c, ..., y_{m,N+q-1}^c]^T \tag{17}$$

where $y_{m,n}$ is the n^{th} sample point of the m^{th} OFDM symbol. The demodulated received signal vector is

$$\vec{Y}_m = [Y_{m,0}, Y_{m,1}, ..., Y_{m,N-1}]^T \tag{18}$$

where

$$Y_{m,k} = \sum_{n=0}^{N-1} y_{m,n} e^{-j\frac{2\pi}{N}nk} = \text{FFT}\left\{y_{m,n}\right\} \tag{19}$$

Suppose the guard interval is longer than the length of the channel impulse response, that is, there is no intersymbol interference between the OFDM symbols, the demodulated sample vector $\vec{Y}_m$ can then be represented as [13]

$$\vec{Y}_m = \vec{X}_m \vec{H}_m + \vec{I}_m + \vec{W}_m \tag{20}$$

$$\vec{H}_m = [H_{m,0}, H_{m,1}, ..., H_{m,N-1}]^T \tag{21}$$

$$\vec{I}_m = [I_{m,0}, I_{m,1}, ..., I_{m,N-1}]^T \tag{22}$$

$$H_{m,k} = \sum_{i=0}^{\gamma-1} h_i^m e^{j\pi f_{D_i} T} \frac{\sin(\pi f_{D_i} T)}{\pi f_{D_i} T} e^{j\frac{2\pi\tau_i}{N}l},\ 0 \le k \le N-1 \tag{23}$$

$$I_{m,k} = \frac{1}{N}\sum_{i=0}^{\gamma-1}\sum_{\substack{K=0\\K\neq k}}^{N-1} h_i^m X_{m,k} \frac{1-e^{j2\pi(f_{D_i}-k+K)}}{1-e^{j\frac{2\pi}{N}(f_{D_i}-k+K)}} e^{-j\frac{2\pi\tau_i}{N}K},\ 0 \le k \le N-1 \tag{24}$$

where $\vec{W}_m = FFT\{\vec{w}_m\}$. $H_{m,k}$ is recognized as the accurate channel frequency response at the k^{th} subcarrier of the m^{th} OFDM symbol, which is independent of the transmitted signals $X_{m,k}$. $I_{m,k}$ is the ICI part of the received signal at the k^{th} subcarrier of the m^{th} OFDM symbol, depending on the signal values $X_{m,k}$ modulated on all subcarriers.

On the highway, the maximum vehicle speed is 200 km/hr. The DSRC system needs more robust frequency and phase synchronization technology. Four uniform pilot subcarriers, which are inserted in the positions of the 6th, 20th, 34th, and 48th subcarriers for each of the transmitted DSRC data symbols, are applied for the DSRC receiver to estimate the frequency and track the phase of the received signals. A pilot-based frequency synchronizer mechanism including least squares estimation (LSE) and interpolation is used for equalizing the pilot signal-aided frequency and phase synchronization [14].

4. Simulations

Packet detection, timing synchronization and coarse frequency offset estimation of the DSRC receiver are performed according to the algorithms provided in [15]. The simulations focus on comparing the DSRC system performance among the proposed S and Π-decision decoder, hard-decision and soft-decision decoders. The DSRC system is specified in the 5.85-5.925GHz ITS radio services band. In a DSRC system, one frame has 100 OFDM symbols [7]. The total number of subcarriers is 64 including four uniformly distributed pilot subcarriers and 12 guard subcarriers. According to the IEEE 802.11p standard, the minimum input signal to noise ratio values of the DSRC receiver for BPSK OFDM (3Mbps), QPSK OFDM (6Mbps), 16-QAM OFDM (12Mbps), 16-QAM OFDM (18Mbps), 64-QAM OFDM (24Mbps) and 64-QAM OFDM (27Mbps) modulations are 9, 12, 17, 24, 25 and 27dB respectively, which are used as a basis for evaluating the receiver performance. The 3Mbps, 6Mbps, and 12Mbps data transmission rates are made by using 1/2-rate convolutional code. The 18Mbps and 27Mbps data transmission rates are produced by using 3/4-rate convolutional code. The 24Mbps data transmission rate is produced by 2/3-rate convolutional code. The quantization loss for the digital decoding is also considered in the decoding BERs of Figures 9 and 12.

Based on analysis results in [16], the quantization loss for convolutional decoding relative to the continuous case for two-level, four-level and eight-level digital Viterbi decoders are evaluated as 7.73, 1.78, and 0.52 dB with the 1/2-rate convolutional code; 11.43, 2.19 and 0.55 dB with the 3/4-rate convolutional code. These losses remain roughly constant across the range of BER plotted. Referring to the SPICE simulation results in [6], the loss of real analog Viterbi decoder with 0.002 ACS noise and 10% comparator offset is less than 0.2 dB roughly compared with the ideal analog Viterbi decoders. Therefore, the ACS noise and the comparator offset will not be considered in the following simulations.

In the DSRC system, the coherence time T_c is calculated by

$$T_c = \frac{0.423}{f_D} = \frac{0.423\lambda}{v_D} = \frac{0.423c}{v_D f_c} = 645.3\mu\text{sec} \tag{25}$$

where f_D=f_{Di} for i=1 and 2. The maximum Doppler shift is given by $f_D = v_D/\lambda$, where v_D is the maximum vehicle speed, f_c is the carrier frequency, and c is the velocity of light. Jakes' channel model [17] is used to produce a time- varying Rayleigh fading channel simulator. The effects of AWGN and carrier frequency shift are also considered in the DSRC channel. Simulations are carried out for the vehicle speed v_D = 200 km/hr, delay spread τ = 200 nsec, 100 data symbols and different decision Viterbi decoder.

Figure 7 shows when the delay spread exceeds 150 *n*sec, the severer frequency selective channel fading will be caused by reducing coherent bandwidth. The BER performance of the DSRC receiver with BPSK OFDM (3Mbps), QPSK OFDM (6Mbps), 16-QAM OFDM (12Mbps), 16-QAM OFDM (18Mbps), 64-QAM OFDM (24Mbps) and 64-QAM OFDM (27Mbps) modulations are shown in Figure 8–12 respectively. Figure 8 shows the BER of QPSK OFDM modulations for Viterbi decoders using three different decision rules are reduced to

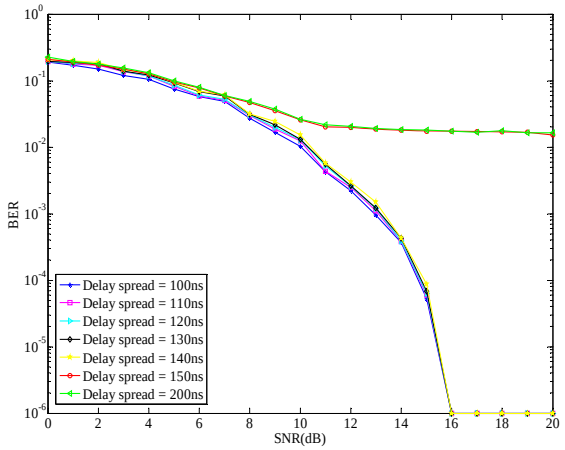


Figure 7. BERs of the DSRC system using pilot subcarrier-aided equalizer in different delay spread for 16-QAM OFDM modulations.

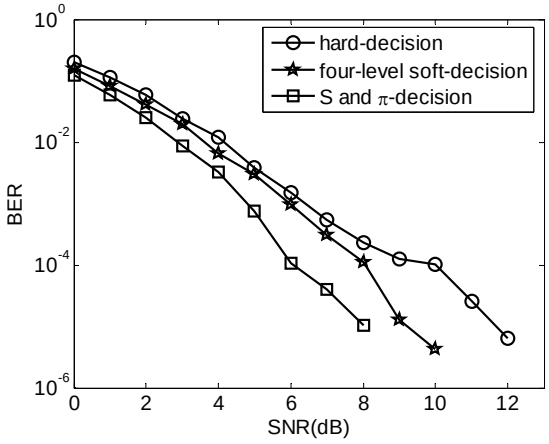


Figure 8. Comparisons of BERs of the QPSK DSRC system (6 Mbits/sec; 200km/h; code-rate is 1/2) in terms of the different decision decoders.

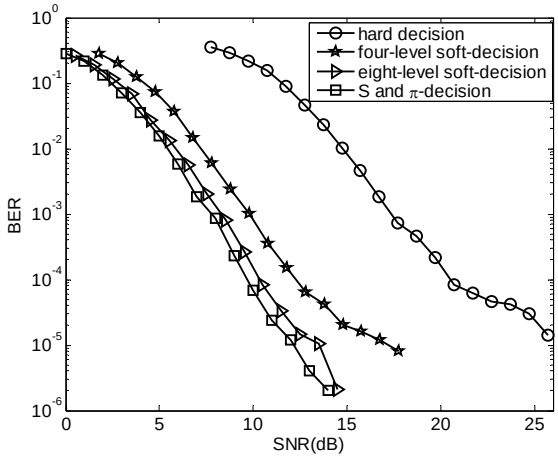


Figure 9. Comparisons of BERs of the 16-QAM DSRC system (12 Mbits/sec; 200km/h; code-rate is 1/2) in terms of the different decision decoders with quantization loss.

less than 10^{-5} at the minimum signal-to-noise ratio (SNR), which meets the requirements specified in the IEEE 802.11p standard.

When the quantization loss for the convolutional decoding is considered, the BER of 16-QAM OFDM DSRC system is shown in Figure 9, where using a hard- decision Viterbi decoder is higher than 10^{-5} at the minimum SNR (17dB) for the case of the 12Mbps data transmission rate. The 16-QAM OFDM DSRC system using the four-level and eight-level soft-decision and analog Viterbi decoders will be reduced to less than 10^{-5} at the minimum SNR (17dB), which meet the requirements specified in the IEEE802.11p standard. It is noted the S and Π-decision decoders results in a coding gain of 1.5 dB and 5.2dB compared to the eight-level and four-level soft-decision decoders, respectively, when the quantization loss for the convolutional decoding is considered. Figure 10 shows the BER of 16-QAM DSRC system using the hard-decision decoder cannot be lower than 10^{-5}, when the data transmission rate increases to 18Mbps. The hard-decision curve has a floor that is generated by the hard-decision loss under the two-ray Rayleigh fading channel environment. The 16-QAM DSRC system using both the soft-decision decoder and the S and Π-decision decoders will reduce the BER to less than 10^{-5} at the minimum SNR (24dB), which meets the requirements specified in the IEEE802.11p standard. The S and Π-decision decoder has a 1.5dB coding gain compared with the four-level soft-decision decoder.

Figure 11 shows the BER of 64-QAM DSRC system under the conditions of 18 Mbits/sec data rate and 200km/h vehicle speed cannot be lower than 10^{-5} for three different Viterbi decoders. All three BER curves have the floors that are caused by the high-order 64-QAM modulation DSRC system operated under the fast fading channel. The BER of 64-QAM DSRC system is shown in Figure 12, where the quantization loss for the

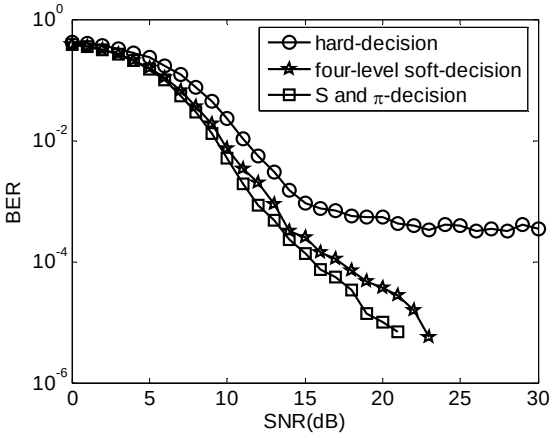


Figure 10. Comparisons of BERs of the 16-QAM DSRC system (18 Mbits/sec; 200km/h; code-rate is 3/4) in terms of the different decision decoders.

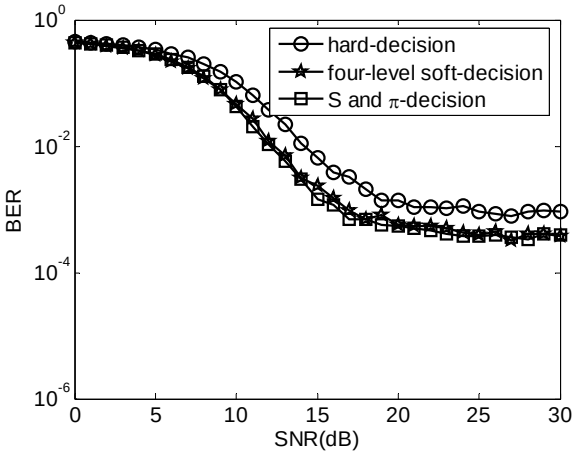


Figure 11. Comparisons of BERs of the 64-QAM DSRC system (24 Mbits/sec; 200km/h; code-rate is 2/3) in terms of the different decision decoders.

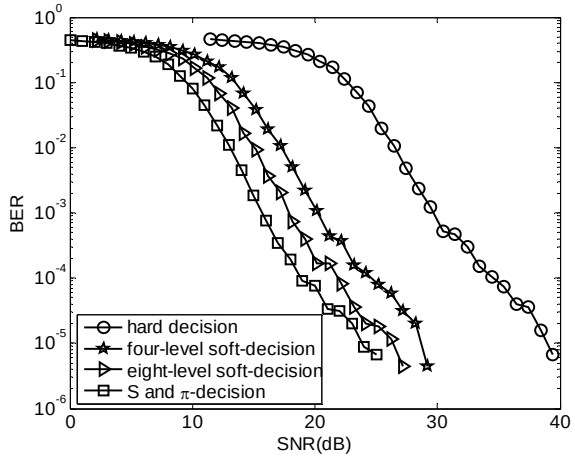


Figure 12. Comparisons of BERs of the 64-QAM DSRC system (27 Mbits/sec; 120km/h; code-rate is 3/4) in terms of the different decision decoders with quantization loss.

convolutional decoding is considered and the vehicle speed reduces to v_m = 120 km/hr. It showed the 64-QAM DSRC system using three different decoders can be reduced to less than 10^{-5} at the minimum SNR (27dB), which meets the requirements specified in IEEE802.11p standard. The S and Π-decision decoder results in a coding gain of 2.5 dB, 5dB and 15 dB compared to the eight-level soft-decision, four-level soft-decision and hard-decision Viterbi decoders, respectively, when the quantization loss for the digital Viterbi decoding is considered.

5. Conclusions

A new S and Π-decision rule is proposed for the analog demapper of the OFDM system. The DSRC system is chosen as an OFDM platform to compare the performance of the S and Π-decision decoder, which consists of an S and Π-decision demapper, analog deinterleaver and an analog Viterbi decoder, with hard-decision and soft-decision decoders. Simulation results show the coding gain of the S and Π-decision decoder relative to four-level and eight-level soft-decision decoders are evaluated as 5.4dB and 1.5 dB, respectively, with the 16-QAM DSRC system; and 5dB and 2.5 dB with the 64-QAM DSRC system when the quantization loss for the convolution decoding is considered. Each analog demapper output can be determined by looking-up S and Π table $\log_2 M$ times for M-ary QAM OFDM systems. Many other applications related to OFDM with the proposed S and Π-decision decoder are possible.

6. Acknowledgement

The authors would like to acknowledge gratefully the research grants from National Science Council, Taiwan, NSC 96-2219-E-155-005.

7. References

[1] K. M. Lee, D. S. Han, and K. B. Kim, "Performance of the Viterbi decoder for DVB-T in Rayleigh fading channels," IEEE Trans. Consumer Electron., Vol. 44, pp. 994–1000, August 1998.

[2] S. B. Wicker, "Error control systems for digital communication and storage," New Jersey, Prentice-Hall, Inc., 1995.

[3] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," in Proc. IEEE Int. Conf. Communications, Vol. 2, pp. 664–668, April 2002.

[4] E. Akay and E. Ayanoglu, "Low complexity decoding of bit-interleaved coded Modulation for M-ary QAM," in Proc. IEEE Int. Conf. Communications, Vol. 2, pp. 901–905, June 2004.

[5] K. He and G. Cauwenberghs, "Performance of analog Viterbi decoding," 42nd Midwest Symposium on Circuits and Systems, Vol. 1, August 1999.

[6] K. He and G. Cauwenberghs, "Itergrated 64-state parallel analog Viterbi decoder," ISCAS 2000, Vol. 4, May 2000.

[7] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle System–5Ghz Band Dedicated Short Range Communications (DSRC) Medium Access Control (Mac) and Physical Layer (PHY) Specifications, IEEE802.11p, December 2005.

[8] S. Coleri, M. Ergen, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," IEEE Trans. Broadcast., Vol. 48, pp. 223–229, September 2002.

[9] J. Giarratano and G. Riley, Expert Systems, 2nd. Edition, PWS Publishing Company, 1994.

[10] J. Heiskala and J. Terry, OFDM Wireless LANs: A Theoretical and Practical Guide, Sams, 2001.

[11] V. C. Gaudet, R. J. Gaudet, and P. G Gulak, "Programmable interleaver design for analog iterative decoders," IEEE Trans. Circuits Syst. II, Vol. 49, pp. 457–464, July 2002.

[12] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, "Statistical and adaptive signal processing," McGraw-Hill, Education, 2000.

[13] Y. Zhao and A. Huang, "A novel channel estimation method for OFDM mobile communication systems based on pilot signals and transform-domain processing," in Proc. IEEE VTC, Vol. 3, pp. 2089–2093, May 1997.

[14] J. Rinnie and M. Renfors, "Pilot spacing in orthogonal frequency division multiplexing systems on practical channels," IEEE Trans. Consumer Electron., Vol. 42, No. 4, pp. 959–962, November 1996.

[15] IEEE 802.11a, IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.

[16] M. R. G. Butler and A. R. Nix, "Quantization loss for convolutional decoding in Rayleigh-fading channels," IEEE Commun. Lett., Vol. 7, No. 9, pp. 446–448, September 2003.

[17] W. C. Jakes, Microwave Mobile Communications, New York: IEEE Press, 1974.

Wireless Sensor Network, **2009, 1, 276-283**
doi:10.4236/wsn.2009.14034 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Real-Time Automatic ECG Diagnosis Method Dedicated to Pervasive Cardiac Care

Haiying ZHOU[1], Kun-Mean HOU[2], Decheng ZUO[1]
[1]*School of Computer Science & Technology, Harbin Institute of Technology, Harbin, China*
[2]*LIMOS Laboratory UMR* 6158 *CNRS, University of Blaise Pascal, Clermont-Ferrand, France*
Email: {*haiyingzhou, zdc*}*@hit.edu.cn, kun-mean.hou@isima.fr*
Received May 1, 2009; *revised May* 25, 2009; *accepted May* 31, 2009

Abstract

Recent developments of the wireless sensor network will revolutionize the way of remote monitoring in different domains such as smart home and smart care, particularly remote cardiac care. Thus, it is challenging to propose an energy efficient technique for automatic ECG diagnosis (AED) to be embedded into the wireless sensor. Due to the high resource requirements, classical AED methods are unsuitable for pervasive cardiac care (PCC) applications. This paper proposes an embedded real-time AED algorithm dedicated to PCC systems. This AED algorithm consists of a QRS detector and a rhythm classifier. The QRS detector adopts the linear time-domain statistical and syntactic analysis method and the geometric feature extraction modeling technique. The rhythm classifier employs the self-learning expert system and the confidence interval method. Currently, this AED algorithm has been implemented and evaluated on the PCC system for 30 patients in the Gabriel Monpied hospital (CHRU of Clermont-Ferrand, France) and the MIT-BIH cardiac arrhythmias database. The overall results show that this energy efficient algorithm provides the same performance as the classical ones.

Keywords: Pervasive Cardiac Care, Automatic ECG Diagnosis, QRS detector, Rhythm Classifier, Wireless Sensor Networks

1. Introduction

Due to the increasing occurrence of sudden death events caused by cardiovascular diseases, there is a need to provide a long-term, real-time continuous PCC service for the sudden death high-risk population. The PCC system has thus been developed for different populations at a variety of environment, including at home, clinical and outdoor.

The studies of AED methods focused mainly on the clinical services. Unlike the clinical applications, the acquisitions of the PCC system is ambulatory ECG signal that is non-stationary and easy-disturbed by interferences. Moreover, the nodes of the PCC system have strict resource constraints, i.e. the capacities of computation, storage and power supply. Classical AED algorithms are thus unfit for the PCC system.

Supported by Doctoral Fund of Youth Scholar of Ministry of Education of China (No.200802131024), French Program of Cooperation with China (No.20974WG), and Scientific Research Fund of Returned Oversea Scholars of Harbin city of China (No.RC2009LX010001).

This paper presents a real-time and low resource consumption AED algorithm for the PCC system. Section 2 introduces the state-of-the-art of the AED algorithms. Section 3 describes this algorithm in detail and section 4 presents the performance evaluation. The conclusions are drawn at the last section.

2. State-of-the-Art

Due to its high potential amplitude, steep slope (R-wave) and wide duration, QRS complex is generally used for the cardiac event diagnosis and analysis. Different AED algorithms are classified by Köhler *et al.* [1]: 1). Time-domain analysis can implement a simple and rapid detection but it is noise-sensitive; 2). Wavelet transform analysis has high detection performance but has huge computation overhead; 3). Syntax analysis exposes the wave pattern elements and their mutual relations, but it is noise-sensitive and has huge computations; 4). Neural network analysis needs a large amount of training sample set and long training time.

Other classical AED techniques include: template matching [2], hidden Markov model [3], Hilbert transform [4], mathematical morphology [5] method, etc. These techniques generally have huge computation overhead. The new AED algorithms generally integrate multiple techniques. For example, Oliveira *et al.* [6] integrates the Hilbert transform and wavelet transform, and Szilágyi *et al.* [7] combines the neural network, wavelet transform and genetic algorithm techniques. Generally, these hybrid methods can improve the detection accuracy, but have huge computation overhead, more resource consumption and less operation efficiency.

3. AED Algorithm

3.1. Signal Preprocessing and Conditioning

Due to the non-stationary and easy-disturbed natures of the ambulatory ECG signals, the acquisitions of PCC system must be de-noised before making detection. Most of artifacts, such as baseline shift, electrical noise and muscle tremor interference, can be effectively eliminated or reduced by choosing suitable filters. In this subsection, we present the filter techniques.

3.1.1. ECG Time Series

There are three ECG signals series, i.e. R(t), AD(t) and RC(t), in our algorithm. The R(t) series is the raw ECG signals acquired from electrodes. It's generally contaminated by different kinds of noises. The AD(t) series is the adaptive differential signals with the processing of the differential filter and the adaptive filter. The inferences of the baseline drift and the motion artifacts can be eliminated in the AD(t) series; hence this series is used to detect and to localize the QRS complexes. The RC(t) series is the de-noised ECG signals with the operations of the band-pass filter and the linear amplifier. Since the electrical noises and the muscle tremors have been removed from the RC(t) series, hence it is used to extract the characteristics of the QRS complexes.

3.1.2. Adaptive Filter

The classical filter for the ECG series, e.g. Notch filter, low-pass filter, and high-pass filter, can effectively remove or reduce most of the interferences. But for the motion artifacts, because of their irregular occurrences and irregular morphological attributes, these filters cannot eliminate these disturbances. These artifacts can make greatly troubles in QRS detection when encountering QRS-like artifacts.

This algorithm adopts an adaptive filter to reduce motion artifacts. The resultant signal series, named A(t), are generated by performing AT operation in the raw series R(t). The expression of adaptive filter is

$$\begin{cases} Aecg(0) = R(0) \\ \qquad\qquad 0 < \alpha < 1, t = 1 \cdots N \\ Aecg(t) = \alpha * Aecg(t-1) + (1-\alpha) * R(t) \end{cases} \quad (1)$$

were α s the balance coefficient that is relative to the signal sample frequency (default 0.95).

Figure 1 shows different ECG series. Figure 1(a) is the raw signals R(t) which are serious polluted by noises. Figure 1(b) represents the reconstructed series RC(t) when filtering the R(t) series by the classical filters, i.e. Notch filter, low-pass filter and high-pass filter. The RC(t) series still contain the interferences generally caused by baseline wandering and motion artifacts. Figure 1(c) is the adaptive filter signal A(t) when filtering the R(t) series by the adaptive filter, which has better signal quality than RC(t). Figure 1(d) is the reconstructed signal RC*(t) based on the adaptive filter signal A(t). Obviously, in contrast to the previous reconstructed signal RC(t), the signal RC* (t) has better signal quality in which the motion artifacts are effectively eliminated.

3.2. QRS Complex Detection

This paper presents a new QRS detector which copes with noises, artifacts and variability of ECG morphology by exploiting a self-adaptive threshold method (SAT), and a particular state transition recognition procedure (STR). The SAT method is used to estimate the peaks of ECG sub-segments and the means of contextual thresholds, which allows estimating the optimum thresholds in segment space. The STR procedure traces the waveform changes of signal series and identifies QRS complexes based on the optimum thresholds and the rules of state transition.

3.2.1. Diagnostic Segment Window (DSW)

A short-term redundant data (default 5 seconds) is important in QRS detection. Firstly, this short-term segment

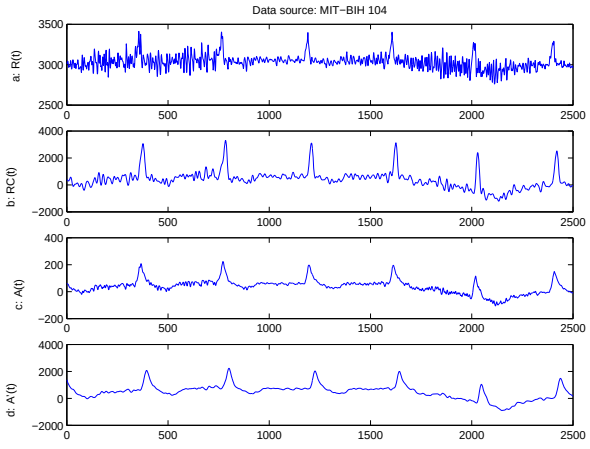


Figure 1. Filtered ECG signal series.

enables the complex contextual correlative analysis and reduces the interferences of baseline drift. In view of the low-frequency baseline drift, a short-term segment has fewer disturbances caused by baseline wandering than long-term signals. The redundant data enable the QRS detector to identify current QRS complex by comparing with the fore-and-aft QRS complexes. Furthermore, in views of the unpredictability and variability of network quality, the redundancy is necessary for the data retransmission and the network communication.

3.2.2. Self-Adaptive Threshold (SAT)

The QRS waveforms in AECG have rapid changes and high potential amplitudes so that the differential series *D*(t) can exactly represent the changes. The QRS signals have higher absolute amplitudes in a cardiac cycle of *D*(t) series. The goal of QRS detection is to search the optimum pair-peak for each QRS complex, i.e. the positive and negative peaks of a cardiac cycle. In DSW, there are generally multiple pair-peaks because several heart beats will occur during the 5s length. These pair-peaks make up of a pair-peak group in a DSW. Based on the pair-threshold obtained from the pair-peaks group of a DSW, the STR procedure is then able to locate QRS complexes. The absolute amplitude of each peak is generally greater than the associated absolute threshold in *D*(t). Furthermore, since the offset of location between the *D*(t) series and the *A*(t) series is constant, we can thus obtain the positions of QRS complexes in *A*(t) by locating the complexes in *D*(t).

The SAT method aims to determine the optimum pair-threshold, which is estimated from two aspects: the mean of the pair-peak group of DSW and the pair-threshold of the previous DSW. The pair-threshold results from the means of the negative and positive pair-peaks group of DSW. In order to accurately estimate these pair-peaks, the diagnostic segment window is divided into 5 sub-segments with the length of one second (see in Figure 2). Because the normal heart rate of a healthy adult is 60bmp-100bmp [8], each sub-segment thus contains one heart beat. Since the differential signals of QRS complex have the maximum absolute amplitudes in a cardiac cycle, a pair-peak will represent a QRS complex and then can be used to estimate the thresholds. Furthermore, the shorter of sub-segment is, the less interference of baseline drift the sub-segment has. A sub-segment with the length of one second can thus be regarded as a stationary series.

3.2.3. QRS Location: State Transition Recognition

In view of the QRS morphology properties in *D*(t) series, the complexes are categorized into two groups: positive and negative. Therefore, the different states are defined to outline the phases of QRS complex in *D*(t). *S*2~*S*9 represent the positive states of QRS complex (see in Figure 3), corresponding *S*20-*S*29 represent negative states. An adaptive and self-corrected procedure, named STR (State Transition Recognition), is developed to automatically track the changes of signal series, to correct error detection and to record detected complexes. The states transitions are based on three basic reference lines: the baseline, the positive threshold and the negative threshold.

3.2.4 Feature Extraction: Geometric Analysis Method

QRS complex has the triangular-alike or triangular-component morphological characteristics, see in Figure 4. This paper thus employs the geometric analysis method (GAM) to extract the features of QRS complexes. GAM has simple operations and low resource consumption, being able to predict and estimate the key points of QRS complexes under noisy situations, such as R wave peak, end point of Q wave (Qt) and onset point of S wave (Si). Therein, R wave peak is obtained from Tpeak1 or Tpeak2 and it has mono-peak or poly-peaks. The measurement and the detection phases of Qt and Si points are illuminated as follows.

- Define two-level thresholds for left and right sides of R wave (LH=1/4*Vpeak1, LL=3/4*Vpeak1, RH= 1/4*Vpeak2 and RL=3/4*Vpeak2).

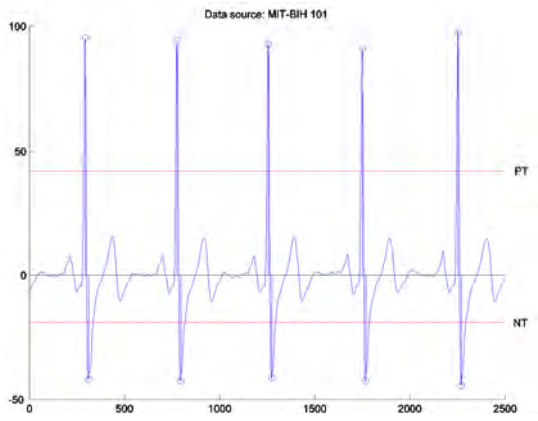


Figure 2. Mean of pair-peak group in diagnostic window.

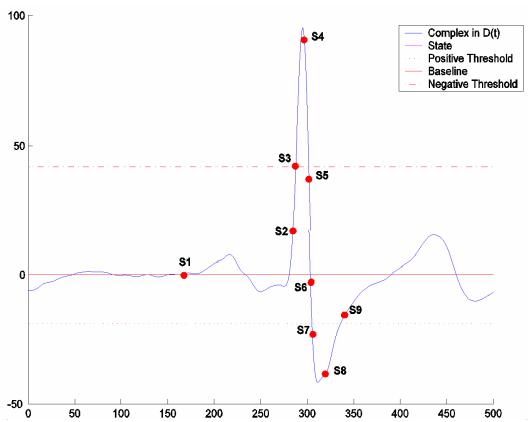


Figure 3. Positive states of QRS complex in D(t).

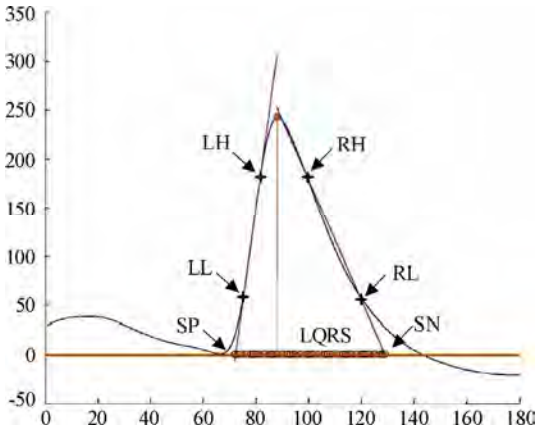


Figure 4. Illumination of geometric analysis method.

- Calculate the intersection points between the threshold values and complex signals. The slopes of two approaching lines represent two characteristics of QRS complex: SP (Positive Slope) and SN (Negative Slope).
- Obtain the duration length of QRS (LQRS) which is the distance of two intersection points between the baseline and two approaching lines.

3.3. Cardiac Arrhythmias Classification

Basing on the features values extracted from ECG signals, a self-diagnosis expert system is implemented to classify heart rhythms and interpret cardiac arrhythmias. The diagnostic rules of the expert system rely on the experiential rules estimated from the self-learning of system and the definitions of cardiologists. The diagnosis system is composed of three phases: a pre-learning machine, a rhythm classifier and an arrhythmia interpreter, see in Figure 5.

Based on the well-known experiential rules of cardiologists and the results of the training procedure, the *pre-learning machine* builds and estimates the diagnostic rules for every lead ECG signals of a patient. The *rhythm classifier* classifies each detected heart rhythm into one of two catalogues: known rhythm or unknown rhythm. For the known rhythms, they are still classified into two types according to the values of the RR intervals: sinus rhythm and ventricular rhythm; and for the unknown rhythms, we will adopt classical methods to classify, the classification results will be verified by the cardiologists. In terms of the known rhythm types and the diagnostic rules, the *cardiac arrhythmias interpreter* is used to explain cardiac arrhythmias with the symptoms of relative heart diseases.

3.3.1. Automatic Learning Machine

Ten seconds ECG signals are used to calculate the rhythm template and to estimate the diagnostic rules. The initial cardiac status, rhythm type, statistical and morphological features are achieved in this module. The diagnostic results will be further fed back to adjust the coefficients of diagnostic rules. Unlike resting ECG, long-term ambulatory ECG has continual tiny changes with the influences of exterior environments and the patient's physical status. The tiny changes are generally normal and the coefficients of diagnostic rules thus should be self-updatable to meet the changes.

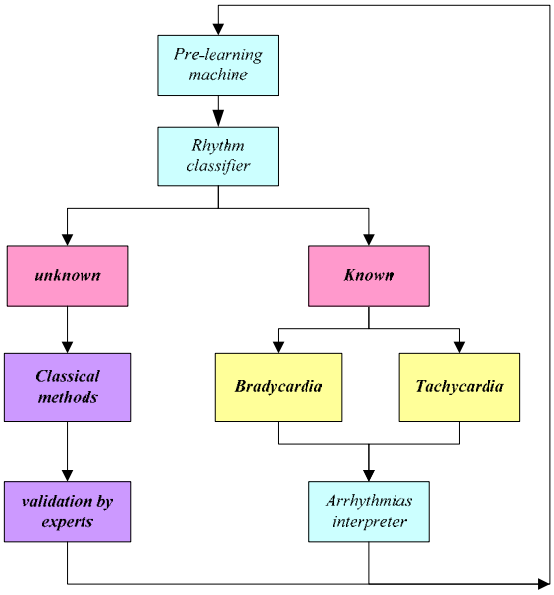


Figure 5. Illumination of automatic diagnosis system.

3.3.2. Rhythm Classifier

By adopting the expert system and the confidence interval method, the rhythm classifier can recognize two kinds of QRS complex rhythms: sinus and ventricular. The details of signal features (RR interval, QRS duration, R wave left- & right-sides slopes, R wave amplitude, and QRS absolute area), the rhythm type and the complex peaks are used to describe a heart rhythm. Hence, they can be used to recognize a rhythm and by comparing with the features of the rhythm template.

The rhythm classifier is based on the features comparison and the interval estimation. Since we have obtained the features of current rhythm and the features of the standard rhythm (rhythm template) in pre-learning machine, the rhythm classification is thus to estimate the confidence intervals, the weighed factors and the deviation coefficients of the features. The classification equation can be expressed as:

$$\rho = \sum_{i=1}^{N} \rho_i = \sum_{i=1}^{N} (\beta_i * \alpha_i) \quad (2)$$

Where the ρ_i is the classification factor that is used to determine the heart rhythm by estimating the confidence

interval that it falls; the β_i is the weighed factor that indicates the contribution of the feature *i*; the α_i is coefficient of deviation that associates the variation of the feature *i*; the *N* indicates the number of the features.

3.3.3. Arrhythmia Interpreter

In terms of the rhythm type and heat rate (HR), a heart rhythm can be recognized and interpreted by the arrhythmia interpreter basing on diagnostic rules. Firstly, the arrhythmia interpreter classifies the rhythms into two catalogues: bradycardia and tachycardia by comparing current HR with the mean HR of the rhythm template.

In arrhythmia interpreter, the heart rhythms are identified and classified into two basic categories: normal cardiac rhythms and cardiac arrhythmias, and the cardiac arrhythmias can be further divided as two classes: the known cardiac arrhythmias and the unknown cardiac arrhythmias. The known cardiac arrhythmias are normally the cardiac tachycardia events which are caused by serious heart diseases, including PVC (Premature ventricular complexes), VT (Ventricular tachycardia), VF (Ventricular Fibrillation), SVT (Supraventricular Tachycardia) and PAC (Premature Atrial Contraction), etc. The known cardiac arrhythmias have distinctive QRS complexes and rapid heart rates which make them be interpreted accurately. The unknown cardiac arrhythmias are normally the bradycardia events which are caused by less serious or benign heart diseases. The known cardiac arrhythmias have regular heart rhythms but slow heart rates, the identifications of which are not reliable when depending only on the heart rate.

4. Performance Analysis

The algorithm has been assessed on two ECG databases: MIT-BIH arrhythmia database [9] and CSD database (Clinic *STAR* Database). The former contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, and the latter is obtained from 30 subjects of the Gabriel Montpied hospital (CHRU de Clermont-Ferrand, France) by using a PCC system named *STAR* [10]. The CSD signals are recorded in the same format (WFDB) as MIT-BIH Database one.

4.1. STAR System

Currently, a real-time remote continuous cardiac arrhythmia detecting and monitoring system, named STAR (Système Télé-Assistance Réparti), has been developed by the SMIR group of LIMOS laboratory of the Blaise Pascal University and been applied on the CHU de Gabriel Montpied hospital (Clermont-Ferrand, France). The STAR system combines the technology advantages of pervasive computing, AED algorithm and remote telemedicine system. Figure 6 shows its system architecture, which consists of local wireless ECG sensor (WES) nodes and remote cardiac surveillance system.

The system description is: a WES device equipped by the surveillance object, which integrates the AED algorithm, can acquire and analyze the patient's ECG signals in real-time. When a cardiac abnormal event is detected, an alarm message and (or) a segment of ECG signals will send to the cardiologists via the available wired or wireless communication mediums. In the remote cardiac surveillance system, the cardiologists can examine cardiac abnormal events by employing AED algorithm and make a respond with the shortest delays. This system aims to provide a rapid detection and diagnosis method for the high-risk population of cardiac arrhythmias to prevent sudden death. It is also used to do long-term heart surveillance for the population who has the history of heart diseases, or to do periodic heart examination for the health population.

4.2. QRS Detector Evaluation

Dotsinsky *et al.* [11] defined four performance parameters to assess the algorithm efficiency (Se: sensitivity and Sp: specificity): TP (true positive), FP (false positive), FN (false negative) and shifted SH beats, shown as follows:

$$Se = \frac{TP}{TP + FN + SH}$$

$$Sp = 1 - \frac{FP}{TP + FP} = \frac{TP}{TP + FP} \quad (3)$$

Comparing with the performance results of other algorithms listed in Table 1, the performance results of this

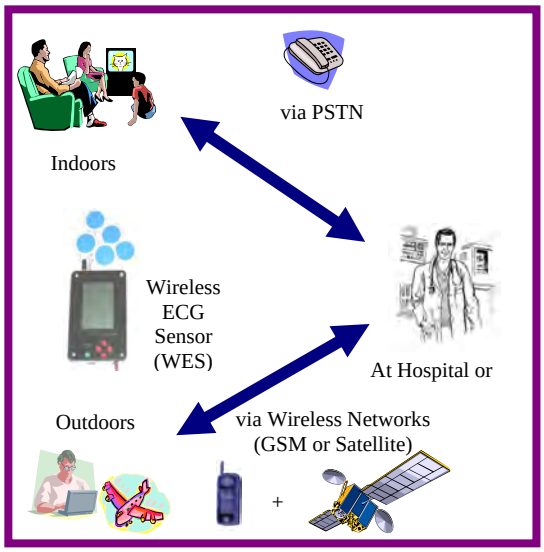


Figure 6. Architecture of STAR system.

Table 1. Performance evaluation of QRS detection algorithms.

		Se(%)		Sp(%)	
Afonso *et al* [12]		99.59		99.56	
Poli *et al* [13]		99.60		99.51	
Dotsinsky *et al* [11]		99.04		99.62	
Kaiser *et al* [14]		99.68		99.72	
Datex-Ohmeda Corp. [16]		99.86		99.88	
Millet *et al* [15]	Alg 1	94.6		98.0	
	Alg 2	97.3		98.0	
Our Algorithm		99.43 MIT	99.25 CSD	98.55 MIT	97.94 CSD

detection algorithm, 99.37% sensitivity and 99.68% specificity on MIT-BIH database, 99.67% sensitivity and 99.74% specificity on CSD database, show the high sensitivity and specificity. This detection algorithm has minimal beat detection latency, low computational consumption and fast detection ability.

4.3. Rhythm Classifier Evaluation

The rhythm classifier classifies heart rhythms into two catalogues: non-alarm and alarm-rhythms. The alarm-rhythms defined in our algorithm are tachycardia, i.e. PAC, PVC, SVT, VT, and VF. They represent serious heart diseases which need to be reported immediately. The non-alarm rhythms include the normal rhythms and some benign or less serious cardiac arrhythmias, such as bradycardia.

The four parameters are used to assess the algorithm performance [17]: A true positive (TP) is a serious cardiac arrhythmia that has been correctly classified as an alarm- rhythm; A false positive (FP) is an organized normal rhythm that has been incorrectly classified as an alarm- rhythm; A true negative (TN) is any normal or less serious rhythm that has been correctly classified as a non-alarm rhythm; A false negative (FN) is a serious cardiac arrhythmia that has been incorrectly classified as a non-alarm rhythm.

The sensitivity (Se) is the number of true positive abnormal rhythms, expressed as a percentage of the total number of abnormal rhythms. Se is calculated by formula (3). The specificity (Sp, also named positive predictive accuracy) is the number of organized rhythms that have been correctly classified as normal rhythms, expressed as a percentage of the total number of normal rhythms and computed by formula (4).

$$Sp = \frac{TN}{FP + TN} \tag{4}$$

Comparing with the performance of other algorithms listed in Table 2, the performance results of this classification algorithm, 90.90% sensitivity and 95.50% specificity on MIT-BIH database, 95.6% sensitivity and 99.5% specificity on CSD records, show its good performance. Since the features extracted by the detection algorithm are the time domain characteristics of QRS complex, this classification algorithm thus can directly utilize the experiences of cardiologists that reduces the complexity of rules training and then improves the accuracy of classification. Another advantage is that this algorithm is able to identify various cardiac arrhythmias comparing to most of other algorithms.

5. Conclusion

The objective of our research is to design a real-time energy efficient Automatic ECG Diagnosis algorithm for the PCC system. The PCC application is free of the limitations of time and space, that is, this system supports long-term monitoring (from few days to one month) and the patient have the freedom of daily actions. The results of the performance evaluation show that our algorithm satisfies application demands.

Table 2. Performance evaluations of rhythm classification algorithm.

	NSR		PAC		PVC		SVT		VT		VF		Total	
	Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp
Horácek [18]									90.3	78.6				
Ge *et al* [19]	93.2	94.4	96.4	96.7	94.8	96.8	100	96.2	97.7	98.6	98.6	97.7	96.83	96.73
Ham & Han [20]					99	97								
Chen *et al* [21]									93		96			
Minami *et al* [22]					>98	>98			>98	>98				
Chen [23]							95.24		96.00		97.78			
Melo *et al* [24]			93		99									
Datex-Ohmeda [16]					94.08	97.55								
Philips AED [17]									84	91	97*,76**	91		
Our Algorithm													90.9[1] 95.6[2]	95.5[1] 99.5[2]

*:Ventricular Fibrillation (amplitude > 0.200mv). **: Fine Ventricular Fibrillation (0.100mv<amplitude < 0.200mv)
[1]Evaluation results on MIT-BIH database [2]Evaluation results on CSD clinical records

In this AED algorithm, the QRS detector adopts linear time-domain statistical analysis and syntactic analysis methods to locate QRS complex from AECG signals. The signal preprocessing and conditioning procedure, adopting adaptive filter and band-pass filter, remove or reduce various interferences caused by physical and technical factors. The most serious noisy, such as motion artifacts, has been effectively eliminated by the adaptive filter. According to the statistical feature and morphologic features of QRS complex, i.e. heart rate, steep edges and sharp amplitude, the QRS complex is located to mark heart beat by applying SAT method and STR procedure on sub-segment diagnosis window.

The rhythm classifier classifies rhythms and interprets cardiac arrhythmias basing upon the diagnostic rules which are obtained from the experiences of cardiologists and the training results of pre-learning phase. The initial ECG signals with the length of 10 seconds are used to estimate the type of QRS complex and to extract the features of normal rhythm template (the means of LQRS, RR, etc.). According to the origination of heart beat, the rhythms are categorized into two classes: sinus rhythm (atria) and ventricular rhythm (ventricle). According to the changes of heart rate, cardiac arrhythmias are categorized into two classes: bradycardia and tachycardia. The cardiac arrhythmias interpretation procedure is adopted to classify cardiac arrhythmias into various types of bradycardia and tachycardia, based on the features extracted in the detection algorithm.

Currently, this algorithm has been applied on the *STAR* system. The performance evaluations results show that this algorithm was effective for the QRS detection and the rhythm classification, and was thus suitable for PCC services. The simple, fast and efficient features of this algorithm enable it to be embedded into microprocessor system or be implemented on chip.

6. Acknowledgement

This project is supported by OSEO (French research agency) and the Conseil Régional d'Auvergne (France).

7. References

[1] B.-U. Köhler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection," IEEE Engineering in Medicine and Biology Magazine, Vol. 21, No. 1, pp. 42–57, 2002.

[2] J. M. Jenkins and S. A. Caswell, "Detection algorithm in implantable cardioverter Defibrillators," Proceedings of the IEEE, Vol. 84, No. 3, pp. 428–445, 1996.

[3] S. A. Coast, R. M. S tem *et al.*, "An approach to cardiac arrhythmia analysis using hidden markov models," IEEE Transaction On Biomedical Engineering, Vol. 37, No. 9, pp. 826–835, 1990.

[4] D. S. Benitez, P. A. Gaydecki, A. Zaidi *et al.*, "A new QRS detection algorithm based on the Hilbert transform," IEEE Computers in Cardiology, pp. 379–382, 2000.

[5] P. E. Trahanias, "An approach to QRS complex detection using mathematical morphology," IEEE Transaction On Biomedical Engineering, Vol. 40, No.2, pp. 201–205. 1993.

[6] F. I. de Oliveira and P. U. Cortez, "A QRS detection based on Hilbert transform and wavelet," Proceedings of 14th IEEE SPSW on MLSP, pp. 481–489, 2004.

[7] S. M. Szilágyi, Z. Benyo, L. Szilagyi, *et al.*, "Adaptive wavelet transform based ECG waveforms detection," Proceedings of 25th Annual IEEE EMBS International Conference, No. 24, pp. 12–15, 2003.

[8] W. A. H. Engelse and C. Zeelenberg, "A single scan algorithm for QRS-detection and feature extraction, Computers in Cardiology," No. 6, pp. 37–42, 1979.

[9] G. B. Moody and R. G. Mark, "The MIT-BIH arrhythmia database on CD-ROM and software for use with it," Computers in Cardiology, Vol. 17, pp. 185–188, 1990.

[10] H. Y. Zhou, K. M. Hou *et al.*, "Remote continuous cardiac arrhythmias detection and monitoring," 2nd International Conference on E-health in Common Europe, Krakow, pp. 11–12. March 2004.

[11] I. A Dotsinsky and T. V Stoyanov, "Ventricular beat detection in single channel electrocardiograms," Biomedical Engineering Online, Vol. 3, No. 3, 2004.

[12] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen and S. Luo, "ECG beat detection using filter banks," IEEE Transaction on Biomedical Engineering, Vol. 46: pp. 192–202. 1999.

[13] R. Poli, S. Cagnoni, and G. Valli, "Genetic design of optimum linear and nonlinear QRS detectors," IEEE Transaction On Biomedical Engineering, Vol. 42, pp. 1137–1141, 1995.

[14] W. Kaiser and M. Findeis, "Novel signal processing Methods for exercise ECG," Special issue on Electrocardiography in Ischemic Heart Disease, Proceedings of IJBEM, Vol. 2, 2000.

[15] J. Millet, M. Perez, G. Joseph, A. Mocholi, and J. Chorro, "Previous identification of QRS onset and offset is not essential for classifying QRS complex in a single lead," Computers in Cardiology, Vol. 24, pp. 299–302, 1997.

[16] Datex-Ohmeda Corp.: Bedside Arrhythmia Monitoring Quick Guide, Education Report, Internal web journal for medical professionals (Focus on Cardiovascular), URL: http://www.clinicalwindow.com, November 2002.

[17] Philips Medical systems: AED Algorithm application Note Philips HeartStart, Technical Document, URL: http://www.medical.philips.com, Philips Electronics North America Corporation, 2003.

[18] M. B Horácek, "Body-surface potential maps can identify substrate for ventricular arrhythmias," International Journal of Bioelectromagnetism," Vol. 5, No. 1, pp. 331–334, 2003.

[19] D. F. Ge, N. Srinivasan, and S. M. Krishnan, "Cardiac

arrhythmia classification using autoregressive modeling," Biomed Eng Online, Vol. 1, No. 5, 2002.

[20] F. M. Ham and S. Han, "Classification of cardiac arrhythmias using fuzzy ARTMAP," IEEE Trans Biomed Eng; Vol. 43, No. 4, pp. 425–30, April 1996.

[21] S. W. Chen, P. M. Clarkson, and Q. Fan, "A robust sequential detection algorithm for cardiac arrhythmia classification," IEEE Trans Biomed Eng, Vol. 43, No. 11, 1120–5, November 1996.

[22] K. C. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network," IEEE Trans Biomed Eng, Vol. 46, pp. 179–185, 1999.

[23] S. W. Chen, "Two-stage discrimination of cardiac arrhythmias using a total least squares-based prony modeling algorithm," IEEE Trans Biomed Eng, Vol. 47, pp. 1317–1326, 2000.

[24] S. L. Melo, L. P. Caloba, and J. Nadal, "Arrhythmia analysis using artificial neural network and decimated electrocardiographic data," Comp Cardiol, Vol. 27, pp. 73−76, 2000.

Wireless Sensor Network, **2009, 1, 284-293**
doi:10.4236/wsn.2009.14035 Published Online November 2009 (http://www.scirp.org/journal/wsn).

The Estimation of Radial Exponential Random Vectors in Additive White Gaussian Noise

Pichid KITTISUWAN[1], Sanparith MARUKATAT[2], Widhyakorn ASDORNWISED[1]
[1]*Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand.*
[2]*Image Laboratory, National Electronics and Computer Technology Center, Bangkok, Thailand*
Email: *pichidkit@yahoo.com, sanparith.marukatat@nectec.or.th, widhyakorn.a@chula.ac.th*
Received April 22, 2009; *revised June* 15, 2009; *accepted June* 22, 2009

Abstract

Image signals are always disturbed by noise during their transmission, such as in mobile or network communication. The received image quality is significantly influenced by noise. Thus, image signal denoising is an in dispensable step during image processing. As we all know, most commonly used methods of image denoising is Bayesian wavelet transform estimators. The Performance of various estimators, such as maximum a posteriori (MAP), or minimum mean square error (MMSE) is strongly dependent on correctness of the proposed model for original data distribution. Therefore, the selection of a proper model for distribution of wavelet coefficients is important in wavelet-based image denoising. This paper presents a new image denoising algorithm based on the modeling of wavelet coefficients in each subband with multivariate radial exponential probability density function (pdf) with local variances. Generally these multivariate extensions do not result in a closed form expression, and the solution requires numerical solutions. However, we drive a closed form MMSE shrinkage functions for a radial exponential random vectors in additive white Gaussian noise (AWGN). The estimator is motivated and tested on the problem of wavelet-based image denoising. In the last, proposed, the same idea is applied to the dual-tree complex wavelet transform (DT-CWT), This Transform is an overcomplete wavelet transform.

Keywords: Minimum Mean Square Error (MMSE), Radial Exponential Random Vectors, Wavelet Transform

1. Introduction

The denoising of a natural image corrupted by Gaussian noise is a classic problem in signal processing. The distortion of images by noise is common during its, acquisition, processing, compression, mobile and network transmission. Traditional algorithms perform image denoising based on threshold function methods, such as soft-threshold and hard-threshold [1]. If the wavelet transform and MMSE estimator are used for this problem, the solution requires a priori knowledge about wavelet coefficients. Therefore, two problems arise: 1) what kinds of distributions represent the wavelet coefficients? 2) What is the corresponding estimator (Shrinkage function)?

Figure 1 illustrates the histogram of the wavelet coefficients in one subband of the wavelet transform of a photographic image. The histogram conforms to the well-known behavior of such histograms-namely, compared to the Gaussian pdf, the histogram has a different behavior both at the center (it is more peaked) and in tails (they are heavier). Wavelet coefficient histograms are more kurtotic than Gaussian distributions. It is known that the amplitude of wavelet coefficients tend to propagate across scales. This parent-child relation is also underlined by the empirical joint histogram between parent and child coefficients as shown in [2]. In [3], they developed a multivariate spherically contoured Laplace density that is similar to the radial exponential density in its function form, but the marginal of the density are not radial exponential density. Although radial exponential pdf specializes to Laplace pdf in the scalar case (d=1).

In this paper we focus on multivariate radial exponential distribution with local variances to model these locality and persistence properties of wavelet coefficients. The rest of this paper is organized as follows. In section II-A, the basic idea of Bayesian denoising will be briefly described. Section II-B describes wavelet coefficients

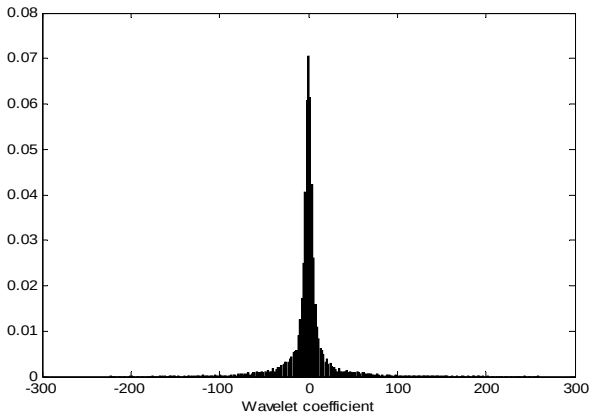


Figure 1. Histogram of wavelet coefficients in the LH subband at scale-2 of 512x512 pixel lena image (256 bins).

model, these models try to capture the dependencies between a coefficients and its group of parent in detail. In section III we derive a closed form of MMSE estimator using multivariate radial exponential distribution with local variance namely, *MMSE_TriShrink_radial*. Section IV describes the approximated MAP (maximum a posteriori) estimation for local variances using Rayleigh density priori with Gaussian distribution (the local variances estimation of wavelet coefficients is the key to get better performance for image denoising). In section V, we use our model for wavelet based denoising of several images corrupted with additive Gaussian noise in various noise levels. The simulation results in comparison with MMSE_TriShrink_Laplace [3], and BLS-GSM [4]. In the last simulation results, the performance of a subband dependent will be demonstrated on the dual-tree complex wavelet transform. The dual-tree complex wavelet transform (DT-CWT) is an overcomplete wavelet transform, which can be implemented by wavelet filter banks operating in parallel [5,6]. The discrete wavelet transform (DWT) used in image denoising can be of many types, such as orthogonal/biorthogonal, real/complex-valued, separable/nonseparable, or decimated/nondecimated. Due to the shift-invariance property, the overcomplete transform improves the image denoising performance in PSNR by 1 dB as compared to that of the decimated representation [7]. Finally the concluding remarks are given in section VI.

2. Bayesian Denoising and Useful Density

2.1. Bayesian Denoising

In this paper we are interested in the problem of estimating a d-component radial exponential random vector, $\mathbf{x}$ (noise-free wavelet coefficients) in additive white Gaussian noise (AWGN), $\mathbf{n}$

$$\mathbf{y}=\mathbf{x}+\mathbf{n} \tag{1}$$

The marginal models are weak models for wavelet coefficients of natural images because they ignore the dependencies between coefficients (although a coefficient and its parent are uncorrelated but are not independent). It is well known those wavelet coefficients are statistically dependent due to two properties of the wavelet transform 1) If a wavelet coefficient is large/small, the adjacent coefficients are likely to be large/small, and 2) large/small coefficients tend to propagate across the scales. Here, we can update the MMSE estimation problem as to take into account the statistical dependency between a coefficient and its group of parent. Let x_2, x_3 ….. x_d represent the group of parent of x_1 (x_2, x_3 ……. x_d is the wavelet coefficient at the same spatial position as x_1, but at the next coarser scale). If we observe a noisy wavelet coefficient $\mathbf{y}$, where $\mathbf{y},\mathbf{x}$ and $\mathbf{n}$ are random vectors (d-dimensional) and $\mathbf{n}$ is additive Gaussian noise with variance σ_n^2, MMSE equation can be rewritten as:

$$\hat{x}_1(\mathbf{y}) = \frac{1}{f_{\mathbf{y}}(\mathbf{y})} \int_{R^d} x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{y}\,|\,\mathbf{x})\, f_{\mathbf{x}}(\mathbf{x}) \left| d\mathbf{x} \right| \tag{2}$$

where $f(.)$ is the probability density function

2.2. Wavelet Coefficients Distribution

In [3], and [8] a multivariate Laplace distribution and bivariate Cauchy distribution are proposed to model wavelet coefficient and group of parent joint pdf. Figure 2 shows the marginal distribution of d-dimension radial exponential distribution when d=1 (solid line) and d=9 (dashed line). When d=1 (scalar) marginal distribution of radial exponential distribution is Laplace distribution.

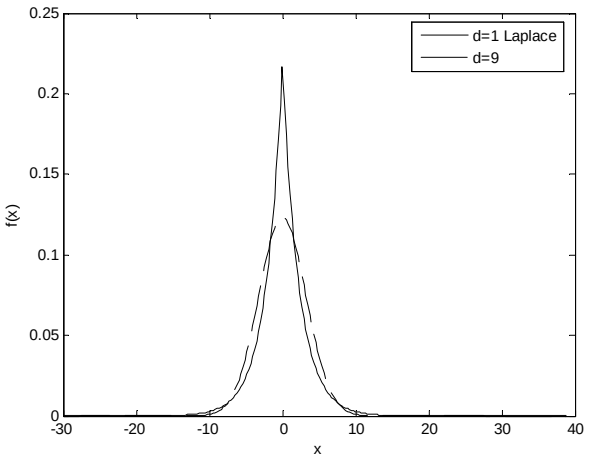


Figure 2. The marginal distribution of the d-component radial exponential distribution (3) for d=1 and d=9, σ^2=4. When d=1 the distribution is Laplace distribution.

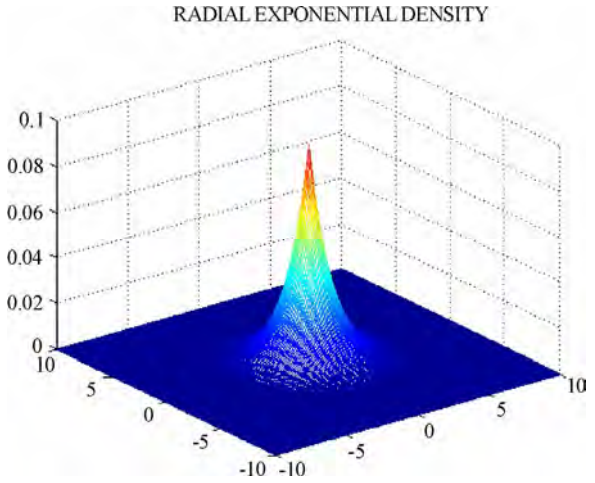


Figure 3. Bivariate radial exponential distribution (3) with σ^2=4, 2-dimensional (d=2).

For d>1 the marginal distribution of radial exponential is less kurtosis than the Laplace distribution. As d-dimension increases, the marginal distribution becomes more Gaussian. Figure 3 show bivariate radial exponential pdf. The multivariate spherically contoured of radial exponential distribution zero mean with variance σ^2 has the density

$$f_{\mathbf{x}}(\mathbf{x}) = C\frac{1}{\sigma^d}\exp(-\frac{\sqrt{d+1}}{\sigma}\|\mathbf{x}\|), \qquad \mathbf{x}\in R^d \qquad (3)$$

Where the normalization constant C is

$$C = \frac{\sqrt{\pi}}{\Gamma(\frac{d+1}{2})}(\frac{d+1}{4\pi})^{d/2},\ \Gamma(.)\ \textit{is gamma function},\ \|.\|\ \textit{is norm 2}$$

3. MMSE Estimator with Radial Exponential Random Vectors

3.1. Generalized Incomplete Gamma Function

In 1994, Chaudhry and Zubair introduced the generalize incomplete gamma function [3], defined as

$$\Gamma(\alpha,x;b) = \int_x^\infty t^{\alpha-1}\exp(-t-\frac{b}{t})dt \qquad (4)$$

For $\alpha = Z+1/2$, Z are set of integer number, there is a closed from expression for the generalized incomplete gamma function. For $\alpha = 1/2$, for example, there is the formula

$$\Gamma(\frac{1}{2},x;b) = 0.5\sqrt{\pi}[\exp(-2\sqrt{b})erfc(\sqrt{x}-\sqrt{\frac{b}{x}}) + \exp(2\sqrt{b})erfc(\sqrt{x}+\sqrt{\frac{b}{x}})]$$

where $erfc(x) = 1-erf(x)$. For $\alpha = -1/2$ there is the formula

$$\Gamma(-\frac{1}{2},x;b) = 0.5\sqrt{\pi}[\exp(-2\sqrt{b})erfc(\sqrt{x}-\sqrt{\frac{b}{x}}) - \exp(2\sqrt{b})erfc(\sqrt{x}+\sqrt{\frac{b}{x}})]$$

The generalized incomplete gamma function satisfies a recurrence relation that is useful for computing its values for other orders α, from [9,10]

$$\Gamma(\alpha-1,x;b) = \frac{1}{b}[\Gamma(\alpha+1,x;b) - \alpha\Gamma(\alpha,x;b) - x^\alpha\exp(-x-\frac{b}{x})]$$

For α not of the form Z+1/2, no closed form expression is available for $\Gamma(\alpha,x;b)$.

3.2. MMSE Estimator with Multivariate Radial Exponential Distribution

Multivariate spherically contoured of radial exponential density can be generated by

$$\mathbf{x} = \sqrt{z}\mathbf{w}$$

If $\mathbf{w}$ is *d*-component Gaussian random vectors having zero mean with variance σ^2 and z is a gamma distribution (scalar) random variable then $\mathbf{x}$ is *d*-dimensional radial exponential random vectors. When $\mathbf{w}$ and z are independent. The pdf of z is

$$f_z(z) = \frac{\beta^\alpha}{\Gamma(\alpha)}z^{\alpha-1}\exp(-\beta z)$$

where $\alpha = \beta = \frac{d+1}{2}$. If d = 3 then $f_Z(z) = 4z\exp(-2z)$ and $\|\mathbf{w}\| = \sqrt{w_1^2+w_2^2+w_3^2}$, $\|\mathbf{x}\| = \sqrt{x_1^2+x_2^2+x_3^2}$.

We selected 3-dimensional (d=3) because the variance of the wavelet coefficients of natural images are quite different from scale to scale. As d-dimensional increases, get worse PSNR value (d=2 no closed from expression is available for MMSE shrinkage function).

In [3] they derive the MMSE estimator of Laplace random vectors. In this paper we call this method MMSE_TriShrink_Laplace (when 3-dimension). From Equation (2), we would like to find $f_{\mathbf{y}}(\mathbf{y})$. If the noise signal $\mathbf{n}$ is independent additive white Gaussian noise (AWGN) with variance σ_n^2 then the pdf of $\mathbf{y}$ is given by the multivariate convolution. The multivariate convolution defines as:

$$f_{\mathbf{y}}(\mathbf{y}) = f_{\mathbf{x}}(\mathbf{x}) * f_{\mathbf{n}}(\mathbf{n}) = \int_{R^d} f_{\mathbf{x}}(\mathbf{y}-\mathbf{s})\,f_{\mathbf{n}}(\mathbf{s})d\mathbf{s} \qquad (5)$$

First, setting $\sqrt{z}=a$, changing the variable, using Jacobian transform $J_{z\to a}=\left|\frac{dz}{da}\right|=2a$, therefore the pdf of a is

$$f_a(a)=|J_{z\to a}|(f_z(z=a^2))=(2a)4a^2\exp(-2a^2),\quad a\ge 0$$

Therefore,

$$f_{\mathbf{w},a}(\mathbf{w},a)=2a(4a^2\exp(-2a^2))\frac{1}{(2\pi\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{w}\|^2}{2\sigma^2})$$

Changing the variable $\mathbf{w}\to\mathbf{x}$, using the Jacobian transform for $\mathbf{w}=\frac{1}{a}\mathbf{x}$, $J_{\mathbf{w}\to\mathbf{x}}=\left|\frac{\partial\mathbf{w}}{\partial\mathbf{x}}\right|=\frac{1}{a^3}$, therefore

$$f_{\mathbf{x},a}(\mathbf{x},a)=|J_{\mathbf{w}\to\mathbf{x}}|(f_{\mathbf{w},a}(\mathbf{w}=\frac{\mathbf{x}}{\mathbf{a}},a))$$

$$=\frac{2a(4a^2\exp(-2a^2))}{(2\pi a^2\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{x}\|^2}{2a^2\sigma^2})$$

$$f_{\mathbf{x}}(\mathbf{x})=\int_0^\infty\frac{2a(4a^2\exp(-2a^2))}{(2\pi a^2\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{x}\|^2}{2a^2\sigma^2})da \qquad (6)$$

Using (5) and (6) to find $f_{\mathbf{y}}(\mathbf{y})$, where $f_{\mathbf{n}}(\mathbf{n})=\frac{1}{(2\pi\sigma_n^2)^{3/2}}\exp(\frac{-\|\mathbf{n}\|^2}{2\sigma_n^2})$ [3], the multivariate convolution is

$$f_{\mathbf{y}}(\mathbf{y})=f_{\mathbf{x}}(\mathbf{x})*f_{\mathbf{n}}(\mathbf{n})=\int_{R^d}f_{\mathbf{x}}(\mathbf{y}-\mathbf{s})f_{\mathbf{n}}(\mathbf{s})d\mathbf{s}$$

$$=\int_{R^3}(\int_0^\infty\frac{2a(4a^2\exp(-2a^2))}{(2\pi a^2\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{y}-\mathbf{s}\|}{2a^2\sigma^2})da)\frac{1}{(2\pi\sigma_n^2)^{3/2}}\exp(\frac{-\|\mathbf{s}\|^2}{2\sigma_n^2})|$$

$$=\int_0^\infty 2a(4a^2\exp(-2a^2))$$

$$\underbrace{(\int_{R^3}\frac{1}{(2\pi a^2\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{y}-\mathbf{s}\|^2}{2a^2\sigma^2})\frac{1}{(2\pi\sigma_n^2)^{3/2}}\exp(\frac{-\|\mathbf{s}\|^2}{2\sigma_n^2})|d\mathbf{s}|)}_{Gaussian\ convolution}da$$

$$=\int_0^\infty 2a(4a^2\exp(-2a^2))\frac{1}{(2\pi(a^2\sigma^2+\sigma_n^2))^{3/2}}\exp(\frac{-\|\mathbf{y}\|^2}{2(a^2\sigma^2+\sigma_n^2)})da$$

When multivariate Gaussian convolution defined as:

$$\int_{R^d}\frac{1}{(2\pi\sigma_1^2)^{d/2}}\exp(\frac{-\|\mathbf{y}-\mathbf{s}\|^2}{2\sigma_1^2})\frac{1}{(2\pi\sigma_2^2)^{d/2}}\exp(\frac{-\|\mathbf{s}\|^2}{2\sigma_2^2})|d\mathbf{s}|$$

$$=\frac{1}{2\pi(\sigma_1^2+\sigma_2^2)^{d/2}}\exp(\frac{-\|\mathbf{y}\|^2}{2(\sigma_1^2+\sigma_2^2)})$$

where σ_1^2 and σ_2^2 are any constant parameter, such as variance.

Changing the variable of integration according to $t=2a^2+2\sigma_n^2/\sigma^2$, using Jacobian transform $J_{a\to t}=\left|\frac{\partial a}{\partial t}\right|=\frac{1}{4a}$, and simplifying gives

$$f_{\mathbf{y}}(\mathbf{y})=\int_{\frac{2\sigma_n^2}{\sigma^2}}^\infty\frac{(t-2\sigma_n^2/\sigma^2)}{(\pi\sigma^2)^{3/2}t^{3/2}}\exp(-t+\frac{2\sigma_n^2}{\sigma^2}-\frac{\|\mathbf{y}\|}{\sigma^2 t})dt$$

$$f_{\mathbf{y}}(\mathbf{y})=\frac{\exp(2\sigma_n^2/\sigma^2)}{(\pi\sigma^2)^{3/2}}[\Gamma(\frac{1}{2},\frac{2\sigma_n^2}{\sigma^2};\frac{\|\mathbf{y}\|^2}{\sigma^2})$$

$$-\frac{2\sigma_n^2}{\sigma^2}\Gamma(\frac{-1}{2},\frac{2\sigma_n^2}{\sigma^2};\frac{\|\mathbf{y}\|^2}{\sigma^2}) \qquad (7)$$

The numerator of $\int_{R^d}x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y})f_{\mathbf{x}}(\mathbf{x})|d\mathbf{x}|$ when $f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y})=f_{\mathbf{n}}(\mathbf{y}-\mathbf{x})=\frac{1}{(2\pi\sigma_n^2)^{3/2}}\exp(\frac{-\|\mathbf{y}-\mathbf{x}\|^2}{2\sigma_n^2})$ is given by

$$\int_{R^d}x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y})f_{\mathbf{x}}(\mathbf{x})|d\mathbf{x}|=\int_{R^3}x_1\frac{1}{(2\pi\sigma_n^2)^{3/2}}\exp(\frac{-\|\mathbf{y}-\mathbf{x}\|^2}{2\sigma_n^2})$$

$$(\int_0^\infty\frac{2a(4a^2\exp(-2a^2))}{(2\pi a^2\sigma^2)^{3/2}}\exp(\frac{-\|\mathbf{x}\|^2}{2a^2\sigma^2})da)|d\mathbf{x}|$$

$$=\int_0^\infty 2a(4a^2\exp(-2a^2))[\int_{-\infty}^\infty\frac{1}{\sqrt{2\pi\sigma_n^2}}x_1\exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2})$$

$$\frac{1}{\sqrt{2\pi a^2\sigma^2}}\exp(\frac{-x_1^2}{2a^2\sigma^2})dx_1]\times$$

$$\underbrace{\int_{R^2}\frac{1}{2\pi\sigma_n^2}\exp(-\frac{(y_2-x_2)^2+(y_3-x_3)^2}{2\sigma_n^2})\frac{1}{2\pi a^2\sigma^2}\exp(-\frac{x_2^2+x_3^2}{2a^2\sigma^2})dx_2dx_3}_{Gaussian\ \ convolution}$$

Consider the term in the brackets

$$\int_{-\infty}^\infty\frac{1}{\sqrt{2\pi\sigma_n^2}}x_1\exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2})\frac{1}{\sqrt{2\pi a^2\sigma^2}}\exp(\frac{-x_1^2}{2a^2\sigma^2})dx_1$$

$$=\frac{a^2\sigma^2 y_1}{\sqrt{2\pi}(a^2\sigma^2+\sigma_n^2)^{3/2}}\exp(\frac{-y_1^2}{2(a^2\sigma^2+\sigma_n^2)})$$

derived in Appendix A, therefore

$$\int_{R^d} x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y}) f_{\mathbf{x}}(\mathbf{x}) |d\mathbf{x}| = \int_0^\infty 2a(4a^2 \exp(-2a^2))$$

$$\times \frac{a^2\sigma^2 y_1}{(2\pi)^{3/2}(a^2\sigma^2+\sigma_n^2)^{5/2}} \exp(\frac{-\|\mathbf{y}\|^2}{2(a^2\sigma^2+\sigma_n^2)}) da$$

Changing the variable of integration according to $t = 2a^2 + 2\sigma_n^2/\sigma^2$, using Jacobian transform $J_{a\to t} = \frac{1}{4a}$, and simplifying gives

$$\int_{R^d} x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y}) f_{\mathbf{x}}(\mathbf{x}) |d\mathbf{x}| = \int_0^\infty \frac{y_1(t-2\sigma_n^2/\sigma^2)(t-2\sigma_n^2/\sigma^2)}{(\pi\sigma^2)^{3/2}}$$

$$\exp(-t + \frac{2\sigma_n^2}{\sigma^2} - \frac{\|\mathbf{y}\|^2}{\sigma^2 t}) dt$$

$$\int_{R^d} x_1 f_{\mathbf{y}|\mathbf{x}}(\mathbf{x},\mathbf{y}) f_{\mathbf{x}}(\mathbf{x}) |d\mathbf{x}| = \frac{y_1 \exp(2\sigma_n^2/\sigma^2)}{(\pi\sigma^2)^{3/2}} [\Gamma(\frac{1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2})$$

$$-(\frac{4\sigma_n^2}{\sigma^2})\Gamma(\frac{-1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2}) + (\frac{2\sigma_n^2}{\sigma^2})^2 \Gamma(\frac{-3}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2})] \tag{8}$$

Solving (2) using (7) and (8) gives the MMSE estimator,

$$\hat{x}_1(\mathbf{y}) = y_1 \frac{[\Gamma(\frac{1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2}) - (\frac{4\sigma_n^2}{\sigma^2})\Gamma(\frac{-1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2}) + (\frac{2\sigma_n^2}{\sigma^2})\Gamma(\frac{-3}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2})]}{[\Gamma(\frac{1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2}) - (\frac{2\sigma_n^2}{\sigma^2})\Gamma(\frac{-1}{2}, \frac{2\sigma_n^2}{\sigma^2}; \frac{\|\mathbf{y}\|^2}{\sigma^2})]}$$

This shrinkage function is called *MMSE_TriShrink_radial*

4. Parameter Estimation

To apply MMSE estimator equation, we need to know noise variance (σ_n^2) and the variance of noise-free (σ^2). To estimate the noise variance from the noisy wavelet coefficients, a robust median estimator is used from the HH_1 subband [11].

$$\sigma_n^2 = (\frac{median(|HH_1|)}{0.6745})^2 \tag{9}$$

Under the assumption that marginal variance in wavelet child coefficient is different for each data point $y(k)$, an estimated $\sigma^2(k)$ can be found using local neighborhood $N(k)$. We use a square window $N(k)$ centered at $y(k)$. To compute the variance of wavelet coefficients, we use the fact that the wavelet coefficients and the additive noise are independent, thus we have the following relation between their variance:

$$\sigma^2(k) = \sigma_y^2(k) - \sigma_n^2 \tag{10}$$

where $\sigma_y^2(k)$ is the variance of noisy wavelet coefficients.

Now, assume that a priori marginal distribution $f_{\sigma_y^2}(\sigma_y^2)$ for each observed variance $\sigma_y^2(k)$ is available. Then we obtain an approximated MAP estimator for $\sigma_y^2(k)$ as

$$\sigma_y^2(k) = \arg\max_{\sigma_y^2>0} [\ln([\prod_{j\in N(k)} f(y(j)|\sigma_y^2)] f_{\sigma_y^2}(\sigma_y^2))] \tag{11}$$

In this paper, we assume $f(y(j)|\sigma_y^2)$ (pdf of noisy wavelet coefficient) is the Gaussian distribution

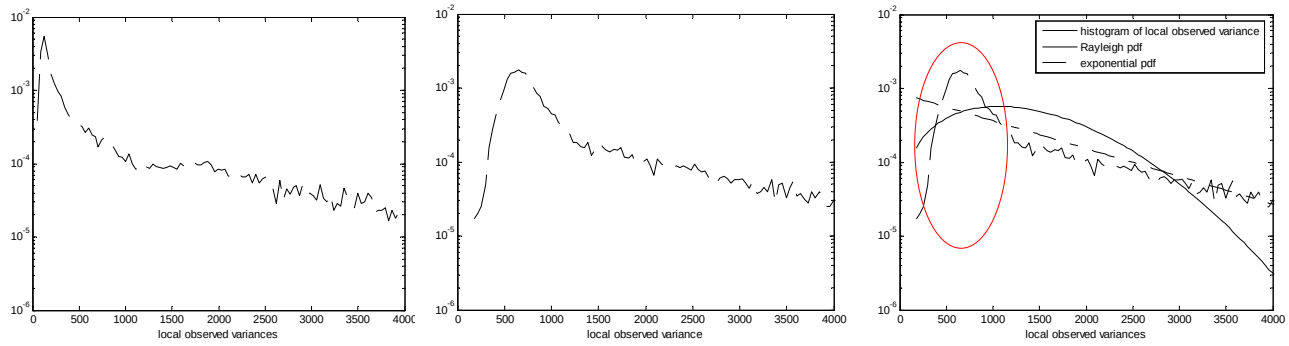


Figure 4. (a) Histogram of local observed variance in the LH subband at scale-2 of the 512x512 pixel Lena image at σ^2_n=10. (b) Histogram of local observed variance in the LH subband at scale-2 of Lena at σ^2_n=25. (c) The pdf to approximate observe variance distribution with Rayleigh pdf (solid line), exponential pdf (dash-dotted line), and Histogram of Lena image at σ^2_n=25 (dashed line).

zero mean with variance σ_y^2. The approximated MAP estimation for $\sigma^2(k)$ using a Rayleigh density priori ($f_{\sigma_y^2}(\sigma_y^2) = \lambda_1^2\sigma_y^2 \exp(\frac{-\lambda_1^2(\sigma_y^2)^2}{2}),\ \lambda_1 > 0$). As derived in Appendix B, we can write

$$(\sigma_y^2(k))^3 + (\frac{M-2}{2\lambda_1^2})\sigma_y^2(k) - \frac{\sum_{j\in N(k)} y_j^2}{2\lambda_1^2} = 0 \quad (12)$$

Coincidently, Equation (12) is also a third order equation. Therefore based on Cardano's method (The description can be found in Appendix C.) we can also obtain:

$$\sigma_y^2(k) = \sqrt[3]{C(k)} + \sqrt[3]{D(k)} \quad (13)$$

Where

$$C(k) = \frac{\sum_{j\in N(k)} y_j^2}{4\lambda_1^2} + \sqrt{\frac{(\sum_{j\in N(k)} y_j^2)^2}{16\lambda_1^4} + \frac{(M-2)^3}{216\lambda_1^6}}$$

$$D(k) = \frac{\sum_{j\in N(k)} y_j^2}{4\lambda_1^2} - \sqrt{\frac{(\sum_{j\in N(k)} y_j^2)^2}{16\lambda_1^4} + \frac{(M-2)^3}{216\lambda_1^6}},$$

giving that λ_1 is parameter of Rayleigh density, and M is number of coefficient in $N(k)$.

To select the parameter λ_1, we use the fact that under our Rayleigh density priori assumption $\sigma_y^2(k)$, computed over all coefficients should distribute according to Rayleigh density. First, the parameter λ_1 has calculated from the maximum likelihood estimation of $\sigma_y^2(k)$, that is

$$\sigma_y^2(k) = \sum_{j\in N(k)} y^2(j) \Big/ M \quad (14)$$

where [12]

$$\lambda_1 = \sqrt{\frac{2N}{\sum_{k=1}^{N} (\sigma_y^2(k))^2}}$$

where N is number of wavelet coefficients in each subband. Using (10), we finally obtain the variance of noise-free coefficient,

$$\sigma^2(k) = \max(0,\ (\sqrt[3]{C(k)} + \sqrt[3]{D(k)}) - \sigma_n^2) \quad (15)$$

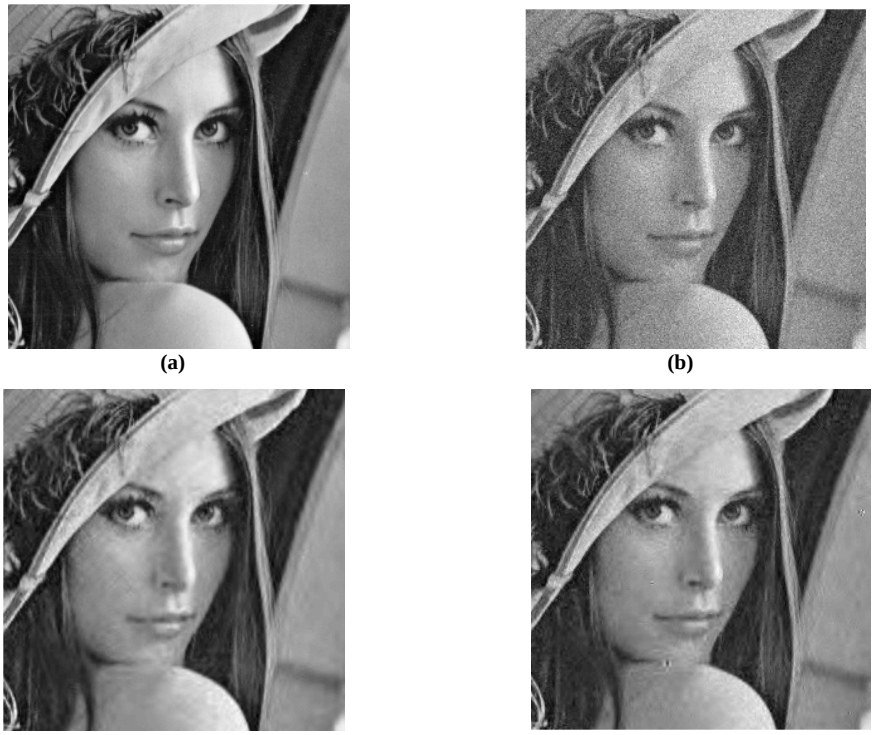
Their motivation for using Rayleigh density priori was two assumption 1) The local observed variances $\sigma_y^2(k)$ had not always concentrated at 0. (The exponential density priori developed in [13] assumed local observed variances concentred at 0 because characteristic of exponential density, $f_{\sigma_y^2}(\sigma_y^2) = \lambda_1 \exp(-\lambda_1\sigma_y^2),\ \lambda_1 > 0$). A plot of histogram of local observed variances (window size 7x7) in LH subband at scale-2 of Lena image and illustrated in Figure 3. This histogram conforms above behavior. 2) Using Rayleigh density priori requires the estimation only one additional parameter (λ_1) per image wavelet subband same using exponential density priori.

5. Experimental Results

This section presents image denoising examples in wavelet domain to show efficiency of our new model and compare it with other methods in literature. Due to space limitation, however, we give in this section results concerning two 512x512 grayscale images, namely, Lena, Barbara and two 256x256 grayscale images, namely, Cameraman and the soldier (animation image) two types of wavelet representations, namely the decimated discrete wavelet transform (DWT) and redundant dual-tree complex wavelet transform (DT-CWT). The images are obtained from USC-SIPI image database. Figure 5 shows the original cropped image Lena, its noisy version $\sigma_n^2 = 15$, and DWT-based denoised versions provide by two different methods, namely MMSE_TriShrink_ Laplace and *MMSE_TriShrink_radial*. Figure 6 shows noise-free image, noisy image, denoised image obtained using BLS-GSM, and proposed method. The soldier image is used for this propose and zero mean white Gaussian with $\sigma_n^2 = 10$ is added to the original image, DT-CWT based denoised version. We also tested our algorithm using different additive Gaussian noise levels $\sigma_n^2 = 10, 15, 20, 25$ and 30 and compared with MMSE_TriShrink_Laplace [3], and BLS_GSM [4]. The window size 7x7 are used (We have also investigated different window sizes. A 5x5 window size can also be a good choice. However, using 3x3 window size resulted in a slight performance loss. In this paper, we have not considered different square shapes for $N(k)$.) Performance analysis is done using the PSNR measure. The results can be seen in Table 1-Table 4. Each PSNR value in these tables is averaged over five runs. In these tables, the highest PSNR value is bolded.

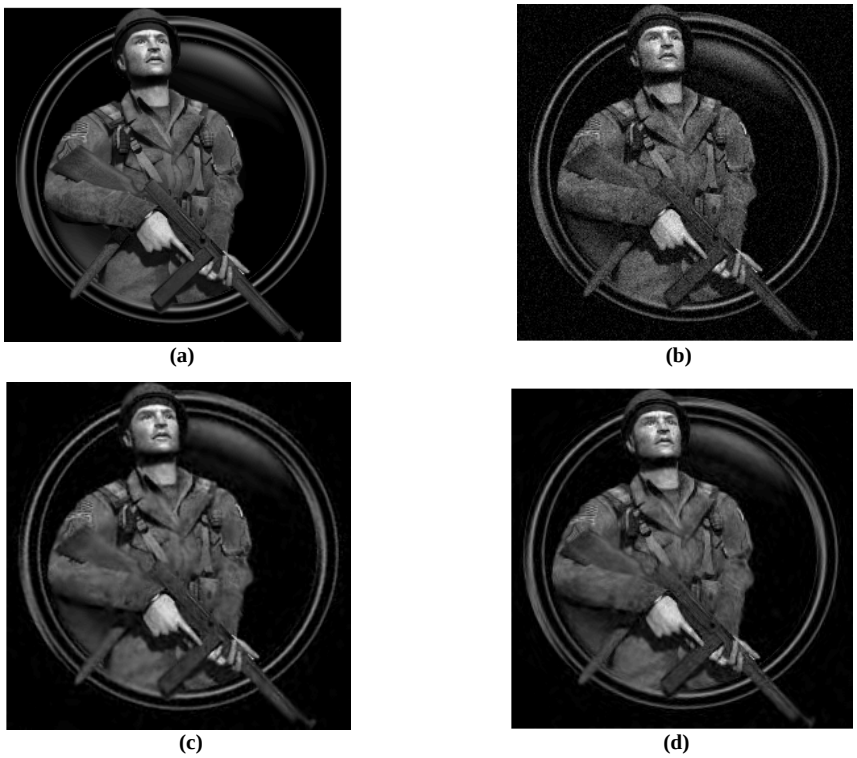
6. Discussion and Conclusion

In this paper, we present a new image denoising algorithm based on radial exponential random vectors (3-dimensional) with local variance for modeling of wavelet coefficients in each subband, namely *MMSE_TriShrink _radial*. Instead of this density model other density mod



(a) **(b)** **(c)** **(d)**

Figure 5. Comparison of the denoising images obtained from the different subband-adaptive Bayesian DWT-based denoising algorithms on Lena with σ^2_n=15: (a) Noise-free image, (b) Noisy image, (c) MMSE_TriShrink_Laplace [3] (PSNR=31.69), and (d) *MMSE_TriShrink_radial* (PSNR=31.79).



(a) **(b)** **(c)** **(d)**

Figure 6. Comparison of the denoising images obtained from the different subband-adaptive Bayesian DT-CWT-based denoising algorithms on the soldier with σ^2_n=10: (a) Noise-free image, (b) Noisy image, (c) BLS-GSM [4] (PSNR=34.00), and (d) *MMSE_TriShrink_radial* (PSNR=34.33).

Table 1. Average PSNR values of denoising image over five runs for lena image.

Denoising Algorithms	Noise variance				
	4	5	10	15	20
Decimated DWT					
MMSE_TriShrink_Laplace [3]	38.70	37.47	33.79	31.69	30.22
MMSE_TriShrink_radial	**38.78**	**37.57**	**34.00**	**31.79**	**30.53**
Redundant wavelet transform					
BLS-GSM (Steerable pyramid) [4]	**39.31**	**38.25**	**35.61**	**33.90**	**32.66**
MMSE_TriShrink_radial (DT-CWT)	38.97	38.11	34.90	33.00	31.85

Table 2. Average PSNR values of denoising image over five runs for barbara image

Denoising Algorithms	Noise variance				
	4	5	10	15	20
Decimated DWT					
MMSE_TriShrink_Laplace [3]	37.86	36.54	32.25	29.85	28.16
MMSE_TriShrink_radial	**37.92**	**36.61**	**32.35**	**29.98**	**28.19**
Redundant wavelet transform					
BLS-GSM (Steerable pyramid) [4]	**38.78**	**37.46**	**34.03**	**31.86**	**30.32**
MMSE_TriShrink_radial (DT-CWT)	37.72	36.68	33.03	30.89	29.35

Table 3. Average PSNR values of denoising image over five runs for cameraman image.

Denoising Algorithms	Noise variance				
	4	5	10	15	20
Decimated DWT					
MMSE_TriShrink_Laplace [3]	**38.32**	36.81	**32.26**	**29.66**	28.01
MMSE_TriShrink_radial	**38.32**	**36.85**	32.22	29.63	**28.05**
Redundant wavelet transform					
BLS-GSM (Steerable pyramid) [4]	**38.61**	**37.16**	**32.81**	**30.44**	**28.87**
MMSE_TriShrink_radial (DT-CWT)	38.40	37.00	32.74	30.34	28.68

Table 4. Average PSNR values of denoising image over five runs for the soldier image.

Denoising Algorithms	Noise variance				
	4	5	10	15	20
Decimated DWT					
MMSE_TriShrink_Laplace [3]	38.90	37.30	32.82	30.48	**28.80**
MMSE_TriShrink_radial	**38.97**	**37.39**	**33.01**	**30.78**	28.33
Redundant wavelet transform					
BLS-GSM (Steerable pyramid) [4]	39.51	38.09	34.00	31.81	30.33
MMSE_TriShrink_radial (DT-CWT)	**39.82**	**38.39**	**34.33**	**32.07**	**30.65**

els can be used. For example, instead of using multivariate radial exponential pdf we can use a mixture model of this pdf.

The performance of proposed technique is fairly good in terms of PSNR. In such a case, the computation load of a denoising technique increases exponentially with a complicate of technique, and that experimentally verified that it does not produce better image denoising visually results [7].

6. References

[1] Y. Zhou, S. Lai, L. Liu, and P. Lv, "An improved approach to threshold function de-noising of mobile image in CL2 multi-wavelet transform domain," IEEE Signal Processing 2000.

[2] L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," IEEE Transaction Signal Proc-

essing, Vol. 50, No. 11, pp. 2744–2756, November 2002.

[3] I. W. Selesnick "Estimation of laplace random vectors in adaptive white Gaussian noise," IEEE Transactions on Signal Processing, Vol. 56, No. 8, pp. 3482–3496. August 2008.

[4] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussian in the wavelet domain," IEEE Transaction Image Processing, Vol. 12, No. 11, pp. 1338–1351, November 2003.

[5] N. G. Kingsbury, "Image processing with complex wavelets," Phil. Transaction London A, September 1999.

[6] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," Applied Computation, Harmon, pp. 243–253., May 2001.

[7] S. M. M. Rahman, M. O. Ahmad, and M. N. S. Swamy, "Bayesian wavelet-based image denoising using the Gauss-Hermite expansion," IEEE Transaction Image Processing, Vol. 17, No. 10, pp 1755–1771, October 2008.

[8] H. Rabbani, M. Vafadust, G. Saeed and I. W. Selesnick. "Image denoising employing a bivariate Cauchy distribution with local variance in complex wavelet domain," IEEE Signal Processing, Vol. 9, pp. 203–208, 2006.

[9] M. A. Chaudhry and S. M. Zubair, "Generalized incomplete gamma functions with application," Journal of Computer Applied Mathematic, Vol. 55, No. 1, pp. 99–124, 1994.

[10] M. A. Chaudhry and S. M. Zubair, "On a class of incomplete gamma functions with applications," New York: Chapman& Hall, 2001.

[11] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," Biometrika, Vol. 81, No. 3, pp. 425–455, 1994.

[12] S. C. Choi and R. Wette, "Maximum likelihood estimation of the parameters of the gamma distribution and their bias," Technometric, Vol. 11, No. 4, pp. 683–690, 1969.

[13] M. K. Mihcak, I. Kozintsev, K. Ramchandran and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients." IEEE Signal Processing Letters, Vol. 6, No. 12, pp. 300–303, December 1999.

[14] R. W. D. Nickalls, "A new approach to solving the cubic: Cardan's solution revealed," The Mathematical Gazette, Vol. 77, pp. 354–359, 1993.

Appendix

A: Derivation of the numerator

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} x_1 \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \frac{1}{\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

Setting $u = \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2})$ and

$$dv = \frac{1}{\sqrt{2\pi\sigma_n^2}} \frac{1}{\sqrt{2\pi a^2\sigma^2}} x_1 \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

Therefore, $du = \frac{(y_1-x_1)}{\sigma_n^2} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) dx_1$ and

$$v = \frac{-a^2\sigma^2}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-x_1^2}{2a^2\sigma^2})$$

Using integrate by part formula

$\int udv = uv - \int vdu$ Therefore,

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} x_1 \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \frac{1}{\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

$$= \frac{a^2\sigma^2}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2})$$

$$+ \int_{-\infty}^{\infty} \frac{a^2\sigma^2}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \frac{(y_1-x_1)}{\sigma_n^2} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

$$= \frac{a^2\sigma^2}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2})$$

$$+ \frac{y_1 a^2\sigma^2}{\sigma_n^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

$$\frac{-a^2\sigma^2}{\sigma_n^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} x_1 \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

Thus,

$$(1+\frac{a^2\sigma^2}{\sigma_n^2}) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

$$= \frac{a^2\sigma^2}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) \Big|_{-\infty}^{\infty}$$

$$+ \frac{a^2\sigma^2 y_1}{\sigma_n^2} \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1}_{Gaussian \quad convolution}$$

Therefore,

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} x_1 \exp(\frac{-(y_1-x_1)^2}{2\sigma_n^2}) \frac{1}{\sqrt{2\pi a^2\sigma^2}} \exp(\frac{-x_1^2}{2a^2\sigma^2}) dx_1$$

$$= \frac{a^2\sigma^2 y_1}{\sqrt{2\pi}(a^2\sigma^2+\sigma_n^2)^{3/2}} \exp(\frac{-y_1^2}{2(a^2\sigma^2+\sigma_n^2)})$$

B: Derivation of the Approximated MAP estimation for $\sigma_y^2(k)$ using Rayleigh Density Priori for observed variance and Gaussian distribution for noisy wavelet coefficients

When $f(y(j)|\sigma_y^2)$ s pdf of the Gaussian distribution with zero mean and variance σ_y^2, $(f(y(j)|\sigma_y^2)=\frac{1}{\sqrt{2\pi\sigma_y^2}}\exp(\frac{-y^2(j)}{2\sigma_y^2}))$ and $f_{\sigma_y^2}(\sigma_y^2)$ is density of Rayleigh distribution $(f_{\sigma_y^2}(\sigma_y^2)=\lambda_1^2\sigma_y^2\exp(\frac{-\lambda_1^2(\sigma_y^2)^2}{2}),\lambda_1>0)$, thus

$$\prod_{j\in N(k)} f(y(j)|\sigma_y^2) f_{\sigma_y^2}(\sigma_y^2) =$$

$$(\frac{1}{\sqrt{2\pi\sigma_y^2(k)}})^M \exp(\frac{-\sum_{j\in N(k)} y_j^2}{2\sigma_y^2(k)})\lambda_1^2\sigma_y^2(k)\exp(\frac{-\lambda_1^2(\sigma_y^2(k))^2}{2})$$

$$\ln[(\prod_{j\in N(k)} f(y(j)|\sigma_y^2)) f_{\sigma_y^2}(\sigma_y^2)] =$$

$$\frac{M}{2}\ln(\frac{1}{2\pi})-\frac{M}{2}\ln(\sigma_y^2(k))-\frac{\sum_{j\in N(k)} y_j^2}{2\sigma_y^2(k)}+\ln(\lambda_1^2)$$

$$+\ln(\sigma_y^2(k))-\frac{\lambda_1(\sigma_y^2(k))^2}{2}$$

In order to find $\sigma_y^2(k)$, we use (11). Then

$$\frac{\partial\ln[(\prod_{j\in N(k)} f(y(j)|\sigma_y^2)) f_{\sigma_y^2}(\sigma_y^2)]}{\partial(\sigma_y^2(k))}=0$$

$$(\sigma_y^2(k))^3+(\frac{M-2}{2\lambda_1^2})\sigma_y^2(k)-\frac{\sum_{j\in N(k)} y_j^2}{2\sigma_y^2(k)}$$

Finally, we use Cardano's method to find optimum $\sigma_y^2(k)$.

C: Cardano's method

In mathematic, a cubic function is a function of the form

$$f(x)=c_4x^3+c_1x^2+c_2x+c_3,$$
$$c_1,c_2,c_3,c_4 \in R, c_4\neq 0$$

We first normalize this standard equation by dividing the equation with the first coefficient. Thus, we can write,

$$x^3+ax^2+bx+c=0$$

In summary, if we use cardano's method [14] to solve the above equation, the roots of the cubic equation will be

$$x=\begin{cases} \sqrt[3]{E}+\sqrt[3]{F}-\frac{a}{3} \\ -\frac{1}{2}(\sqrt[3]{E}+\sqrt[3]{F})\pm\frac{\sqrt{3}}{2}j(\sqrt[3]{E}+\sqrt[3]{F})-\frac{a}{3} \end{cases}$$

Where

$$p=b-\frac{a^2}{3},\quad q=c-\frac{ab}{3}+\frac{2a^3}{27}$$

$$E=\frac{-q}{2}+\sqrt{\frac{q^2}{4}+\frac{p^3}{27}},\quad F=\frac{-q}{2}-\sqrt{\frac{q^2}{4}+\frac{p^3}{27}}$$

Only a real root will be used here.

Wireless Sensor Network, **2009, 1, 294-299**
doi:10.4236/wsn.2009.14036 Published Online November 2009 (http://www.scirp.org/journal/wsn).

A New Method for Anti-Noise FM Interference

Changyong JIANG, Meiguo GAO, Defeng CHEN
Department of Electronics Engineering, Beijing Institute of Technology, Beijing, China
Email: *jieshi*08*@gmail.com*
Received May 30, 2009; *revised June* 20, 2009; *accepted June* 21, 2009

Abstract

Noise Frequency Modulated (NFM) interference causes a disaster to almost all types of Radar systems. The echo signal and the interference are overlapped and because of strong energy of the NFM interference nothing could be detected except the interference in the Radar receiver system. Up to now no good method against NFM has been declared, conventional methods are based on the passive Radar to track the interference source which are not applicable under most conditions. Here a novel anti-noise FM method is proposed to suppress the NFM interference, the method multiply the mixed signal two times by different reference signals. The principle and some key factors of the new method are analyzed in detail and some rules for parameters designing are given. What's more, results show that the method can eradicate NFM effectively.

Keywords: ECM, ECCM, NFM Interference, Anti-NFM

1. Introduction

FM jamming is a common jamming forms in oppressive jamming on radar systems [1–2]. Noise FM is a mostly used ECM method, and can cause disasters to nearly all types of Radar systems. So, analysis of the performance of NFM in ECM and solutions against NFM in ECCM has developed for years. [3] Proposed the noise FM jamming method. The effect of noise FM jamming against ISAR one or two-dimensional imaging is described in detail, and the power requirement of noise FM jamming is compared with that of RF noise jamming. Song [4] uses growth factor analyzed the capability of radar MTI in noise FM jamming. Liu [5] uses signal to jamming ratio (SJR) gains discussed the performance of anti-noise FM jamming of PRC-BPM fuze. [6] uses the effect of Doppler frequency, pseudo-random code width, the effect of period pseudo-random code serial and aiming frequency deviation analyzed the performance of the pseudo-random code binary phase modulated (PRC-BPM) fuze. Xu [7] gave methods of multipath jammer tracking with a passive radar seeker. Chen [8] studied the formula of composite phase-difference of two noise FM jamming. Deergha Rao. K. [9] presented an approach based on jammer instantaneous frequency estimation for suppression of frequency modulated jammers in spread spectrum systems. [10] Focus on Subspace Projection Technique for suppression of jamming in narrowband FM jammers, however, NFM interference is a wideband interference that this technique in [10] cannot be applied. [11] presented performance analysis of subspace projection array processing techniques for suppression of frequency modulated (FM) jammers in GPS receivers, and based on this [12] made the approach applied to AM-FM jammers as well, however, the subspace projection techniques are not available under some radar receivers. [13] offered a method against NFM based on Square Transformation, however, it is only described in the application of Pseudo-random Coded Fuze and analog circuits. Above all, these researches made big progress in finding solutions against NFM, but the methods cannot totally resolve the problem in the radar systems.

Based on all the previous researches, a method is proposed to eliminate NFM in this paper. It multiply the mixed signal by two different reference signals two times and with followed signal processing the needed signal can be obtained from the output.

The paper is organized as follows. In Section 2, the echo signal model and NFM interference model are described in detail. Section 3 depicts how the new method supposed here excise NFM interference and some key factors of the method are analyzed. Section 4 gives the performance analysis of the method. And some conclusions are given in the last section.

2. Signal Model

Noise FM is a commonly used method for jamming wireless communication systems such as Radar systems, GPS etc. It has a strong suppress to the needed signal and

its bandwidth is much wider than the needed signal. The noise-FM is modeled as1

$$s_{NFM}(t) = A_i \cos\left(\omega_{ci} t + k\int_0^t f(\tau)\,d\tau\right) \quad (1)$$

where A_i is the amplitude of NFM interference, ω_{ci} is the carrier frequency of NFM interference, and k is the FM slope. The bandwidth of the NFM interference is BW_i.

The echo signal from the target is defined as

$$s_{use}(t) = A_u \cos\left(\omega_c t + k_0 \pi t^2\right) \quad (2)$$

where A_u is the amplitude of the echo signal, ω_c is the carrier frequency of the echo signal, and k_0 is the FM slope (k_0=0 when the signal $s_{use}(t)$ is CW and $k_0 \neq 0$ when the signal $s_{use}(t)$ is chirp). The bandwidth of the echo signal is BW. It is known that in order to make the interference more effective, $\omega_{ci} \approx \omega_c$, $A_i \gg A_u$ and $BW_i \gg BW$ must be satisfied. Thus it is hard to obtain the needed signal $s_{use}(t)$ neither from time domain nor frequency domain.

Without loss of generality, mixed signal which enters the radar receiver is defined as

$$s(t) = s_{NFM}(t) + s_{use}(t) \quad (3)$$

where $s_{NFM}(t)$ is defined in Equation (1), $s_{use}(t)$ is defined in Equation (2).

3. Nfm Excision

3.1. Basic Concept

This part simply shows what the new method derivate from. It is supposed that two variables, "*a*" and "*b*" are here. How to change each other's value without any other variable? A simple description of solving this question is shown below.

First, let

$$a = a + b \quad (4)$$

Now the value of variable "*a*" becomes sum of "*a*" and "*b*", the value of variable "*b*" remains the same.

Second, let

$$b = a - b \quad (5)$$

then the value of variable "*a*" remains the same, the value of variable "*b*" becomes the value of "*a*" which is before Equation (4).

Last, let

$$a = a - b \quad (6)$$

And the aim of changing values of "*a*" and "*b*" is reached.

Similarly, if two signals are mixed together, it is possible to separate them in the same way above.

3.2. Principle of the New Method

As is known to all, it is easy to get two signals added with each other in the frequency domain just by multiplying each other. Two signals multiplied with each other in the time domain means that their frequencies are added with each other in the frequency domain. In this way the new method contains two steps which mainly consist of two multiplications, so it is called "double-multiplication" method in the next chapters.

3.2.1. The First Step of Double-Multiplication Method

The first step of double-multiplication method can be seen from Figure 1. It contains a multiplication, a low pass filter and DC blocked module. The multiplication is

$$s_{M0}(t) = s(t) \cdot s(t) \quad (7)$$

And all parts obtained after this multiplication are as follows,

- direct current:

$$s_{DC}(t) = A_u^2/2 + A_i^2/2 \quad (8)$$

- low frequency part:

$$s_{M1}(t) = A_i A_u \cos\left(k\int_0^t f(\tau)\,d\tau + (\omega_{ci} - \omega_c)t - k_0\pi t^2\right) \quad (9)$$

- the part whose carrier frequency is nearly twice as large as ω_c:

$$s_{M01}(t) = \left(A_u^2/2\right)\cos(2\omega_c t) \quad (10)$$

$$s_{M02}(t) = \left(A_i^2/2\right)\cos\left(2\omega_{ci} t + 2k\int_0^t f(\tau)\,d\tau\right) \quad (11)$$

$$s_{M03}(t) = A_i A_u \cos\left((\omega_{ci} + \omega_c)t + k\int_0^t f(\tau)\,d\tau + k_0\pi t^2\right) \quad (12)$$

After $s_{M0}(t)$ goes through a low pass filter and DC blocked module, signals described in Equation (8), (10), (11), (12) are filtered and only $s_{M1}(t)$ is left.

3.2.2. The Second Step of Double-Multiplication Method

The second step of double-multiplication method can be seen from Figure 1. It contains a multiplication, a band pass filter. The second multiplication is

$$s_{M2}(t) = s(t) \cdot s_{M1}(t) \quad (13)$$

And all the parts obtained are as follows.

- The parts whose carrier frequency is nearly the same with ω_c:

$$s_{M21}(t) = A_u^2 A_i \cos\left((2\omega_c - \omega_{ci})t + 2k_0\pi t^2 - k\int_0^t f(\tau)\,d\tau\right) \quad (14)$$

$$s_{M22}(t) = A_u^2 A_i \cos\left(\omega_{ci} t + k\int_0^t f(\tau)d\tau\right) \quad (15)$$

$$s_{M23}(t) = A_i^2 A_u \cos\left(\omega_c t + k_0 \pi t^2\right) \quad (16)$$

$$s_{M24}(t) = A_i^2 A_u \cos\left((2\omega_{ci} - \omega_c)t + 2k\int_0^t f(\tau)d\tau - k_0\pi t^2\right) \quad (17)$$

It is obvious that $s_{M23}(t)$ is the needed signal which just only has a different amplitude with $s_{use}(t)$. Let $s_{M2}(t)$ go through a band pass filter (BPF) which has a center frequency of f_c and a bandwidth of BW. As is known that $A_i >> A_u$, $s_{M21}(t)$ and $s_{M22}(t)$ can be omitted compared to $s_{M23}(t)$. What's more, as $BW_i >> BW$ and $s_{M24}(t)$ has a bandwidth of $2BW_i$, $s_{M24}(t)$ left little after the band pass filter. Thus the needed signal is obtained.

The whole process of the method is shown in Figure 1.

3.3. Analysis of Key Factors in Double-Multiplication Method

The output signal, $s_{out}(t)$, from the process of double-multiplication method contains four parts which are described in Equation (14), (15), (16) and (17). Usually the higher the interference to signal ratio (ISR) is, the more effectively the interference works; and the wider the bandwidth of NFM interference is, the more effectively the interference works. So the two factors, ISR and the bandwidth of NFM interference, are analyzed as follows.

3.3.1. Interference to Signal Ratio

1) Relationship between ISR and SIR

The interference to signal ratio (ISR) is defined as

$$ISR = 20\log 10(A_i / A_u) \quad (18)$$

From Equation (18) it is known that the higher the ISR is, the bigger the A_i/A_u is. According to Equation (15), (16) and (17), among the output signal $s_{M2}(t)$ the needed signal to interference ratio is

$$\begin{aligned} SIR &= 10\log 10\left\{P\left[s_{M23}(t)\right] / \left[P\left(s_{M22}(t)\right) + P\left(s_{M24}(t)\right)\right]\right\} \\ &= 20\log 10\left\{\left(A_i^2 A_u\right) / \left[\left(A_u^2 A_i\right) + \left(A_i^2 A_u\right)\left(BW / BW_i\right)\right]\right\} \\ &= 20\log 10\left[1 / \left(A_u / A_i + BW / BW_i\right)\right] \end{aligned} \quad (19)$$

For a certain bandwidth of NFM interference, the BW/BW_i is a constant. As ISR increases, A_i/A_u also increases, thus SIR described in Equation (19) increases, too. Hence a conclusion is obtained: the higher the ISR is, the bigger the SIR is, which means that the higher the ISR is, the more efficient the new method is.

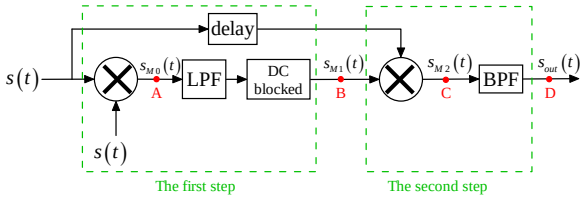


Figure 1. Structure of double-multiplication method.

2) Relationship between ISR and low pass filter

Within the first step of double-multiplication method, after multiplication the power of $s_{M02}(t)$ to $s_{M1}(t)$ ratio is

$$s_{M02}s_{M1}R = 20\log 10\left(A_u / \left(2A_i\right)\right) \quad (20)$$

Although the carrier frequency of $s_{M02}(t)$ is nearly twice as large as ω_c and $s_{M02}(t)$ is out of the passband of LPF, if the stopband attenuation of LPF is smaller than $s_{M02}1s_{M1}R$, the interference signal $s_{M02}(t)$ may not be filtered from $s_{M1}(t)$ by the low pass filter. So the design of stopband attenuation of LPF must be bigger than $s_{M02}1s_{M1}R$ dB. Hence another conclusion is obtained: the higher the ISR is, the bigger the stopband attenuation of LPF must be.

Above all, two conclusions related to ISR are obtained as follows.

- ✧ The higher the ISR is, the more efficient the double-multiplication method is. The higher the ISR is, the bigger the stopband attenuation of LPF must be.

3.3.2. Bandwidth of NFM Interference

For a certain ISR, the A_u/A_i is a constant. As BW_i, the bandwidth of NFM increases, SIR described in Equation (19) increases, too. So the bigger the bandwidth of NFM interference is, the higher the SIR is, which means that the more efficient the double-multiplication method is.

However, the bandwidth of NFM interference is unknown under most conditions. Thus the stopband of the low pass filter cannot be decided. If the stopband of the LPF is smaller than the bandwidth of NFM interference, the output of the first step, i.e. $s_{M1}(t)$, may not be correctly obtained. There are two ways to solve this problem: 1) Measuring the bandwidth of NFM interference if the receiver system is capable of this; 2) Designing the LPF with a high stopband as possible as the receiver system can.

Above all, conclusions related to bandwidth of NFM interference are drawn as follows.

- ✧ The bigger the bandwidth of NFM interference is, the more efficient the new method is.
- ✧ When bandwidth of NFM interference is unknown, it is better to measure the bandwidth of NFM interference, otherwise to design the LPF with a high stopband as possible as the receiver system can.

4. Performance Analysis

Without loss of generality, considering the echo signal is CW signal and A_{use}=1, f_c=100.2*MHz*. And simulation results are depicted as follows.

It is supposed that the bandwidth of NFM interference is known as 20*MHz* and ISR is 40dB. Simulation results can be seen from Figure 2 and Figure 3. In Figuer 2 the graph above is the spectrum of the mixed signal which contained the echo signal and the NFM interference, the graph below shows the spectrum of the signal which is the output (at dot "B" in Figure 1) after the first step. In Figure 3 the graph above is the spectrum of signal which is the output (at dot "C" in Figure 1) after the second multiplication, the graph below shows the spectrum of signal which is the output (at dot "D" in Figure 1) after the whole process of double-multiplication method.

From the graph above in Figure 2 it is obvious that the echo signal and NFM interference are overlapped with each other and the echo signal cannot be distinguished from the interference. However, the graph below in Figure 3 shows that the output signal is mainly the echo signal after the process of double-multiplication method.

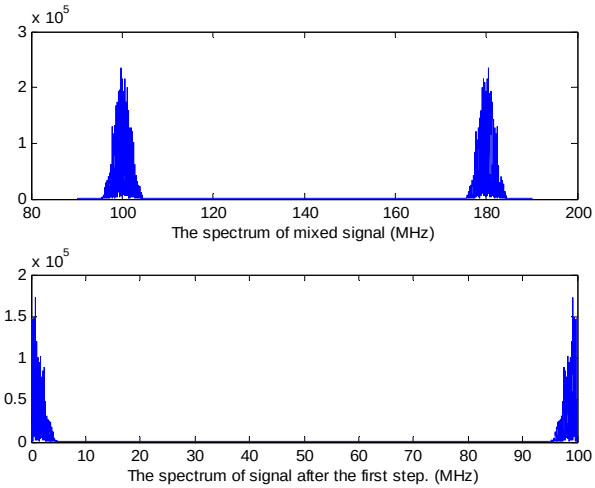


Figure 2. Spectrum of signals at different time.

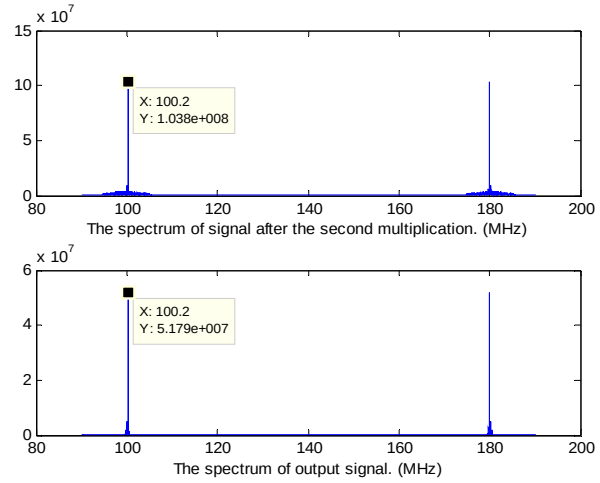


Figure 3. Spectrum of signals at different time.

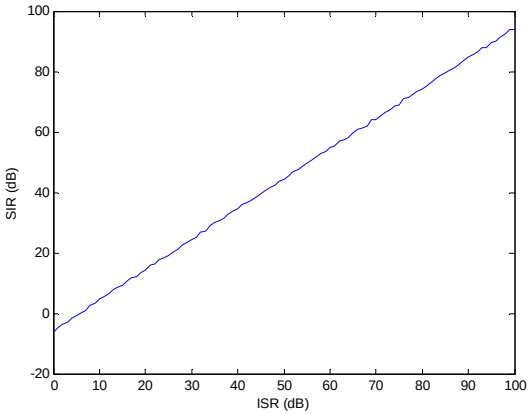


Figure 4. SIR at different ISR.

4.1. Simulation of ISR

As mentioned above, when the ISR increases the stopband attenuation of LPF increases and the SIR increase. These two relationships are simulated as follows.

4.1.1. The Relationship between ISR and SIR

It is supposed that the ISR varies from 0~100dB, the bandwidth of NFM interference is known as 20MHz, and the bandwidth of band pass filter (BPF) is 1MHz.

Figure 4 shows the SIR at different ISR. It is obvious that the larger the ISR is, the larger the SIR is, which confirms the conclusion obtained above.

4.1.2. Design of Stopband Attenuation of LPF

Consider the ISR varies from 0~100dB, the stopband attenuation of LPF are 20dB and 100dB. The graph below in Figure 5 shows the results when the stopband attenuation of is 20dB and the graph above shows the results when the stopband attenuation of is 100dB. It is obvious that a small stopband attenuation of LPF will cause errors to the output and high ISR needs large stopband attenuation of LPF, which also confirms the conclusion above.

4.2. Bandwidth of NFM Interference

Consider the bandwidth of NFM interference varies from 2~40MHz, ISR is 40dB.

4.2.1. Bandwidth of NFM Interference is Known

As the bandwidth of NFM interference is known, the stopband of LPF is bigger than all the bandwidth of NFM. Figure 6 shows the frequency of output signal with different bandwidth of NFM interference. And a conclusion is obtained: if the stopband of LPF is larger than the bandwidth of NFM interference, the needed signal can be got correctly.

4.2.2. Bandwidth of NFM Interference is Unknown

As the bandwidth of NFM interference is unknown, it is supposed that it is 2MHz and 20MHz. Figure 7 shows the obtained frequency when the stopband of LPF is 2MHz and Figure 8 shows the obtained frequency when the stopband of LPF is 20MHz.

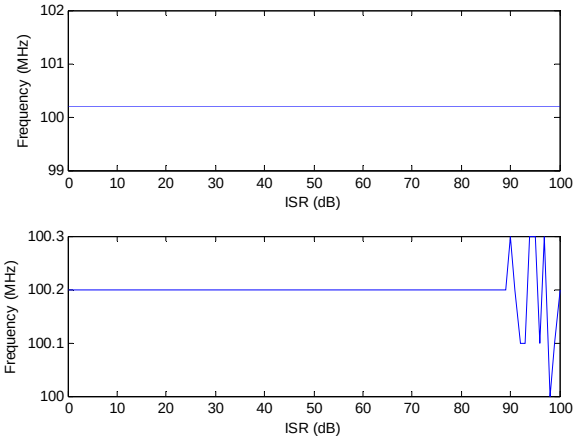


Figure 5. Frequency of output at different ISR.

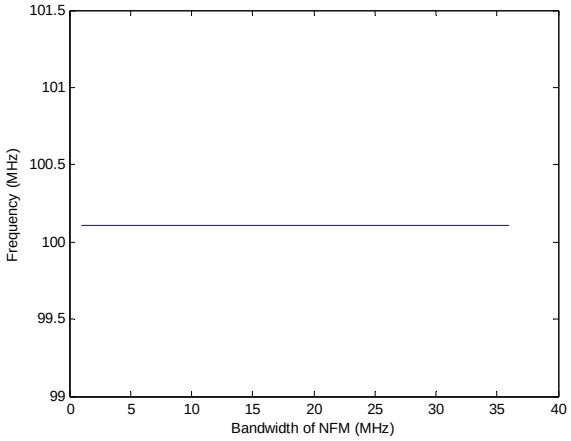


Figure 6. Frequency of output signal.

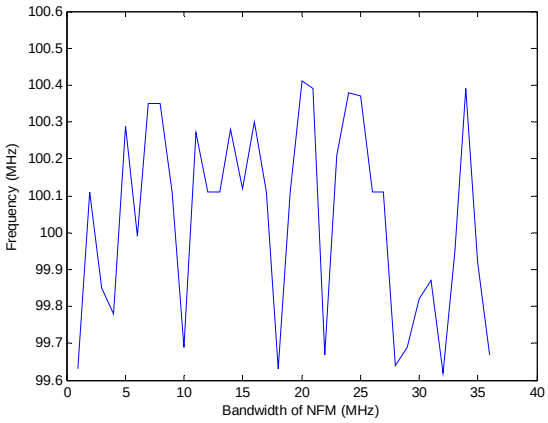


Figure 7. Frequency of output signal.

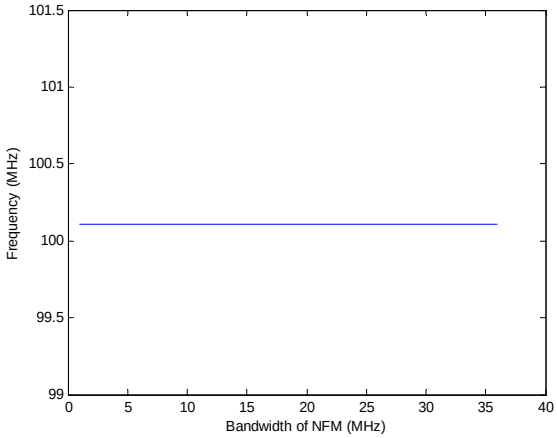


Figure 8. Frequency of output signal.

From Figure 7 and Figure 8, it is known that when the bandwidth of NFM interference is unknown, to design the stopband of LPF as big as possible will help to make the method more efficient.

5. Conclusion

NFM interference can suppress the useful signal both in the time domain and in the frequency domain. The new method supposed in this paper can eliminate the effect of the NFM interference. Some conclusions are obtained:

- ✧ The higher the ISR is, the more efficient the double-multiplication method is.
- ✧ The higher the ISR is, the bigger the stopband attenuation of LPF must be.
- ✧ The bigger the bandwidth of NFM interference is, the more efficient the double-multiplication method is.
- ✧ When bandwidth of NFM interference is unknown, it is better to measure the bandwidth of NFM interference if possible, otherwise to design the LPF with a high stopband as possible as the receiver system can.

Further studies will focus on the signal model for SAR/ISAR and the real application on radar systems or other communication systems.

6. References

[1] G. S. Liu, X. Q. Shi, J. H. Lu, etc. “Design of noise FM-CW radar and its implementation,” IEE Proc–F, Vol. 138, No. 5, pp. 420–426, 1991.

[2] J. H. Lu and X. Q. Shi, “The study on the stationariness and the ergodicity of zero IF signal in Noise FM-CW Radar,” Modern Radar, No. 4, pp. 28–36, 1992.

[3] C. X. Dong, S. Q. Yang, G. Q. Zhao, and Y. Zhang, “Effect of noise FM jamming against ISAR imaging,”

CIE International Conference of Radar Proceedings, 2007.

[4] C. J. Song and J. Y. Zhang, "The analysis of the performance of radar MTI in noise FM jamming environment," Proceeding of ICSP2000, Beijing, pp. 1947–1950, 2000.

[5] J. B. Liu, L. J. Wang, and H. C. Zhao, "Performance analysis of anti-noise FM jamming of pseudo-random code fuzes," Journal of Electronics & Information Technology, Vol. 26, No. 12, pp. 1925–1932, 2004.

[6] X. G. Zhou, H. C. Zhao, and Y. C. Tu, "Performance analysis of anti-noise FM jamming of pseudo-random code binary phase modulation fuze based on Doppler effect," IEEE 2007 International Sym posium on Microwave, Antenna, Propagation, and EMC Technologies for wireless communications, pp. 1424–1427.

[7] S. T. Xu, and S. Q. Yang, "Multipath jammer tracking with a passive radar seeker," Systems Engineering and Electronics, Vol. 25, pp. 31–109, 2003.

[8] X. Q. Chen, "Direction-finding technology of two noise FM jamming for passive seeker," Modern Defense Technology, Vol. 34, pp. 59–62, 2006.

[9] K. D. Rao, etc., "Instantaneous frequency based nonlinear adaptive filter for interference suppression in spread spectrum systems," The 47th Midwest Symposium on Circuits and Systems, 2004.

[10] Monfared, Mohsen Tavoosi, Yargholi, and Mostafa, "Suppression of jamming in GPS receivers using subspace projection technique," Proceedings of the Fifth IASTED International Conference on Communications, Internet, and Information Technology, CIIT 2006.

[11] M. G. Amin, etc., "Performance analysis of subspace projection techniques for anti-jamming GPS using spatio-temporal interference signatures," IEEE Workshop on Statistical Signal Processing Proceedings, 2001.

[12] S. C. Jang, Loughlin, and J. Patrick, "AM-FM interference excision in spread spectrum communications via projection filtering," Eurasip Journal on Applied Signal Processing, 2001.

[13] S. N. Zhang, H. C. Zhao, and G. Xiong, "The method of excising noise frequency modulation interference for pseudo-random coded fuze based on square transformation," Journal of Missile and Homing, 2006.

Wireless Sensor Network, **2009, 1, 300-305**
doi:10.4236/wsn.2009.14037 Published Online December 2009 (http://www.scirp.org/journal/wsn).

Blending Sensor Scheduling Strategy with Particle Filter to Track a Smart Target

Bin LIU[1], Chunlin JI[1], Yangyang ZHANG[2], Chengpeng HAO[3]
[1]*Department of Statistical Science, Duke University, Durham, U. S. A*
[2]*Adastral Park Research Campus, University College London, London, UK*
[3]*Institute of Acoustics, Chinese Academy of Sciences, Beijing, China*
Email: {*bin.liu2, chunlin.ji*}*@duke.edu, y.zhang@adastral.ucl.ac.uk, haochengp@sohu.com*
Received April 17, 2009; *revised July* 20, 2009; *accepted July* 21, 2009

Abstract

We discuss blending sensor scheduling strategies with particle filtering (PF) methods to deal with the problem of tracking a 'smart' target, that is, a target being able to be aware it is being tracked and act in a manner that makes the future track more difficult. We concern here how to accurately track the target with a care on concealing the observer to a possible extent. We propose a PF method, which is tailored to mix a sensor scheduling technique, called covariance control, within its framework. A Rao-blackwellised unscented Kalman filter (UKF) is used to produce proposal distributions for the PF method, making it more robust and computationally efficient. We show that the proposed method can balance the tracking filter performance with the observer's concealment.

Keywords: Particle Filter, Sensor Scheduling, Smart Target, Tracking

1. Introduction

The problem of target tracking has received considerable attention from both academic and engineering communities. Generally, people formulate this problem as a state estimation or filtering problem, focusing on the tracking filter's performance. A limitation of such works in the literature is that it assumes that any changes in the behavior of the target are unconnected to the action of the target tracking process. However, in many real-world situations, this is an unrealistic assumption. Taking a sonar application as an example, where the observer is an autonomous underwater vehicle (AUV), the target, e.g., a submarine equipped with elaborate detection instruments, is able to detect, and, once it is aware it is being tracked, it can modify its behavior quickly to escape from this track and make the future track more difficult.

A complete solution to the problem of tracking a smart target is still an open problem. However, some initial results are available. Kreucher *et al* perform a reinforcement learning approach to schedule a multi-modality sensor to detect and track smart targets [1]. For their approach, a multi-step ahead scheduling policy is essential to provide sensible performance. Savage et al. consider an idealized problem where the target has a set of possible motion models and selects the one to best reduce the sensor's tracking performance, and treat this problem in the framework of game theory [2]. Gittins and Roberts use game theory to investigate the case in which a target is trying to escape detection [3,4]. We consider the problem of tracking a smart target with a care on concealing the observer to an extent and propose a smart tracker by mixing a sensor scheduling technique with particle filtering (PF) methods [5].

This paper is an expanded version of [5]. Here we assume there are two sensors to be used by the observer, with passive and active modalities, respectively. The passive sensor measures the energy that has already existed in the environment, without emitting any energy outside. Such a quite mode makes the observer conceal itself well, but cannot guarantee the tracking performance, especially when the SNR is small. Differing from the passive sensor, the active one emits energy to the environment and before collecting reflected energy to do detection. Such an active mode has substantially better detection and tracking capabilities than the passive one,

This material is based upon work supported by the National Science Foundation of the USA under Grant No. 0507481. C. Hao's work was supported by the National Natural Science Foundation of China under Grant No. 60802072.

however, it makes the observer easily detected by the target. So, employing these two sensors, there is a contradiction between the tracking filter's performance and the concealment of the observer. The goal of this paper is actually to design a method, which can both guarantee the tracking filter's performance and conceal the observer to a reasonable extent. We resort to sensor scheduling strategies and particle filtering (PF) methods to seek a balance between these two aspects.

Sensor job (time) scheduling is within the context of multi-sensor management. It has become increasingly important in the research and development of modern multi-sensor systems. Sensor scheduling lies in the first level of a top-down policy of sensor management with the role of assigning each sensor with a detailed schedule on what to do [6]. PF is a Sequential Monte Carlo method which founds great research and applications in the last decade (see [7–9] and references therein). It beats Kalman filter, a classical method used in the target tracking discipline, in dealing with nonlinear dynamical and measurement models and non-Gaussian noises in the model. Theoretically, employing enough particles, PF can provide an approximate optimal Bayesian solution to any state-space based estimation problem. In this paper we mix a specific sensor scheduling technique, namely covariance control [10], with PF methods to deal with the problem of tracking a smart target. We use a Rao-blackwellised unscented Kalman filter (UKF) [11] to produce proposal distributions for the PF, making it more robust and computationally efficient. It is shown that the proposed method provides a balance between the tracking filter's performance and the observer's concealment, hence it satisfies our needs for the problem under consideration.

The remaining of this paper is organized as follows. Section 2 describes the dynamic models involved. Section 3 presents the sensor scheduling technique, covariance control. The proposed PF algorithm is illustrated in Section 4 and its performance is evaluated in Section 5. Finally we conclude this paper in Section 6.

2. Models

In this section, we describe the models involved in this paper. First the dynamic model for the target is presented. Then the measurement models are derived for both the passive and the active sensors.

The evolution of the target state, $\mathbf{x}_k$, is modeled by a discrete time linear Gaussian:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + v_k \tag{1}$$

where $v_k \sim N(0,\mathbf{Q})$. Here the target state vector is composed of the position and velocity items in the x and y coordinates and is defined as follows:

$$\mathbf{x}_k = \begin{bmatrix} x_{t,k} & \dot{x}_{t,k} & y_{t,k} & \dot{y}_{t,k} \end{bmatrix}^T \tag{2}$$

where the dot denotes the operation of first order derivative and the superscript T denotes transposition of a matrix. We use a constant-velocity process model for the target, so that

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_s & 0 \\ 0 & \mathbf{F}_s \end{bmatrix}, \mathbf{F}_s = \begin{bmatrix} 1 & \mathrm{T} \\ 0 & 1 \end{bmatrix} \tag{3}$$

and

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_s & 0 \\ 0 & \mathbf{Q}_s \end{bmatrix}, \mathbf{Q}_s = q_s \begin{bmatrix} \mathrm{T}^3/3 & \mathrm{T}^2/2 \\ \mathrm{T}^2/2 & \mathrm{T} \end{bmatrix} \tag{4}$$

where T is the sampling period, q_s is the power spectral density of the acceleration noise in the spatial dimensions.

Defining $\varphi_k = (x_{o,k}, y_{o,k})$ as the observer's position at time step k, we derive measurement functions for both the passive and the active sensors in the following. We consider the case where the passive sensor only provides relative bearings measurements originated from the target, then the associated measurement function is

$$\mathbf{z}_k = \operatorname{atan}\left(\frac{x_{t,k} - x_{o,k}}{y_{t,k} - y_{o,k}}\right) + n_k \tag{5}$$

where $n_k \sim N(0,\mathbf{R}_b)$.

We assume the observer adopts a track-while-scan sensor [10] to do active sensing, which can measure both the bearings and the ranges. The associated measurement function is denoted as

$$\mathbf{z}_k = \left[\operatorname{atan}\left(\frac{x_{t,k} - x_{o,k}}{y_{t,k} - y_{o,k}}\right), \sqrt{(x_{t,k} - x_{o,k})^2 + (y_{t,k} - y_{o,k})^2}\right]^T + \left[n_k, r_k\right]^T \tag{6}$$

where $r_k \sim N(0,\mathbf{R}_d)$ denotes the noise item in the range. So the covariance matrix of the measurement noise is $diag[\mathbf{R}_b, \mathbf{R}_d]$. Here $diag$ denotes the operation of diagonalization.

3. A Sensor Scheduling Technique: Covariance Control

In this section we present the sensor scheduling technique, called covariance control, which will be embedded in the PF framework described in Section 4.

Covariance control begins with a desired covariance matrix, which is this approach differs from many other sensor management algorithms. A desired covariance matrix for an n-dimensional state estimate, $\mathbf{P}_D$, is defined by all $n \times n$ elements of that matrix. The goal is to find a specific sensor combination i that produces covariance matrix $\mathbf{P}^i$, assuring the difference $\mathbf{P}_D - \mathbf{P}^i$ is positive semi-definite. To properly evaluate that difference, a scalar metric is needed. A variety of these exist, including functions based on the determinant or the trace of the matrix. However, these metrics rely on the positive definiteness of the matrix to provide accurate evaluations. If a difference is only semi-definite, then the determinant is zero, possibly masking a large difference in a different direction (note that a covariance can be represented as an ellipsoid, whose axes directions can be indicated by the eigenvectors of the covariance matrix). A similar problem exists with the trace, where a large positive difference can mask a large negative difference along a different direction. To avoid these problems, M. Kalandros and L. Y. Pao, examined other techniques, such as the eigenvalue/minimum sensors algorithm, the matrix norm algorithm and the norm/sensors algorithm [10]. The norm/sensors algorithm relaxes the requirements of the matrix norm technique, allowing the norm of the covariance difference to vary within a predefined boundary $\pm\delta$. So we borrow the idea of the norm/sensors algorithm and propose the following sensor scheduling strategy:

• If

$$\left\| \mathbf{P}_{\mathrm{D}} - \mathbf{P}_k \right\|_2 > \delta \tag{7}$$

select the active sensor to work for next time step;

• Else

select the passive sensor to work for next time step.

($\mathbf{P}_k$ denotes the covariance matrix associated with the estimate for the target state at the *k* th time step)

Note that the aim of this sensor scheduling strategy is to select an appropriate sensor for use for next iteration of the tracking process, other than to search a sensor combination that can work with the fewest sensors involved, which is the purpose of the methods proposed in [10].

4. Particle Filtering Algorithm

This section presents our proposed PF algorithm. First we give a brief introduction for a basic PF method. Then we describe the Rao-blackwellised UKF [11], which is used to produce proposal distributions for our PF method. Finally we mix the sensor scheduling technique presented in Section 3 with the PF algorithm, leading to the proposed method for tracking a smart target.

Particle filter is a Sequential Monte Carlo method, whose basic idea is very simple: the target distribution is represented by a weighted set of Monte Carlo samples. These samples are propagated and updated using a sequential version of importance sampling as new measurements become available. We summarize a basic PF algorithm as follows, while referring the reader to [7–9] for detail discussions on PF methods.

Algorithm 1: Basic Particle Filter Algorithm

❖ Initialization. Sample N equally weighted particles from the initial *pdf* of the target state, $p(\mathbf{x}_0)$:

- For $i = 1, \ldots, N$

$$\mathbf{x}_0^i \sim p(\mathbf{x}_0); \quad \omega_0^i = \frac{1}{N}$$

- Set $k \leftarrow 0$

❖ Iteration $k+1$

- Sampling new particles from proposal distribution $q(\cdot)$, i.e.,

For $i = 1, \ldots, N$ $\quad \mathbf{x}_{k+1}^i \sim q(\cdot)$

- Evaluate importance weights:

$$\tilde{\omega}_{k+1}^i = \frac{p\left(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1}^i\right) p\left(\mathbf{x}_{k+1}^i \mid \mathbf{x}_k^i\right)}{q\left(\mathbf{x}_{k+1}^i\right)}$$

- Normalize the importance weights such that

$$\omega_{k+1}^i \propto \tilde{\omega}_{k+1}^i, \text{ and } \sum_{i=1}^{N} \omega_{k+1}^i = 1$$

- Selection step: Multiply/Suppress particles with high/low importance weights respectively, resulting in a set of equally weighted particles, $\mathbf{x}_{k+1}^i, i = 1, \ldots, N$.

- Output:

$$E(\mathbf{x}_{k+1}) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_{k+1}^i\right)$$

$$Cov(\mathbf{x}_{k+1}) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_{k+1}^i - E(\mathbf{x}_{k+1})\right)\left(\mathbf{x}_{k+1}^i - E(\mathbf{x}_{k+1})\right)^T$$

The design of the proposal distribution, i.e., $q(\cdot)$, is of paramount importance for the PF algorithm. It has been shown that UKF can be used to produce good proposal distributions, particularly when the observation model is nonlinear [12]. The idea is that one treats a Guassian distribution outputted by the UKF as the PF's proposal distribution. It is shown that Rao-blackwellization technique can be used to improve the UKF's computational efficiency [11]. So here we adopt the Rao-blackwellised

UKF (RB-UKF) to generate the PF's proposal. An implementation of RB-UKF based on the models described in Section 2 is summarized as follows.

Algorithm 2: RB-UKF Algorithm

Assume we have got the estimate for the target state at time step *k*, $\mathbf{x}_k$, with its corresponding covariance, $\mathbf{P}_k$, the goal is to solve $\mathbf{x}_{k+1}$ and $\mathbf{P}_{k+1}$, as a new measurement $\mathbf{z}_{k+1}$ arrives.

❖ Linear State Prediction:

- $\mu_p = \mathbf{F}\mathbf{x}_k$
- $\mathbf{P}_p = \mathbf{Q} + \mathbf{F}\mathbf{P}_k\mathbf{F}^T$

❖ *Sigma* points sampling

- $\chi_0 = \mu_p, \quad \omega_0^c = \lambda/(n+\lambda) + (1-\alpha^2+\beta)$
- $\chi_i = \mu_p + \left(\sqrt{(n+\lambda)\mathbf{P}_p}\right)_i, \quad \omega_i^c = \frac{1}{2(n+\lambda)},$ for $i = 1,\ldots,n$
- $\chi_i = \mu_p - \left(\sqrt{(n+\lambda)\mathbf{P}_p}\right)_i, \quad \omega_i^c = \frac{1}{2(n+\lambda)},$ for $i = n+1,\ldots,2n$

where n is the dimension of the state vector, and α, β, and λ are parameters prescribed beforehand for the UKF.

❖ Nonlinear measurement update based on Unscented Transform

- $\mathbf{z}_{i,u} = h(\chi_i),\ i = 0,1,\ldots,2n$. ($h(\cdot)$ denotes the measurement function)
- $\hat{\mathbf{z}}_u = \sum_{i=0}^{2n} \omega_i^c \mathbf{z}_{i,u}$
- $\mathbf{P}_u = \sum_{i=0}^{2n} \omega_i^c \left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)\left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)^T$
- $\mathbf{P}_c = \sum_{i=0}^{2n} \omega_i^c \left(\chi_i - \mu_p\right)\left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)^T$
- $\mathbf{K} = \mathbf{P}_c\mathbf{P}_u^{-1}$
- $\mathbf{x}_{k+1} = \mu_p + \mathbf{K}\left(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_u\right)$
- $\mathbf{P}_{k+1} = \mathbf{P}_p - \mathbf{K}\mathbf{P}_u\mathbf{K}^T$

Next we use such RB-UKF algorithm to generate proposal distributions for the PF, and mix the sensor scheduling technique proposed in Section 3 into the PF framework, leading to the proposed PF algorithm.

Algorithm 3: The Proposed PF Algorithm for Smart Target Tracking

❖ Initialization.

- Sample N equally weighted particles from the initial *pdf* of the target state, $p(\mathbf{x}_0)$
- Assign specific values for the desired covariance matrix, $\mathbf{P}_d$, and δ
- Set $k \leftarrow 0$
- Sensor scheduling for the next time step: use the active sensor while keep the passive one idle

❖ Iteration $k+1$

- For $i = 1,\ldots,N$

➢ Perform RB-UKF algorithm to $\mathbf{x}_k^i$ to get

$$\left\{\tilde{\mathbf{x}}^i_{k+1},\ \mathbf{P}^i_{k+1}\right\}$$

➢ Sample a new particle from the proposal distribution: $\mathbf{x}_{k+1}^i \sim q(\cdot) = N\left(\tilde{\mathbf{x}}_{k+1}^i, \mathbf{P}_{k+1}^i\right)$

- Evaluate importance weights:

$$\tilde{\omega}^i_{k+1} = \frac{p\left(\mathbf{z}_{k+1} \mid \mathbf{x}^i_{k+1}\right) p\left(\mathbf{x}^i_{k+1} \mid \mathbf{x}^i_k\right)}{q\left(\mathbf{x}^i_{k+1}\right)}$$

Normalize the importance weights such that

$$\omega^i_{k+1} \propto \tilde{\omega}^i_{k+1}, \text{ and } \sum_{i=1}^{N} \omega^i_{k+1} = 1$$

- Selection step: Multiply/Suppress particles with high/low importance weights respectively, resulting in a set of equally weighted particles, $\mathbf{x}^i_{k+1}, i = 1,\ldots,N$.
- Output:

$$E\left(\mathbf{x}_{k+1}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}^i_{k+1}\right)$$

$$Cov\left(\mathbf{x}_{k+1}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}^i_{k+1} - E\left(\mathbf{x}_{k+1}\right)\right)\left(\mathbf{x}^i_{k+1} - E\left(\mathbf{x}_{k+1}\right)\right)^T$$

- Sensor scheduling for the next time step:

➢ If $\left\|\mathbf{P}_D - Cov\left(\mathbf{x}_{k+1}\right)\right\|_2 > \delta$

select the active sensor to work while keep the passive one idle;

➢ Else, select the passive sensor to work while keep the active one idle.

5. Performance Evaluation

In this section, we evaluate the performance of our proposed method in Section 4 by simulations. First, we compare the tracking performance of our method with those of two other trackers, one adopting the passive sensor for detection and the other utilizing the active sensor for detection, based on a set of Monte-Carlo (MC) simulations. The purpose of this comparison is to demonstrate the effect of the sensor scheduling technique in the aspect of concealing the observer. Next, we investigate the effects of the parameter δ on our method's performance. This parameter is used to measure the difference between the desired covariance and the current estimation covariance in the sensor scheduling stage of our method.

The scenario to be investigated is shown in Figure 1. The observer travels at a fixed speed of 10m/s and executes 2 maneuvers. The observation period lasts 40 seconds. The target motion, described by (1) in this simulation, is subjected to an amount of process noise with $q_s = 1$. The initial position and speed of the target are $(300m, 300m)$ and $(12.25m/s, -12.25m/s)$, respectively. The other parameters for simulation initialization are summarized in Table 1.

For performance comparison, we take the root-mean square (RMS) position error as the index:

$$\mathrm{RMS}_k = \sqrt{\frac{1}{M}\cdot\sum_{i=1}^{M}\left(\bar{x}_{t,k}^{i} - x_{t,k}^{i}\right)^2 + \left(\bar{y}_{t,k}^{i} - y_{t,k}^{i}\right)^2} \qquad (8)$$

where $(x_{t,k}^{i}, y_{t,k}^{i})$ and $(\bar{x}_{t,k}^{i}, \bar{y}_{t,k}^{i})$ denote the true and the estimated target positions at time step k at the ith MC run, and M is the total number of independent MC runs. Here $M = 50$ runs are done for the following three trackers, the proposed sensor scheduling based PF (SS-PF) tracker, the passive/active mode PF (PaPF/AcPF) tracker which only use the passive/active sensor in the filtering process. As shown in Figure 2, the performance of the proposed SS-PF tracker is comparable to that of the AcPF tracker, and it is much better than that of the PaPF tracker. For the SS-PF tracker, the average number of time epochs, when the active sensor is used during the whole tracking process, is only 13. It means that the SS-PF tracker gets a similar filtering performance as that of the tracker which uses the active sensor all the time, while concealing the observer to an extent by reducing the use of the active sensor. A specific estimation result of this SS-PF for the target's trajectory is shown in Figure 1; the associated sensor scheduling result is also illustrated in Figure 4. As can be seen, at first, the SS-PF tracker selects the active sensor to do detection to get a good enough tracking initialization, then it dynamically switch the uses of the passive and the active sensors online. The sensor switch uses of the passive and the active sensors online. The sen-

Table 1. Parameters used for initialization.

Symbol	Quantity	Value
T	Sampling period	1s
σ_b	Standard error of bearing noise	1°
σ_d	Standard error of range noise	5m
N	Particle Number	200
$\mathbf{P}_D$	desired covariance matrix	diag([5 0.25 0.2])
δ	predefined boundary for the norm of covariance	5

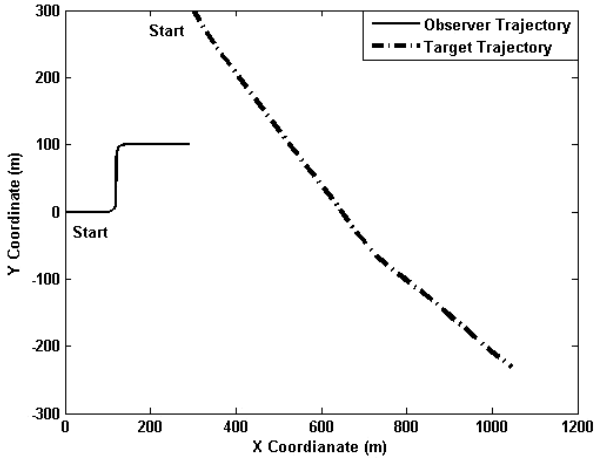


Figure 1. The observer's and the target's movement trajectories in this experiment.

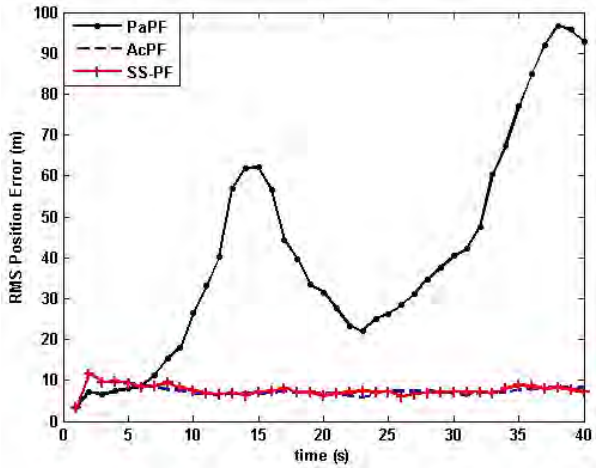


Figure 2. RMS position error versus time. PaPF and AcPF denotes passive mode and active mode PF tracker respectively, and SS-PF denotes the proposed tracker in this paper.

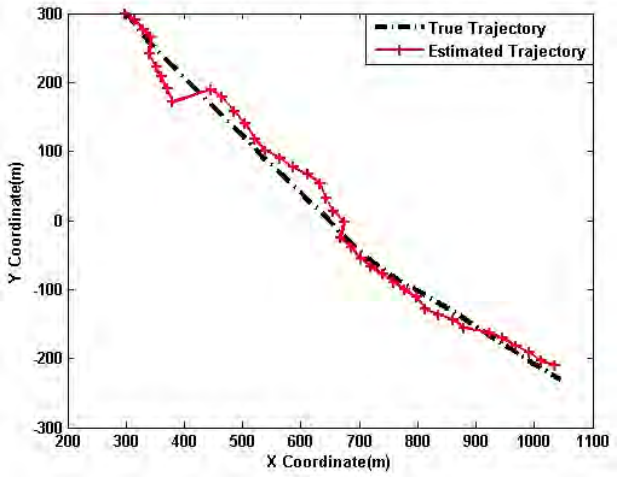


Figure 3. The true target trajectory against the estimated one by the proposed PF tracker.

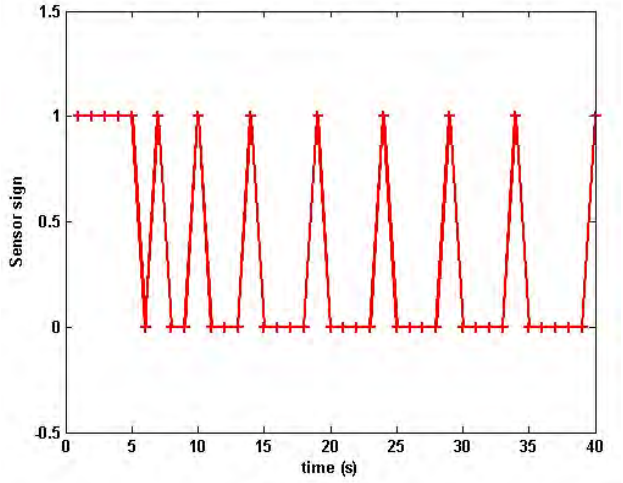


Figure 4. One instance of the sensor scheduling result: 0/1 denotes passive/active sensor being used.

Table 2. Performance evaluation with different δ values.

δ	The averaged number of time epochs when the active senor is used/ total number of time epochs	RTAMS (m)
5	13/40	8.08
10	11.5/40	9.06
50	7/40	9.11
100	6/40	19.66

sor switch process is actually a process of balancing the tracking filter performance with the concealment of the observer.

Next we evaluate the performance of the SS-PF tracker with respect to the value of δ. We use as index the root time averaged mean square (RTAMS) error defined as follows

$$\text{RTAMS} = \sqrt{\frac{1}{(t_{\max} - l)M} \sum_{k=l+1}^{t_{\max}} \sum_{i=1}^{M} \left(\bar{x}_{t,k}^{i} - x_{t,k}^{i} \right)^2 + \left(\bar{y}_{t,k}^{i} - y_{t,k}^{i} \right)^2}$$

where $t_{\max}$ is the total number of the time epochs for a single run. Here $t_{\max} = 40$. l is the time index after which the averaging is carried out. Here $l = 0$. For each case with a specific δ value, $M = 50$ independent MC runs are done. We summarize the result in Table 2.

It is shown that, the proposed SS-PF method actually balances the tracking filter performance with the concealment of the observer, and such balance is controlled by the parameter δ.

6. Conclusions

In this paper, we address the problem of tracking a smart target. This problem requires that the observer conceal itself well, for that once it is detected by the smart target, the latter may react in a manner that makes the future track more difficult. We analyze the relationship between the tracking filter performance and the observer's concealment. Based on such analysis, we propose a novel tracking method, in which a sensor scheduling technique, covariance control, is blended with an elaborately devised PF algorithm. Both theoretical analysis and simulation results demonstrate the efficiency of this method in dealing with the problem under consideration. It is shown that this method can balance the state filtering performance with the concealment of the observer well.

7. References

[1] C. Kreucher, D. Blatt, A. Hero, and K. Kastella, "Adaptive multi-modality sensor scheduling for detection and tracking of smart targets," Digital Signal Process, Vol. 16, pp. 546–567, 2006.

[2] C. Savage and B. L. Scala, "Sensor management for tracking smart targets," Digital Signal Process, doi:10.1016/j.dsp.2007.10.013 , 2007.

[3] J. C. Gittins and D. M. Roberts, "Search for an intelligent evader concealed in one of an arbitrary number of regions," Naval Research Logistics Quarterly, Vol. 26, No. 4, pp. 657–666, 1979.

[4] D. M. Roberts and J. C. Gittins, "Search for an intelligent evader: strategies for searcher and evader in the two-region problem," Naval Research Logistics Quarterly, Vol. 25, No.1, pp. 95–106, 1978.

[5] B. Liu, X. Ma, and C. Hou, "Smart target tracking using sensor scheduling and particle filter," in Proc. of Inter. Conf. on Signal Processing, Beijing, pp. 2620– 2623, 2008.

[6] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: Issues and approaches," Information Fusion, Vol. 3, No. 2, pp. 163–186, 2002.

[7] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, 2004.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonliear/non-gaussian bayesian tracking," IEEE Trans. on Signal Process, Vol. 50, No. 2, pp. 174–188, 2002.

[9] A. Doucet, N. De. Freitas, and N. Gordon, Sequential Monte Carlo in Practice, Springer Verlag, New York, 2001.

[10] M. Kalandros and L. Y. PAO, "Covariance control for multisensor systems," IEEE Trans. on Aerospace and Electronics Systems, Vol. 38, No. 4, pp. 1138–1157, 2002.

[11] M. Briers, S. Maskell, and R. Wright, "A rao-blackwellised unscented kalman filter," in Proc. of the 6th Int. Conf of Info. Fusion, Vol. 1, pp. 55–61, 2003.

[12] R. der Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," Tech. Rep, Department of engineering, University of Cambridge, CB21PZ Cambridge, 2000.

Wireless Sensor Network, **2009, 1, 306-315**
doi:10.4236/wsn.2009.14038 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Scientific Research

Tree Based Aggregation Algorithm Design Issues in Wireless Sensor Networks

Periyathambi EZHUMALAI[1], S. MANOJ KUMAR[1], Chokkalingam ARUN[2], D. SRIDAHARAN[3]
[1]*Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Tamilnadu, India*
[2]*Department of Information Technology, Sri Venkateswara College of Engineering, Tamilnadu, India*
[3]*Department of Electronics and Communication, Anna University, Tamilnadu, India.*
Email: {*ezhumalaip, manoj, carun*} *@svce.ac.in*
Received June 18, 2009; *revised July* 23, 2009; accepted *August 11*, 2009

Abstract

Wireless Sensor networks (WSN) consists of tiny sensor nodes which are having limited CPU, memory, battery and communication capabilities. WSN differs from conventional wireless networks in several ways such as sensor nodes have severe energy constraints produces redundant low-rate data traffic, and many-to-one flows. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these constraints. Hence, it is essential to have data-centric technologies that perform in-network aggregation which gives energy-efficient dissemination. We focus on data aggregation problems in energy constrained sensor networks. The main goal of data aggregation algorithms is to gather and aggregate data in an energy efficient manner so that the network lifetime can be increased. In this paper we present an elaborate survey on different data aggregation algorithms based tree architecture and compare them in terms of lifetime, latency and data accuracy. Also we present the different network issues such as reliability and security while performing aggregation.

Keywords: Wireless Sensor Networks, Data Aggregation, Tree Based Algorithms

1. Introduction

Wireless sensor networks (WSNs) consist of a large numbers of inexpensive sensor nodes which are deployed densely in harsh and inaccessible terrains like gigantic mountains, valleys, oceans, deep forest etc [1]. These nodes have capabilities for sensing the parameter like temperature, humidity, vibration, etc, processing the sensed data and transmitting it to the base station either by direct link or by multihop communication. Though these nodes have no fixed topology, they can form a multihop self-organized network by sending beacons through wireless channels and configure themselves into adhoc wireless network. Sensor networks can be used in plenty of applications like battle field and enemy terrain surveillance and reconnaissance in military, patient health monitoring in medical application, environment and habitat monitoring application, building infrastructure monitoring, building a smart and automated home and office, search and rescue operations during emergency conditions like earthquake, flood, tsunami, etc [1,2]. In this paper, the various issues while aggregating the information, the various techniques for aggregation and their strength and weakness in terms of energy, data freshness and data accuracy are discussed briefly.

Sensor nodes are battery driven and hence operate on an extremely frugal energy budget. It is impracticable to replace the battery for the network with thousands of physically embedded nodes. The network lifetime can be maximized by incorporating energy-awareness into every stage of wireless sensor network design and operation [3]. In a mote, the battery power is utilized by the computing sub system (MCU), sensing subsystems, and communication sub system. The microcontroller (MCU) is responsible for control of the sensors, execution of communication protocols, and signal processing on the gathered sensor data. The sensor node radio is responsible for wireless communication with neighboring nodes and the outside world. Sensor transducers translate the physical phenomena to electrical signals. There are several sources of power consumptions in the sensor including 1. Sampling 2. Signal conditioning and 3. Analog to digital

conversion. All these sub systems consume power. From the data sheet of many commercially available motes, it is understood that the power used by the microcontroller and the sensing sub systems is less compared to that used by communication unit. In order to increase the life time of the network, it is good to design an algorithm that reduces the number of transmissions. The data aggregation algorithm can reduce the number of transmission by allowing the aggregator node to transmit only the required data, not the redundant information.

This paper is organized as follows: section 2 discuss about the data aggregation, aggregation operators and the method of aggregation. Section 3 describes the parameters considered for analyzing the performance of the aggregation algorithms and section 4 discuss about network architecture, various tree based protocols in detail. Finally section 5 describes the network issues like reliability and security during aggregation.

2. Data Gathering and Data Aggregation

Data gathering is defined as the systematic collection of sensed data at predefined time interval from the multiple sensor nodes and transmitted to the base station for further processing. Since sensor nodes are energy constrained, it is inefficient for all the sensors to transmit the data directly to the base station due to the following reasons. Data generated from neighboring sensors is often redundant and highly correlated. Hence these sensors report the same data about an event and hence it is not needed to transmit the multiple copies of the same information. In addition, the amount of data generated in large sensor networks is usually enormous for the base station to process. Hence, we need methods for combining data into high quality information at the sensors or intermediate nodes which can reduce the number of packets transmitted to the base station resulting in conservation of energy and bandwidth. This can be accomplished by data aggregation. Data aggregation is defined as the process of aggregating the data from multiple sensors to eliminate redundant transmission and provide fused information to the base station. Data aggregation usually involves the fusion of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station (sink).

2.1. Data Aggregation Methods

In energy constrained wireless sensor nodes, the Energy efficiency can be achieved by using some of the in-network aggregation techniques. While forwarding the data, the information from various sources are combined along its path to the sink and this process is called In-network aggregation. The data aggregation operator is simple SUM, AVERAGE, MAXIMUM, MINIMUM and COUNT, etc to more complicated data aggregation methods like, MEDIAN, Wavelet Histogram [1]. The energy saving depends on the type of aggregation operator employed. For example, if the MAX operator is used for aggregation and the results in a single packet of same size as that of individual sensor readings. If the aggregation ratio is n:1, then the energy saving will be n-fold. Suppose, the concatenate operator is used for aggregation, i.e, the individual sensor readings are appended by the aggregator node, the energy saving takes place only on medium access. The processing of data takes place in side the network, hence the aggregation process suppress the transmission of unwanted information. This increases the lifetime of the sensor nodes and hence the sensor network. The benefits of in-network aggregation is also extended to the sink by receiving less amount of useful information from the sensor sources and hence the sink can perform less processing and filtering to get useful information from these data by consuming less resources. Even though, aggregation reduces the energy consumption, it increases the delay of delivering the packet to the sink. This is due to the fact that each aggregator has to wait for a predefined time interval to collect data from its children. This leads to the delay in the delivery of the data and hence the sink may not get the fresh data. Hence there is a tradeoff between energy saving, data accuracy and freshness, i.e., the longer a node waits, the more readings it is likely to receive and therefore, the more accurate the information it sends out. On the other hand, waiting too long may result in stale data. Furthermore, if a node waits too long, it may interfere with the next "data wave".

By considering the above factors, there are three different methods of data aggregations namely, Periodic simple, periodic per-hop, and periodic per-hop adjusted are proposed. Periodic simple aggregation works by having each node wait a pre-defined period of time, aggregate all data items received, and send out a single packet containing the result. In Periodic per-hop, the aggregator nodes send out the fused packet as soon as they received packets from all their children or till the clock times out, the time out is set to data generation rate. Periodic per hop adjusted uses the same basic principle of periodic per-hop but schedules a node's timeout based on its position in the distribution tree [4].

Ignacio Solis and Katia Obraczka proposed the cascading timeouts aggregation mechanism which falls on periodic per-hop adjusted category, in which the nodes timeout depends on the single hop delay. The performance of this is compared with the other methods in terms of energy, data accuracy and data freshness. They showed that the performance of the algorithm depends on the time out value of the aggregator. When compared to other existing periodic per-hop adjusted algorithms, cascading time out reduces the traffic by 6 times while maintaining the data accuracy and freshness. Also it pre

sents other benefits such as not requiring clock synchronization among nodes and minimizing the time-out-scheduling overhead. It is a simple aggregation algorithm, minimum control overhead, no clock synchronization and independent of routing algorithms.

3. Performance Measures

Network lifetime, data accuracy, and latency are some of the important performance measures of data aggregation algorithms. The definitions of these measures are highly dependent on the desired application.

Network lifetime: Network lifetime is defined as the number of data aggregation rounds till the specified percentages of the total nodes dies and the percentage depends on the application. In some applications, simultaneous working of the all the sensor nodes is crucial hence the lifetime of the network is the number of rounds until the first node dies. The data aggregation methods ensure the uniform draining out power from all the sensor nodes which improves the energy efficiency of the nodes and enhances the lifetime of the entire network.

Latency: Latency is defined as the delay involved in data transmission, routing and data aggregation. It can be measured as the time delay between the data packets received at the sink and the data generated at the source node. It is also called as data freshness.

Data accuracy: The definition of data accuracy depends on the specific application for which the sensor network is designed. For instance, in a target localization problem, the estimate of target location at the sink determines the data accuracy. In general it is a measure of ratio of total number of readings received at the sink to the total number of readings generated.

Communication Overhead: It measures the communication complexity of the in-network aggregation algorithms. The control packets are transmitted between nodes to maintain the network. This packet will not relay any data to the sink and hence these are considered as overhead. The amount of overhead should be kept minimum, since these packets drains energy from the battery.

The performance of the data aggregation algorithms should be analyzed based on the above metrics. These metrics are interdependent on each other. Improving the energy efficiency is achieved by fusing more packets. This gives more accurate data but increases the delay.

Ramesh Rajagopalan *et al* discussed in their survey paper about the various data aggregation algorithms and compared their performance in detail. According to them, the existing data aggregation algorithms are classified under the following categories: Network Architecture based, Network flow based, and QoS based [5]. In this paper we concentrate more on the aggregation schemes based on the architecture and we give brief research work going on tree based network.

4. Data Aggregation Based on the Network Architecture

The architecture of the network will have a great impact on the performance of the data aggregation. The network may have either a flat or hierarchical architecture.

4.1. Flat Networks

In the flat networks, all the nodes are having similar capabilities and responsibilities and play the similar role. They are not having any fixed topology; the route to the sink from a data source is established by the routing protocol. Classic routing protocols which are based on the shortest path are not suitable for data aggregation paradigm. To promote data aggregation, the packet should be routed based on its content and the next hop candidate should be selected based on the most suitable aggregation points, data types, the priority of information, etc [6]. This type of routing is classified as data centric routing and it is interlinked with the data aggregation mechanism. The Sensor Protocol for Information via Negotiation (SPIN), DIRECTED DIFFUSUION and PEGASIS are the few examples in this group and are discussed in [5].

4.2. Hierarchical Networks

Flat network architecture will not be suitable if the size of the network is large. It is either Cluster based or Tree based. Hierarchical networks contain heterogeneous nodes, which can be either an end device/child or a cluster head/parent. In cluster based networks, the cluster head is selected based on the residual energy in it, the node with more energy is selected as cluster head. The end nodes are not transmitting the data directly to the sink; instead it transmits it to the cluster head which will be behaving as aggregator. The cluster heads will aggregate the data coming from its children and forward it to the sink as shown in the Figure 1. The brief description about the following algorithms is given in [5]. Low Energy Adaptive Clustering Hierarchy (LEACH), Hybrid Energy Efficient Distributed Clustering Approach (HEED) and clustered diffusion with dynamic data aggregation (CLUDDA) fall in the category.

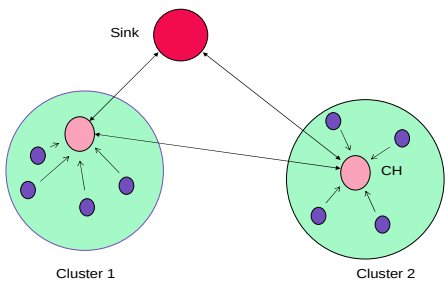


Figure 1. Cluster routing.

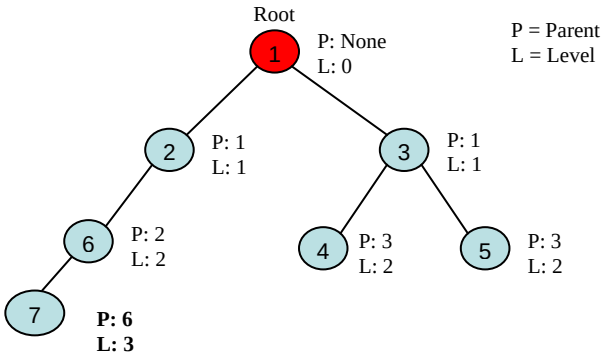


Figure 2. Data aggregation tree.

In the tree based networks, the sensor nodes are organized into a tree and the data aggregation takes place at the intermediate parent nodes along the tree as shown in Figure 2.

In this Figure 2, the nodes 4, 5, and 7 are the leave nodes which send the raw data to its parent and the nodes 6, 2 and 3 performs the role of parent which are responsible for data aggregation from its children and forwarding the aggregated information to the root. The main challenge in this type of data aggregation is the construction of efficient tree. In a network graph G =(V, E) where V is the set of nodes and E is the set of edges that connect nodes which can communicate directly. Let S1, S2..., Sk $\in$ S be data sources and D be a sink node. For optimal aggregation, a minimal cost tree connecting nodes in S and node D with minimal number of edges should be found. This is the Steiner Tree problem, which is an NP-hard variant of the minimum spanning tree problem, some suboptimal aggregations are proposed in [7]. In Center at Nearest Source (CNS), the source which is nearest the sink acts as the aggregator, all other sources send their data directly to this source which then sends the aggregated information to the sink. The Shortest Paths Tree (SPT) allows each source to send its information to the sink along the shortest path and the overlapping paths are combined to perform aggregation. The Greedy Incremental Tree (GIT) builds the aggregation tree by allowing the source which is nearest to the sink to send its data via shortest path. Then the next source nearest to the tree is allowed to join the tree and the entire tree is constructed. In the following paragraphs, we discuss about the various tree-based protocols available.

4.2.1. TAG: A Tiny Aggregation Service for Adhoc Sensor Networks

It is the very first algorithm proposed by Madden et al in 2004 and is more efficient in terms of energy. The tree is constructed by the root node which sends the broadcast message with level 0 and its sensor Id. All nodes hearing the message increase the level field, attach their id and rebroadcast it again. They select the source of the message as their parent. The process continues down the tree. It can be used as periodic monitoring or query driven.

In periodic monitoring mode, the root sends the query. The child nodes send their current aggregated value and rebroadcast the query to the next level. Now the root receives the information from the first level child nodes. This process continues until, the root receives the packets from the last level. The TAG offers a lot of advantages-saves energy, minimizes the number of messages transfer, use of epoch allows nodes to sleep during idle time thereby saving energy [8].

4.2.2. TiNA: Temporal Coherency Aware In-Network Aggregation

The approach is to send the data only when there is a significant change in the data value in the adjacent readings over time. The concept of epoch as in TAG is also used here for synchronizing the receipts of the packets from the child nodes and sending the aggregate. A data value can be ignored if the variation from the previous value is within the specified range called filters. This needs higher memory requirements at each node because they need to store the intermediate results of the child nodes, (partial aggregation). The advantage is the significant reduction in number of messages over TAG [9].

4.2.3. DQEB-Dynamic Query-Tree Energy Balancing Protocol

The DQEB [10] protocol is an energy balanced protocol by dynamically modifying the tree structure based on the energy left at nodes. In this approach the nodes are organized into clusters with cluster heads. Each node is assigned a weight which goes on increasing with decreasing lifetime or energy. As the energy decreases, it is wiser to move the node down the tree i.e. turn it in to a leaf node so that the tree does not get disconnected. The energy cost depends on the number of leaf and non leaf nodes and the energy remaining at the node. Whenever the weight of a non leaf node goes down a threshold, the coordinator node asks all its child for alternating parents, then using a greedy approach it selects the alternating parents for all its children and itself becomes a leaf node. Since the nodes with less energy has become the leaf node it will live a little longer as now it only has to send its data. This increases the life time.

4.2.4. Adaptive Application-Independent Data Aggregation in WSN (AIDA)

As the name suggested, the aggregation decision is independent of the application and it also ensures the timely delivery of the packets. It resides between the Routing and MAC layers of the network stack as in Figure 3 and hence it doesn't require any modification in the existing network and medium access protocols [11].

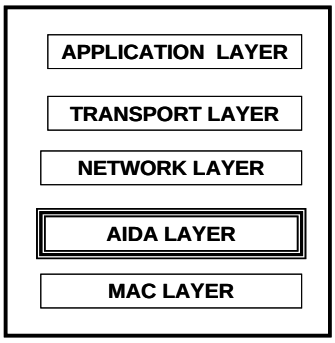


Figure 3. Architecture of AIDA.

It adaptively adjusts its aggregation strategies according to the traffic conditions and the sensor network requirements. There are four aggregation strategies supported in this framework. It buffers the packets from the network layer, and the network units are aggregated by using any of the following methods and schedules this aggregated packets to MAC layer for transmission. No Aggregation, where packets are not aggregated, Fixed Aggregated in which fixed numbers of packets are aggregated, On-demand scheme, in which the aggregation takes place until the channel is available for transmission and Dynamic Feed back loop combines the fixed scheme and the on-demand scheme.

Tian He *et al* simulated this frame work using GloMoSim simulator and verified the results with the testbed with Berekely MICA motes. The end-to-end delay, energy consumption, MAC control packet overhead, Degree of Aggregation and AIDA overhead per packet delivered are considered as the performance metrics. This shows that the end-to-end delay of dynamic aggregation scheme is 80% less than that of non aggregation scheme under heavy traffic load condition. Also, the energy consumption is reduced by 50% with reduced overheads.

4.2.5. Load Balanced Tree Protocol (LBTP)

The main purpose of LBTP is to gather periodical data from all sensors to a sink. In LBTP, the non leaf nodes have similar amount of children and the tree structure changes when the energy of the non leaf node is lower than the predefined threshold. [12] It first forms the Breath First search (BFS) tree and the sink adjusts this tree to load balancing tree.

Each node broadcasts the HELLO message periodically. The HELLO message contains the packet type, broadcasting node id, remaining energy, parent id, and set of possible parent candidates. The node which receives the HELLO message will update its neighbor table. To balance the tree, after receiving the TREQ packet, each node has to wait for the predefined time to receive any HELLO message. After updating the NT by HELLO message, it will send the Tree Reply (TREP) packet to its parent, so that the tree can be adjusted properly.

The performance of the algorithm in terms of the amount of data received vs. number of sensors and the lifetime of the sensors vs. number of sensors is compared with that of DQEB, BFT and SLBTP. They claim that LBTP can receive more data when the density of the network is very high since the lifetime of the nodes is extended.

4.2.6. Heuristic Algorithms for Real-Time Data Aggregation

In real time applications, the packet should be delivered to the sink in prescribed time bound. The packet arriving after the specified time limit is worthless. There are some applications like real time control system applications, which require this time limitation very firmly. This stringent requirement of time is called hard real time requirement. Some applications like live video streaming doesn't require this type rigid time requirements and are called soft real time. In soft real time applications, some packets arrive after the time bound will not collapse the entire system but the quality will be poor. Many researchers are trying to realize the soft real time data aggregation. In order to conserve the energy, the WSN nodes go to sleep state very often. By having such sleeping nodes, it is very difficult to achieve the hard real time bounds. The time constraints of a WSN application can be satisfied by constructing the tree such a way that the number of hops traveled by a packets should be minimum (HOP constraint H) and also each aggregator should have certain number of children (DEGREE constraint D). The end-to-end delay can be decreased by having less HOP count, which reduces the number of hops traveled by a packet and less DEGREE count, which decreases the waiting time of the aggregator to receive packets from its children. To have less number of hops, more children are added to each aggregator, which increases the degree count. Hence there is a tradeoff between H and D.

Hongju Cheng Q *et al*. suggested the three different tree construction algorithms for real time data gathering in which the packet should be transmitted with in specified time bound. They proposed three heuristics algorithms to build a MST with hop and degree constraints Node-First Heuristic (NFH), Tree-First Heuristic (TFH), and Hop-Bounded Heuristic (HBH) [13]. Simulation results reveal that they are all suitable to solve the real-time data aggregation problem and the performance of these algorithms is tested in terms of total energy cost against the increase in transmission range. It shows that, NFH is the performing better compare to other two algorithms since the energy cost is close to that of non constraint MST.

4.2.7. Correlation Aware Data Aggregation Tree

Due to the resource constraint nature of the sensor nodes, they are deployed densely in the field, wherein the phe-

nomena of interest to be monitored. The readings reported by these sensors are highly correlated and termed as spatial correlation. The data aggregation trees discussed above is not exploiting the correlation between the sensor nodes. In the following paragraphs, the data aggregation tree, Semantic/Spatial correlation-aware tree (SCT) that exploits the correlation strategies can be elaborated

In SCT, the entire network is divided into concentric rings and these rings are further divided into sectors. A node from each sector can have an aggregator depends on the residual energy and its location. The nodes, which are in a sector are closely packed and will report the data which are highly correlated. Hence this structure exploits the spatial correlation [14].

The given network with n no of nodes which are spread over the radius R is divided into m concentric rings and each ring is divided into sector that contains n_0 nodes. During the initial setup period, the sink broadcasts the packet that contains the information about the sink location, number of nodes in the network, radius of the network, number of concentric rings m, and the number of nodes in each sector n_0. By knowing the sink location**,** each node in the network will calculate the ring number i in which it resides and also calculates the number of sectors in that ring i as shown in the Figure 4.

Once the network is divided into circles and sectors, the node which acts as an aggregator for each sector must be selected. The node which is close to the geometric centre of the lower arc of the sector is selected as aggregator node for that sector. When a node in a sector has a data to send, it calculates the geometric centre and send the data to the node nearest to this centre. If a node receives the first packet, it takes the roll of data aggregator and broadcasts the message to its entire neighbor, declaring that it is the aggregator for that sector. Hearing this message all other nodes will route their packets to this node.

The role of the aggregator is shifted after few rounds of query propagation to achieve load balancing. Also, the location of the ring is shifted slightly and the sector is also shifted by some offset value. SCT ensures the reliable delivery of packets both during query as well as data delivery phase. During the query transmission phase, a node which is closer to the ideal location broadcasts itself as aggregator if the current aggregator fails. If the aggregator fails during the data collection phase, it is notified to the source node and hence the source node will retransmits the packet. This retransmission will initiate the selection of new aggregator. The performance of the tree is compared to that of correlation unaware structures and they show that the SCT perform better in terms of message cost and latency. The advantages of SCT are it does not require any centralized coordination, it requires very low overhead to maintain the tree and the tree can be constructed instantaneously.

5. Networking Issues

Due to the adhoc nature of the wireless networks and the dynamic channel conditions, the wireless channels are unreliable and prone to error. While doing aggregation, the packets contain more information and hence the packet loss is not acceptable. Hence the packet delivery should be reliable over unreliable wireless medium. This raises the reliability issues in WSN. Also the packets should be transmitted to the sink in secured manner. The hackers should not modify the content or they should not send the wrong information to the sink which will mislead the sink. Hence the security issues are also very important while aggregating the information. In this section we will discuss the reliability and security issues in details.

5.1. Reliability

The loss of aggregated packets in WSN causes more energy loss since lot of resources is already invested to transmit the sensor readings from various sensors and retransmission of the lost packets requires more energy. This leads to more energy wastage in the sensor nodes

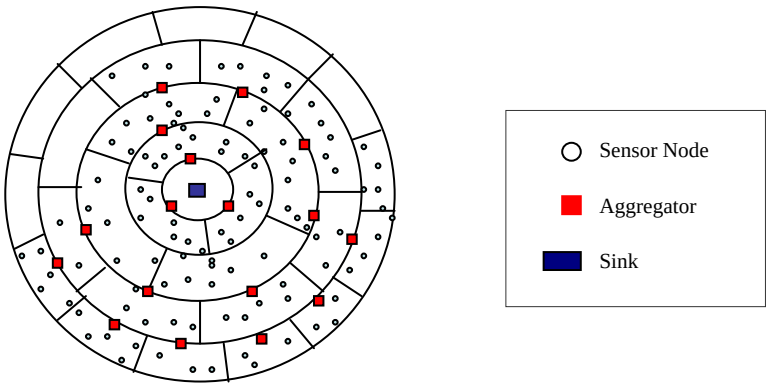


Figure 4. SCT structure.

which is undesirable. Holger Karl, Marc Lobbers investigated the strategies to adaptively employ different link-level error control mechanisms (FES and ARQ) depending on how precise the information in the message packet [15]. Lost messages in the child-parent link will not create much loss but the lost of aggregated message leads to lot of information loss as well as lot of energy has been invested would have been lost.

5.2. Security

The two fundamental issues to be considered in wireless communication are data confidentiality and data Integrity. Data Confidentiality deals with the protection against the eavesdropping of the transmitted data by the intruders. Sensor nodes are deployed in the unattended area and are easy for the intruders to get or change the information physically and it can be made as malicious node. Also the communication medium is wireless, the information can be taped or modified easily. It necessitates the use of encryption methods to transmit the data. The data integrity ensures the consistency of the received data. It is easy for the intruder to spread the erroneous information or modify the transmitted information in the network. A proper Message Authentication Algorithm (MAC) schemes can verify the integrity of the message.

These security issues are more critical while aggregating the data. The aggregated packets are having more information and hence this should not be hacked or modified by the intruder. The source node or the aggregator node may become malicious node. A compromised node can modify, forge or discard messages. If the source node is compromised, it may send the wrong reading to the aggregator which results in corrupted aggregation at the aggregator. If the aggregator is compromised, it can either send the wrong aggregated result to the sink or it can use the wrong aggregator operator. Both of them made the sink difficult to estimate the original readings from the altered aggregated readings.

In summary an adversary can damage the data confidentiality by the following attacks: 1) eavesdropping the messages in the wireless channel; 2) compromising a node and obtaining all keys stored in it; 3) using the compromised node's keys to deduce the keys employed elsewhere in the network; 4) using the compromised node's keys to inject unauthorized malicious sensor nodes in the network. The adversary can also spoil the data integrity by the following attacks: 1) injecting arbitrary chosen malicious data into the compromised sensing nodes 2) modifying, forging, or discarding messages in the compromised aggregator nodes [16].

The secure aggregation should be designed to address these issues as well it should take care of the sensor nodes constraints. The size of the encrypted message should be less, the execution time and memory foot print of the security algorithm should be minimal. Also, the secrete key distribution algorithms should be secure and efficient.

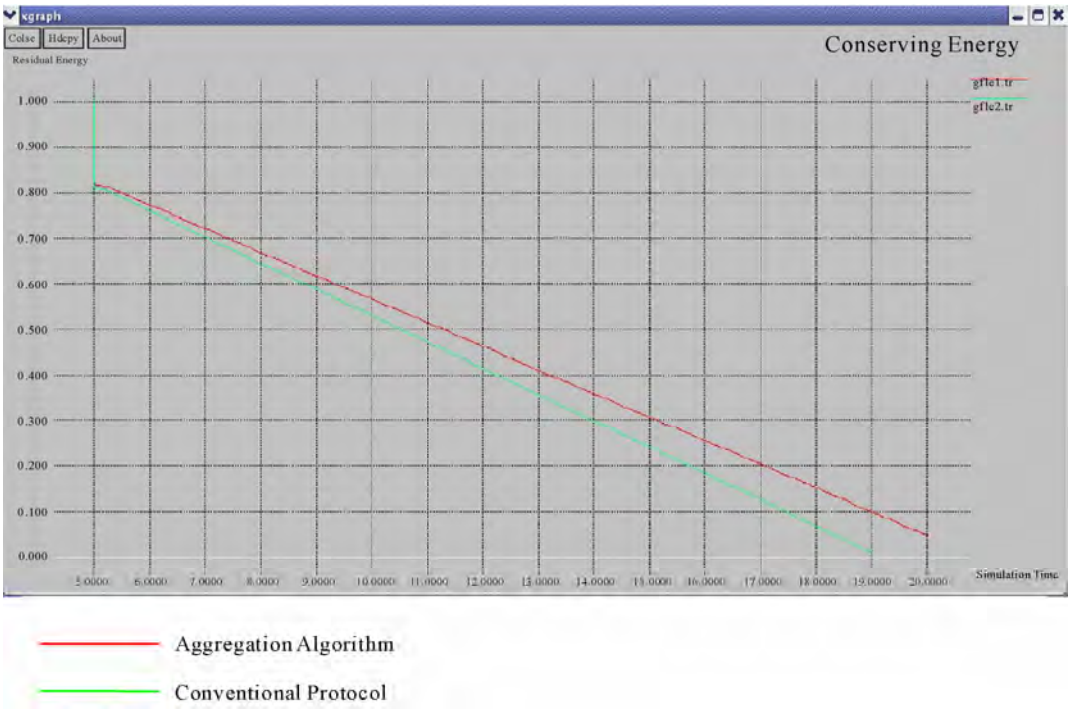


Figure 5. Residual energy of source as a function of time.

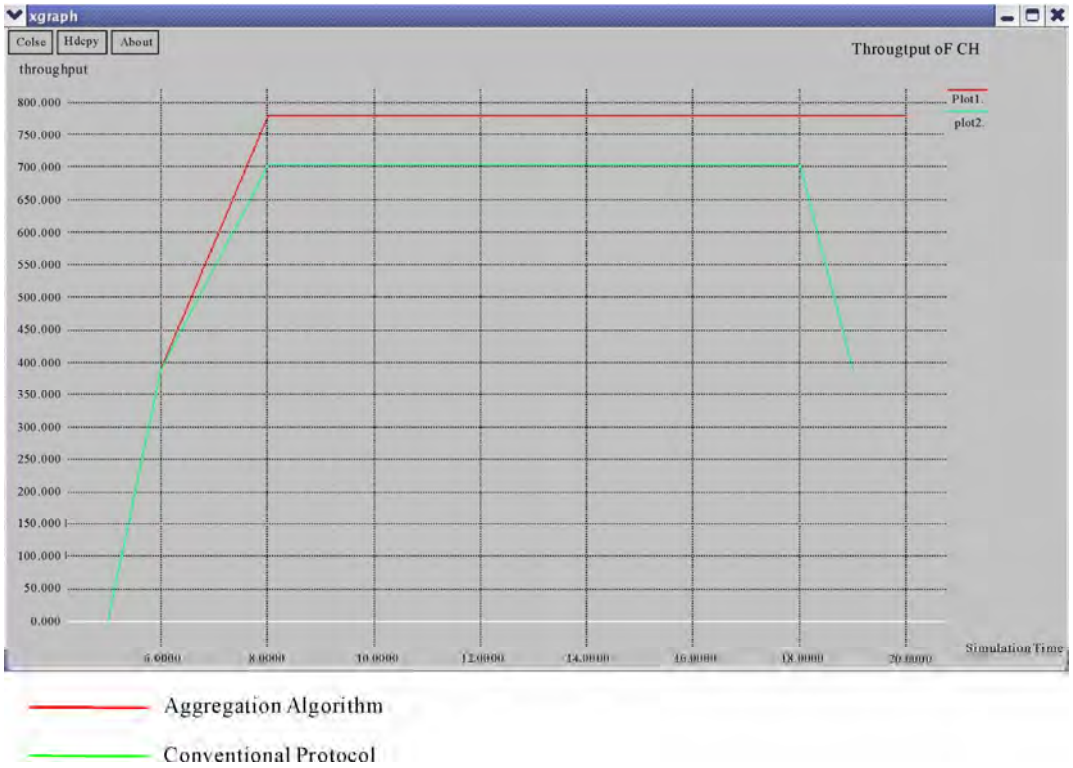


Figure 6. Throughput as function of time.

6. Simulation Results

6.1. Conserving Energy

We determine residual energy of the source node, which is defined as the remaining energy of a node and considered that as the metric to prove energy efficiency of our proposed protocol. We used this metric to show the impact of transmission power on energy reduction. Figure 5 shows the significant reduction in energy consumption by using aggregation algorithm when compared with conventional protocol. This shows the benefit of sending data in a multi-hop fashion towards cluster-head.

6.2. Throughput

We have also measured the throughput of the receiving node i.e. cluster-head node 10 in our scenario for both the cases. Throughput of a node is defined as the average rate of successful message delivery over a communication channel. Figure 6 show that aggregation algorithm achieves high throughput in comparison with conventional protocol.

6.3. Network Density

By considering the changes in the network density, we also study the relationship between the network lifetime and network density. In our experiment we have considered the change in the residual energy of source node i.e. node 0 in the end of simulation. The density of network is calculated via Equation [9]:

$$\lambda = N\pi R^2/A^2$$

where, N is sensor number, R is sensor range, A is sensor area.

By keeping network area constant and increasing the number of nodes, we have increased network density. Due to increase in the network density, the hop count between source node and sink node also increases. When hop count increases node now transmit data to nearer node with less transmit power and hence consume less energy. Figure 7 shows the increase in the residual energy when we increase the hop count. We have taken N as 11, 21, 31, 41, and 51.

6.4. Packet Delivery Ratio

Besides examining the network lifetime extension roughly via energy saving, we also evaluate the network efficiency influenced by aggregation algorithm. Here, we measure the efficiency in term of data delivery ratio, which is defined as the number of received packets divided by the number of sent packets for a certain time period. From our simulation results illustrated in Figure 8, we find that this ratio does not change much while the network is alive. It shows the stable performance of our

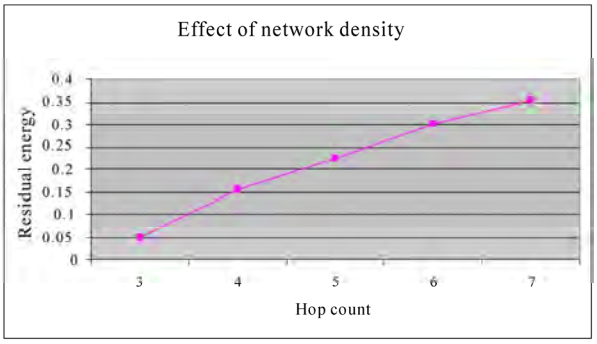


Figure 7. Effect of network density.

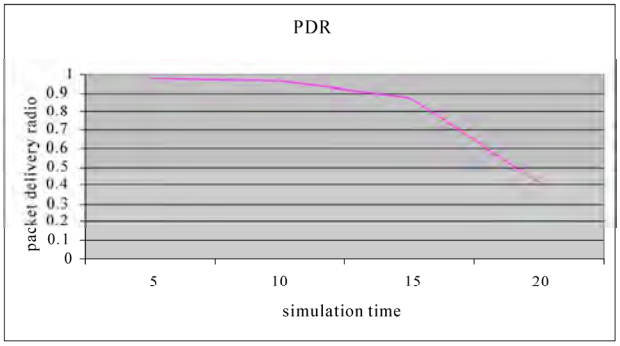


Figure 8. Packet delivery ratio of the network.

protocol. When the network energy is running out, the data delivery ratio collapses rapidly. This phenomenon probably can be taken as a sign of the network death.

7. Conclusion

Wireless sensor networks are energy constrained network. Since most of the energy consumed for transmitting and receiving data, the process of data aggregation becomes an important issue and optimization is needed. Efficient data aggregation protocols not only provide energy conservation but also remove redundancy in the data and hence provide useful data only. There exist several protocols for data aggregation which uses different approaches to provide energy efficiency. In Tree-based approaches, nodes send their data directly to Tree-head and Tree-head then aggregate and forward the data towards sink. We exploited this approach and proposed a new protocol Network Data Aggregation Algorithm.

Normally, the tree based structure will take time and energy for their construction and their maintenance. Also, if a non leaf node breaks, the particular sub section of the tree is disconnected from the network. It necessitates the implementation of reliability methods to ensure the delivery of the packets. Due to the dense deployment nature of the sensor networks, it is necessary to take care of the spatial and temporal correlations into account while designing data aggregation algorithms. Also, appropriate security algorithm should be used to ensure the confidentiality and integrity of the message. In addition to all the above limitations, it uses a simple routing method reduces the complexity of the routing algorithms.

The simulation result shows that when the data from source node is send to Tree-head through neighbors nodes in a multi-hop fashion by reducing transmission and receiving power, the energy consumption is low as compared to that of sending data directly to Tree-head.

8. References

[1] I. Akyldiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, “A survey on sensor Networks,” IEEE Communication Magazine, Vol. 40, No. 8, pp. 103–114, August 2002.

[2] Y. Yang, V. K. Prasanna, and B. Krishnamachari, “Information processing and routing in wireless sensor networks,” World Scientific, 2006.

[3] V Raghnathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy aware wireless sensor networks,” IEEE Signal Processing Magazine, Vol. 19, No. 2, pp. 40–50, March 2002.

[4] I. Solis and K. Obraczka, “In-network aggregation tradeoffs for data collection in wireless sensor networks,” International Journal of Sensor Networks, Vol. 1, No. 3/4, pp. 200–212, 2006.

[5] R. Rajagopalan and P. K. Varshney, “Data aggregation techniques in sensor networks: A survey,” Journal on IEEE Communications, Surveys and Tutorials, Vol. 8, No. 4, 2006.

[6] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: A survey,” IEEE Wireless communications, Vol. 14, No. 2, pp. 70–87, 2007.

[7] B. Krishnamachari, D. Estrin, and S. Wicker, “Modeling data-centric routing in wireless sensor networks,” Computer Engineering Technical Report CENG 02–14, 2004.

[8] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: A tiny aggregation service for adhoc sensor networks,” OSDI December 2002.

[9] A. Sharaf, J. Beaver, A. Labrinidis, and K. Chrysanthis, “Balancing energy efficiency and quality of aggregate data in sensor networks,” VLDB Journal, Vol. 13, No. 4, pp. 384–403, December 2004.

[10] H. Yang, F. Ye, and B. Sikdar, “A dynamic query-tree energy balancing protocol for sensor networks,” Wireless Communications and Networking Conference, WCNC, Vol. 3, pp. 1715–1720, 2004.

[11] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “AIDA: Adaptive application-independent data aggregation in wireless sensor networks,” ACM Transactions on Embedded Computing Systems, Vol. 3, No. 2, pp. 426–457, May 2004.

[12] T. S. Chen, H. W. Tsai, and C. P. Chu, “Gathering-load-balanced tree protocol for wireless sensor networks,” IEEE conference on sensor Networks, Ubiquitous, and Trust-worthy computing, 2006.

[13] H. J. Cheng, Q. Liu, and X. H. Jia, "Heuristic algorithms for real-time data aggregation in wireless sensor networks," International conference on Wireless and Mobile Computing, 2006.

[14] Y. J. Zhu, R. Vedantham, S. J. Park, and S. Raghupathy, "A scalable correlation aware aggregation strategy for wireless sensor networks," Science Direct-Information fusion, Vol. 9, No. 3, pp. 364– 369, 2008.

[15] H. Karl, M. Lobbers, and T. Nieberg, "A data aggregation framework for wireless sensor networks," TKN technical Report Series.

[16] Y. P. Sang, H. Shen Y. Inoguchi, Y. S. Tan, N. X. Xiong, "Secure data aggregation in wireless sensor networks: A survey," Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006.

Wireless Sensor Network, **2009, 1, 316-323**
doi:10.4236/wsn.2009.14039 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Scientific Research

AEESPAN: Automata Based Energy Efficient Spanning Tree for Data Aggregation in Wireless Sensor Networks

Zahra ESKANDARI, Mohammad Hossien YAGHMAEE
Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
Email: *za_es73@stu-mail.um.ac.ir, hyaghmae@ferdowsi.um.ac.ir*
Received May 6, 2009; *revised July* 5, 2009; *accepted July* 10, 2009

Abstract

In Wireless Sensor Networks (WSNs), sensor nodes are developed densely. They have limit processing capability and low power resources. Thus, energy is one of most important constraints in these networks. In some applications of sensor networks, sensor nodes sense data from the environment periodically and transmit these data to sink node. In order to decrease energy consumption and so, increase network's lifetime, volume of transmitted data should be decreased. A solution, which is suggested, is aggregation. In aggregation mechanisms, the nodes aggregate received data and send aggregated result instead of raw data to sink, so, the volume of the transmitted data is decreased. Aggregation algorithms should construct aggregation tree and transmit data to sink based on this tree. In this paper, we propose an automaton based algorithm to construct aggregation tree by using energy and distance parameters. Automaton is a decision-making machine that is able-to-learn. Since network's topology is dynamic, algorithm should construct aggregation tree periodically. In order to aware nodes of topology and so, select optimal path, routing packets must be flooded in entire network that led to high energy consumption. By using automaton machine which is in interaction with environment, we solve this problem based on automat learning. By using this strategy, aggregation tree is reconstructed locally, that result in decreasing energy consumption. Simulation results show that the proposed algorithm has better performance in terms of energy efficiency which increase the network lifetime and support better coverage.

Keywords: Automata Learning, Wireless Sensor Networks, Data Aggregation, Energy Efficient, Spanning Tree

1. Introduction

Wireless Sensor Networks (WSNs) are networks that consist of low power nodes with limited processing ability. These nodes have sensors which sense light, temperature, jitter and etc. in the environment. These nodes are deployed in environment densely and randomly. In monitoring application, these sensor nodes sense data from the environment periodically and transmit these data to sink node. Since transmitting the data is the most costly function in the network and power of the nodes is limited and cannot usually be charged; this leads to decrease node's power quickly.

After some rounds, network nodes energy is ran out and this leads to situations which the network can not work anymore. To the points mentioned above in order to increase network's lifetime, number of transmitted data packet should be minimized [1,2].

Network nodes, after event occurrence and sensing data from the environment, forward the sensed data to the sink. In addition to sensed data, each node must transmit other node's data to the sink. As mentioned above, data transmission consumes node's energy quickly. The solution which is suggested to decrease the number of data transmissions is aggregation mechanism. Aggregation mechanism works as follow: each node senses data from the environment and receives other node's data, then aggregates these data, based on the aggregation function and transmits the aggregation result to the sink. Therefore aggregation decreases the data volume that is transmitted and this leads to less energy consumption. In addition to mentioned improvements, aggregation decreases collision and retransmission delay [3]. In aggregation algorithms, we must construct aggregation spanning tree [4]. The spanning tree is a tree contains all network nodes and doesn't have any loop.

Like routing algorithms [5], aggregation algorithms should also be aware of the network topology and based on these information and queries which are propagated by root, network nodes select aggregation function and

aggregate the data, and then forward aggregated data to sink.

Cluster based algorithms [6] needs only local information to construct aggregation tree, therefore they transmit fewer packets to construct the aggregation tree.

In [7], authors investigate the computational complexity of optimal data aggregation in sensor networks and show that it is generally NP-hard; they present some suboptimal data aggregation tree generation heuristics, Center at Nearest Source (CNS), Shortest Paths Tree (SPT) and Greedy Incremental Tree (GIT) and showed the existence of polynomial special cases. Different aggregation algorithms have been presented in recent years [1,4,6,8–10].

Espan [4] is an energy-aware spanning tree algorithm that constructs the aggregation tree to aggregate the data. In Espan, the source node which has the highest residual energy is chosen as the root and other nodes choose their corresponding parent node among their neighbors based on distance to the root and residual energy. In LPT [8] after selecting the node with most energy as root, each node selects neighbors with most energy as parent and its parent forwards its data to the sink.

In this paper, we present an automata based Energy Efficient Spanning tree (AEEspan) algorithm which is a new energy efficient aggregation algorithm for wireless sensor networks. The current work is a modified version of our already published papers [1,11]. In [1] we propose an energy aware data aggregation spanning tree algorithm. In [11], we present an automata based algorithm to construct spanning tree. Automata is an able-to-learn decision-making structure that selects the best action among a number of actions then wait for environment's response to this selection. By using the environment feedbacks, automata update the probability of the selection of each action among the set of actions and select best action for the next step. The main idea of proposed protocol is as follow: each node has an automaton to select the best nodes among its neighbors as its parent.

Since the status of the network is dynamic, the aggregation algorithm should construct the routing tree periodically. When a timer is expired our some nodes are failed in the network, the new aggregation tree must be constructed [4,8].

At the beginning of each time interval, routing packets are flooded into the network. In each routing packet has some routing information likes: number of hops to the root, remaining energy, number of child node. Each node selects optimal path to the sink, based on algorithm parameters. Since the node's energy is limited, transmitting and receiving this volume of routing information is not a good solution to construct aggregation tree. This overhead causes a lot of energy consumption. So, some nodes run out of energy quickly and fail. This causes network to be disconnected.

To solve this problem we use an automaton based approach; if a node in the aggregation tree fails, and a part of tree is disconnected, only this part of tree starts to reconstruct. So it is not necessary to flood routing packets into entire network. To do this, each node uses the environment feedbacks, and updates its automata.

The remainder of this paper is as follow: in section 2, we review some existing aggregation algorithms. The system model is given in section 3. In section 4, we present the proposed algorithm and the performance evaluation of proposed algorithm is presented in section 5. Finally, section 6 concludes the paper.

2. Related Work

Different aggregation algorithms have been presented in recent years. In this section we review them briefly. As presented in [9], DCTC algorithm dynamically constructs the aggregation tree for mobile target tracking. In the presented algorithm depending on the target location, a subset of nodes participates in tree construction.

In [12], the sink saves the entire network state and then by considering link cost, in centralized form, constructs the tree by minimum cost. In cluster algorithm [6], after partitioning the network into clusters, cluster's members construct aggregation tree and transmit data to cluster head. After aggregation, cluster heads transmit aggregated data to the sink in one hop or multihop manner [13].

Espan [4] is an energy-aware spanning tree algorithm that constructs the aggregation tree to aggregate the data. In Espan, the source node which has the highest residual energy is chosen as the root and other nodes choose their corresponding parent node among their neighbors based on distance to the root and residual energy. One of the most important problems of Espan is that the nodes with least distance to root may be selected as parent by many nodes. So these nodes consume their energy quickly and then they will be failed sooner than other network nodes.

In LPT [8] after selecting the node with most energy as root, each node selects neighbors with most energy as parent forwards its data to the sink. In the mentioned algorithm, when a node in the tree fails, the tree will be reconstructed.

In [1] an energy efficient algorithm, which constructs the aggregation tree, is presented. To prevent failing the nodes and to increase the network lifetime, the algorithm considers both the remaining energy and the distance parameters. Each node selects a node which has the most energy within neighbors as its parent. Furthermore, the distance from this parent to the root must be reasonable. To balance the energy and distance parameters, the algorithm uses path's energy and length parameters.

In [14], the proposed algorithm uses machine learning to transmit the sensed data to the sink. Learning algo-

rithm is executed in the sink and its result is propagated throughout the network. In [15] Q-leaner is used to construct aggregation tree, to maximize aggregation ratio.

In [16] an algorithm to construct aggregation tree, based on automata, is proposed. In this algorithm, in which each node is equipped with an automaton, the automaton selects a path for transmitting data via the path which the aggregation ratio is maximized. In [17], the algorithm considers an automaton for each node, which selects a path to transmit data to the sink in accordance with network conditions.

3. System Model

We consider a network of N sensor nodes uniformly distributed over a region and one sink. These nodes are non mobile. The sensor nodes have radio communications; two nodes can receive and transmit data if they are in communication range of each other. There are three types of data collection in sensor networks [18]. Event-based data, such as intrusion detection or object tracking, is collected when an event within the deployment region occurs. The event is confirmed by sensors and reported to the sink. State-based data is collected in response to a query sent to selected sensors requesting relevant data. Global state-based data, such as temperature or humidity, is collected by sensors all over the deployment area and is transmitted toward the sink. Our interest here is in global state-based data. All sensor nodes are sources, sense environment and transmit sensed data to the sink periodically. In the following subsections we describe the energy model and data transmission model used in this work.

3.1. Energy Model

We use the same energy consumption model described in [19]. In this model a sensor node consumes Eelec (J/bit) in transmitter or receiver circuitry and Eamp (J/bit/m2) in transmitter amplifier to achieve an acceptable signal noise ratio. A sensor node expends energy ETij (k) or ERi(k) in transmitting or receiving a k-bit packet to or from distance distij, given by the following equations:

$$ET_{ij}(k) = E_{elec} * k + E_{amp} * k * dist_{ij}^{\lambda} \quad (1)$$

$$ER_i(k) = E_{elec} * k \quad (2)$$

The exponent λ heavily depends on the communication medium [20]. In the current work, we assume that the transmission power is directly related to the squared distance, means λ=2, which hold for free space. When the distance is small, the free space propagation model is adopted for energy loss, and when the distance is large, the two-ray ground model is adopted for energy loss, which means λ=4.

In the above functions, k represents the length of transmitted and received data packet. Sensor nodes transmit or receive two types of packet; routing packet and data packet. Routing packets flood in the network to construct or reconfigure the aggregation tree. Data packets include data which sense by nodes from environment and are transmitted to sink. We assume the aggregation function is simple, for example, max or average function, so the input data length is equal to the output data length. Based on this assumption, all data packets in the network have the same length. According above descriptions, we should try to minimize not only distance but also the volume of transmitted and received packets.

As described in [6] if aggregation function is simple, the energy consumption for data aggregation will be negligible.

3.2. Data Transmission Model

After determining children of a node, a node creates a TDMA schedule and notifies its children about it. In the data transmission phase, children send their data to their parent according to the specified TDMA schedule. By using TDMA scheduling mechanism, we can solve the collision problem of data transmission, too. In addition, after sending data each node goes to sleep mode until next round, which cause power saving.

As described in [20], round is defined as the collection of one data unit from every node in the network and delivering the resulting aggregated data to the sink node. In every round, each parent in the tree will wait till it receives data from all its children. A node after participating in a round, wait until next round. Based on [20], lifetime of a tree is defined as the number of rounds that can be performed before the failure of certain percentage of total nodes. Therefore, in this paper, lifetime is defined as the failure of 10% of the total nodes of the tree.

3.3. Automata

Learning automata is an abstract model which has a finite set of actions as its input. Each member of the input set has a selection probability parameter. Automata select an input with highest selection probability as their output. Then the environment evaluates the selected action and responses to the automata. Automata use the response for learning process.

Learning process is as follow: if the environment response is unfavorable based on network parameter, the automata penalize the selected input by decreasing its selection probability and increasing selection probability of the other members of the input set members. But if the environment response is favorable, the automata reward the selected input by increasing its selection probability and decreasing selection probability of the other members of

the input set. The rewarding process increases selection probability of the awarded input for the next step.

As seen in Figure 1, an automaton is learned based on the feedback of the environment.

As described above, an automaton is defined by the quadruple {α, β, P, T} in which α= {α1, α_2, α_3… α_n} represent the output set, β= {β1, β_2, β_3… β_m} represent the input set, P= {p1, p2, p3… p4} represent probability set and finally p (i+1) = T [α, β, P] represent the learning process.

4. Proposed Algorithm

As mentioned in section 1, data aggregation tree construction algorithms construct tree periodically. To construct an aggregation tree, at the beginning each period, routing packets are flooded into the entire network to inform all nodes. After this step, each node selects the best path toward the sink node and transmits data via selected path until the next period. Transmitting these routing packets periodically consumes a lot of energy and has unfavorable overhead for the network.

In automata based algorithms [17,16], at the beginning, routing packets are flooded into the entire network. Each node considers each neighbor as entry in its routing table and then calculates the selection probability of each entry based on the algorithm's parameters, energy or distance and etc., and then each node selects the neighbor with highest selection probability as its parent and sends its data via this parent to root.

In [16] after receiving data, root sends acknowledgment to the sender node; this acknowledgment has some information for automata. Based on acknowledgment information, automata penalize or reward the path's nodes, on the way that if the selected path was optimal based on network parameters, selection probability is increased for the next step, but if selected path was not optimal, selection probability is decreased for next step. This process is called automata learning.

In the next steps, each node selects a new parent based on the updated selection probability of the nodes in the network and this process is repeated by the end of the network's lifetime. By using of this property of automata -learning-the algorithm prevents flooding the routing packets periodically, at the same time, by using ack information, nodes are aware from changes in network topology and paths are updated.

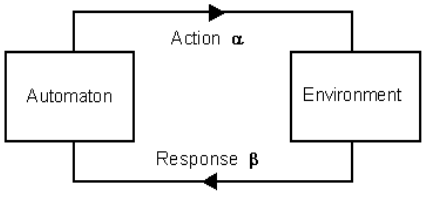


Figure 1. Learning automata.

Proposed algorithm works as follow: at the beginning, routing packets are flooded into the network by the way that each node sends its energy and the distance to root to all its neighbors. Each neighbor, after receiving this message, considers the sender as a new entry in its routing table, and puts sender ID, sender energy and sender distance to root as entry fields.

This sends/receives is performed in entire network, so each node maintains neighbors information in its routing table. Then the routing table entries are considered as input set of automata and the automata calculate the selection probability of each entry as follow:

$$Sel - prob = C_i * \frac{energy_j}{dis\tan ce_j} \quad (3)$$

In Equation (3), C_i is a constant which is calculated by node and is depended on the sum of energy and distance to root of entries in routing tables of node *i*. This means that node *i* adds energy and as well distance to root of all the input set members. Then automaton considers the result of dividing the entry energy by its distance to root multiply a fixed number (C_i) as the selection probability of the entry.

Each node selects neighbor with highest selection probability as its parent, nodes in the network sense data and aggregate them with collected data from their child, then send the result of aggregation to their parents. Their parents forward data to the sink by repeating this process.

In fact, trees show paths that each algorithm selects for transmitting data to sink. These paths have an important effect on energy consumption, if the algorithm selects shortest paths mostly, the nodes in these paths fail quickly, and in continue nodes must send their data via other paths that my be longer, which led to high energy consumption, decrease lifetime and disturb coverage of the network. And as well, if the algorithm selects paths by considering energy parameter as main parameter and does not regard to path's length, nodes send data via longer paths which causes to high energy consumption. Thus, the algorithms should consider parameters, not one parameter, and balance between these parameters.

In these work, we try to select an optimal path by considering both parameters-energy and distance-as main parameters and construct the tree based on both of them.

In order to update automata, each node must collect some information from the network. By using this information, an automaton becomes aware of network changing. In [17] to be aware of the network state, each node after receiving data sends feedback or acknowledgment message to the sender of the data and as mentioned before, this message has some information. By using these feedbacks, automata penalize or reward the selected parent, but sending these acknowledgments have a lot of overhead. In [16] to decrease this overhead, acknowledgment is sent after some data transmissions.

In the proposed algorithm, to be aware of network's changes, one solution that was presented in [11] worked as follow: each node broadcasts its energy and distance to root, in data packet, and does not send in a separate packet, so, node's neighbors, after receiving these packets, update their routing tables. This procedure, perform automata learning. Since node's information of network is updated, optimal path to root is continuously selected.

But, transmitting these addition data led to waste energy because parent's energy becomes less than other nodes in neighborhood after some rounds, mostly. Thus, transmitting additional data in each data packet, based on energy model that mentioned in section 3. A wastes energy.

So, we can improve algorithm performance by working as follow: If a node fails in aggregation tree or the node's energy is lower than a pre determined threshold, then the node's children select a new parent from the nodes in their neighborhoods. Then, it is not necessary to reconstruct aggregation tree globally and periodically. By using this strategy the tree is reconstructed when it is needed, and reconstruction packet broadcast locally. This leads to reduction in data transmission in the network and power saving.

Thus, the reconstruction section of proposed algorithm works as follow: by failing a node in tree, node's children select a new parent base on their automata inputs; each node's children broadcasts an update packet. The neighbors, after receiving this packet, send their information to packet's sender. Then, each sender node updates its automata and then, an input with highest selection probability is selected as a new parent. By using this property, we prevent flooding the routing packets, and also, nodes are aware from changes in the network topology and paths are updated. This process is repeated by the end of the network's lifetime.

In this work, the input and output sets-α, β- of each node's automaton are the entries of its routing table, and the probability set-P-is the set of selection probability of entries. To increase efficiency in proposed algorithm, learning process does not perform each round, but when energy consumption reaches a threshold, automaton, based on responses from environment, performs learning process.

Reconstruction property is an important section in tree construction algorithm that is noted rarely. In this work, we try to achieve two main goals:

- Construct an energy efficient tree by considering both energy and distance parameters.
- Add the reconstruction property, to prevent from flooding packets globally.

The pseudo code of the proposed algorithm which helps us to understand the details of the proposed algorithm is given in Figure 2. In this pseudo code, *m* represents the message which is sent by each node and contains three fields: node's ID, node's energy and node's distance to root. Input of AEEspan algorithm is the network nodes vector.

5. Performance Evaluation

In this section, using computer simulation, we evaluate the performance of the proposed algorithm and compare it with other algorithms [1,4,8] algorithm. We call the algorithm presented in [1], EEspan in below figures.

We consider a sensor networks with N sensor nodes randomly arranged in a 600m×600m region. The number of nodes (N) varies from 300 to 700. The initial energy of each node varies from 8J to 20J. The communication range of all nodes is set to 60 meter. The size of sensor data packet is 320 bits and a routing packet is 30 bits length. In the following curves the average values over 20 simulation experiments are depicted. Also, we assume all nodes in the network sense the area periodically and send their data to the sink node.

```
Define:
  Message M(sender_Id,sender_Energy,sender_Distance)

AEEspan(nodes)
{
  For each node N in nodes
  {
    Message M1(N.Id,N.Energy,N.Distance)
    N.Broadcast(M1)

    While(N.Receive(M2))
      N.Add_routing_table(M2.Id,M2.Energy,M2.Distance)
  }//flooding routing packets

  For each node N in nodes
  {
    For each entry E in N routing table
      N.Caculate(E.Selection_probability)

     N.Parent=Highest_selection_probability
       (N.Routing_table_entries)
  }//constructing aggregation tree

  While(true)
  {
    N.Transmit(Data,N.Parent)
    N.Receive(Data,N.Parent)
    If(N.Parent.Energy<Threshold)
      {
        N.Broadcast(Update_packet)
        For each entry E in N routing table
          N.Caculate(E.Selection_probability)

        N.Parent=Highest_selection_probability
          (N.Routing_table_entries)
      }//reconstruction
  }
}
```

Figure 2. The pseudo code of the proposed algorithm.

To do simulation, we use centralized version of LPT, however Espan, EEspan and AEEspan work in distributed manner. In below figure, to compare performance of the trees which are constructed by algorithms, we do not consider transmission and receiving energy for routing packet that flooded for tree reconstruction. As mentioned before, nodes send their data via the tree which is constructed by the algorithm, so it is important to compare paths that each algorithm selects to send data to sink.

At the first simulation trial, to evaluate the energy efficiency of proposed algorithm, AEEspan, each node is assigned with an initial energy that is randomly chosen between 8J and 20J. After some simulation rounds, we measure remained energy of network nodes. In Figure 3, sum of the remained energy of all nodes in network is plotted versus number of nodes for four algorithms.

Since LPT algorithm selects paths by considering only energy parameter, nodes transmit their data via longer paths which make higher energy consumption. In Espan algorithm, nodes transmit data via shortest paths, but by failing low power nodes in these paths, data must be transmitted via other paths which may be longer. While in EEspan [1] and AEEspan, nodes consume less energy, because in these algorithms, tree is constructed by applying a reasonable relation between energy and distance parameters, Unlike the algorithms given in [4,8] use only one of these parameters as the main parameter and the other parameter is used as lower priority parameter.

To verify above claim about path length, we study suggested paths in these algorithms. In the aggregation tree construction algorithms, average path length parameter represents average depth of tree that is the number of the hops between the nodes and the root, so, the tree with deeper branch means that nodes transmit data via longer path, and this causes more delay and also more energy consumption.

In Figure 4, the average path length is plotted versus the number of nodes. As in AEEspan, automata select their parents with the highest selection probability, and this value has converse relation to distance parameter, so the node with less distance has higher priority to be selected as parent that causes the parent with higher energy and less distance is selected.

As shown above, LPT tree has longer branches, because of not regarding distance parameter at all, while in Espan which regard distance as main parameter, tree has shorter branches. While in thus work, branches are between these two bounds.

By considering distributed algorithms, and apply tree construction cost, consumed energy to transmit and receive routing packet, we evaluate the performance of proposed algorithm by considering reconstruction section.

As describe earlier, the algorithm with automata learning property consumes less energy as a result of prevent flooding routing packet. By considering learning property, transmission volume is decreased that leads to more power saving. To show this, the remaining energy of network nodes is measured. In Figure 5, sum of the remained energy of all nodes in network is plotted versus number of nodes.

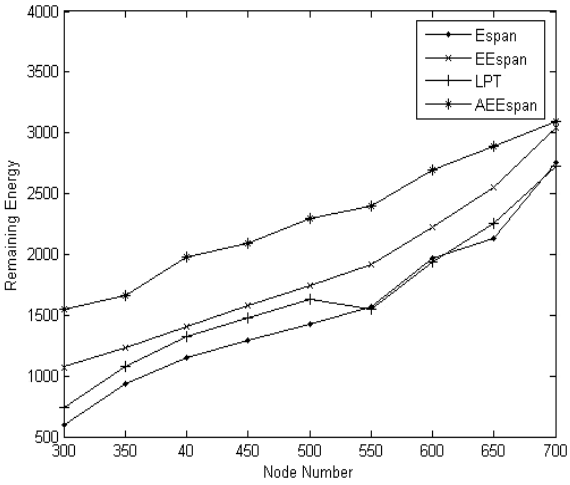


Figure 3. The remaining energy of algorithms without considering tree reconstruction cost.

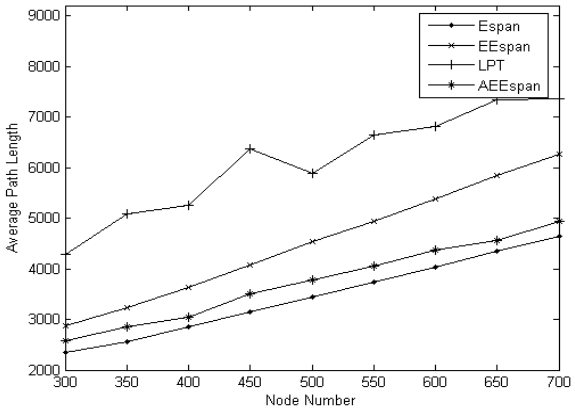


Figure 4. Average hop count to root.

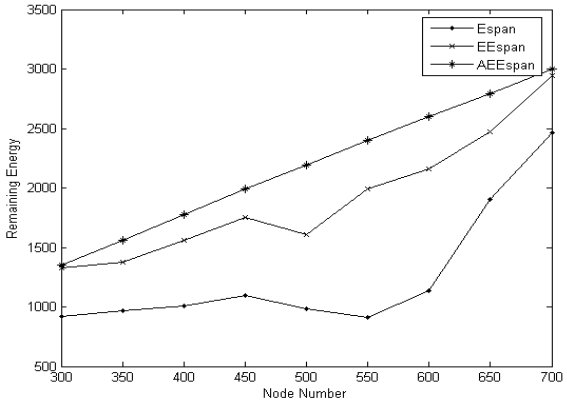


Figure 5. The remaining energy of distributed algorithms with considering tree reconstruction cost.

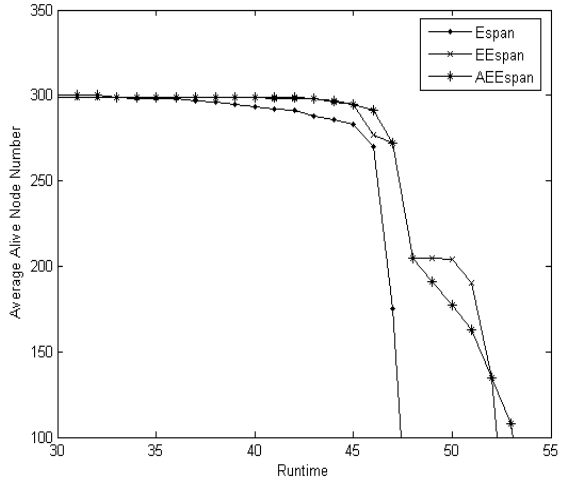


Figure 6. Number of alive nodes at N=300.

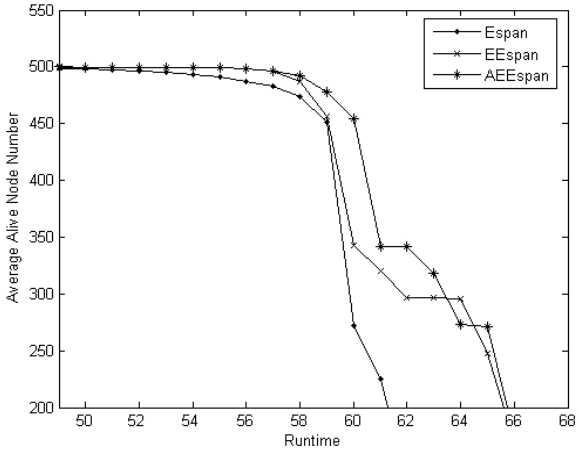


Figure 7. Number of alive nodes at N=500.

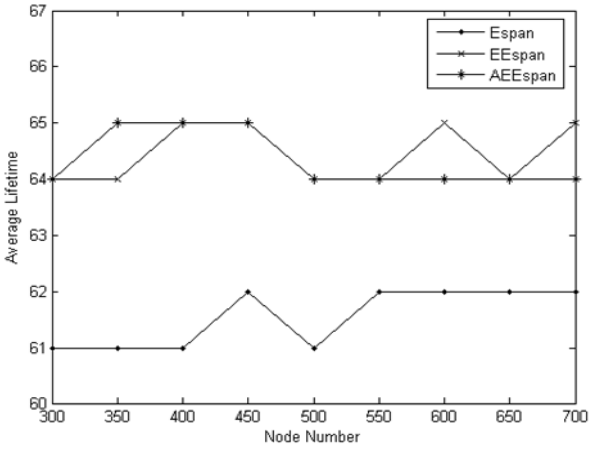


Figure 8. Average lifetime comparison.

We measure the number of alive nodes after each simulation round in Figure 6, 7 when N = 300, and 500 nodes, respectively. As in AEEspan, automata select a parent with the highest selection probability which has direct relation to energy parameter, so the nodes with low energy remain a longer time in the network rather than the other algorithms.

For example, in Figure 7, three algorithms work similarly by the round 52, but after that, nodes in Espan tree start to fail sharply, while in AEEspan, nodes failing start later and in slighter manner.

As mentioned before, energy efficiency is a main goal of algorithms in wireless sensor networks. By decreasing energy consumption that led to prevent from failing network nodes, network's coverage whether spatial or temporal is supported better and network's lifetime increases. AEEspan algorithm by decreasing transmission volume, can meet this goal.

In Figure 8, for these algorithms, the average lifetime is plotted versus the number of nodes. The results are obtained after 20 different simulation trials. As seen in Figure 8, the proposed algorithm has higher lifetime than other algorithms. Based on lifetime definition, lifetime has direct relation to alive node numbers.

6. Conclusion

One of the most important limitations of the wireless sensor networks is the network's energy. Aggregation algorithms have a considerable role in decreasing the energy consumption due to the reduction of the transmitted data volume. Aggregation algorithms construct the Aggregation tree based on the algorithm parameters, and determine the transmitting path of the root for each group. In this paper, the tree construction is presented based on automata. An automaton is an able-to-learn structure which tries to choose the best path to send the data to the root by getting feedback from the environment. Also, by prevent from flooding routing packet in entire network, proposed algorithm consume less energy. As the simulation results show, the proposed algorithm has more lifetime and lower energy consumption.

7. References

[1] Z. Eskandari, M. H. Yaghmaee, and A. H. Mohajerzade, "Energy efficient spanning tree for data aggregation in wireless sensor networks," ICCCN, 2008.

[2] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey, computer networks," Computer Networks Journal, 2002.

[3] Y. Hu, N. Yu, X. H. Jia, "Energy efficient real time data aggregation in wireless sensor network," IWCMC, 2006.

[4] M. Lee and V. W. S. Wong, "An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks," IEEE, 2005.

[5] J. N. Al-Karaki, and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey, supported by the ICUBE initiative of Iowa State University, Ames.

[6] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE, 2004.

[7] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," International Workshop on Distributed Event-Based Systems, 2002.

[8] M. Lee and V. W. S. Wong, "LPT for data aggregation in wireless sensor networks," IEEE GLOBECOM, 2005.

[9] W. Zhang and G. Cao, "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," IEEE, 2004.

[10] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A low latency and energy-efficient algorithm for convergecast," IEEE GLOBECOM, 2003.

[11] Z. eskandari, M. H. Yaghmaee, A. H. Mohajerzade, "Automata based energy efficient spanning tree for data aggregation in wireless sensor networks," IEEE ICCS, 2008.

[12] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A lowlatency and energy-efficient algorithm for convergecast," EEE GLOBECOM, 2003.

[13] Y. P. Chen, A. L. Liestman and J. Liu, "A hierarchical energy-efficient framework for data aggregation in wireless sensor networks," IEEE, 2006.

[14] P. Radivojac, U. Korad, K. M. Sivalingam and Z. Obradovic, "Learning from class-imbalanced data in wireless sensor networks," IEEE VTC, Fall 2003.

[15] P. Beyens, M. Peeters, K. Steenhaut, and A. Nowe, "Routing with compression in aireless sensor networks: A Q-learning approah," AAMAS, 2005.

[16] M. Esnaashari, M. R. Meybodi," "A learning automata based data aggregation method doe sensor networks," CSICC, 2007.

[17] M. Ankit, M. Arpit, T. J. Deepak, R. Venkateswarlu and D. janakiram, "TinyLAP: A scalable learning automata-based energy aware routing protocol for sensor networks," IEEE, 2006.

[18] Y. P. Chen, A. L. Liestman, J. Liu, "Energy-efficient data aggregation hierarchy for wireless sensor networks," Proceedings of the 2nd Int'l Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks, 2005.

[19] J. Kamimura, N. Wakamiya, and M. Murata, "Energy-efficient clustering method for data gathering in sensor networks," BROADNETS, 2004.

[20] S. Upadhyayula and S. K. S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks," Elsevier, 2006.

Wireless Sensor Network, **2009, 1, 324-333**
doi:10.4236/wsn.2009.14040 Published Online November 2009 (http://www.scirp.org/journal/wsn).

H-TOSSIM: Extending TOSSIM with Physical Nodes*

Wenjun LI[1], Xiaobin ZHANG[2], Weihua TAN[2], Xiaocong ZHOU[2]
[1]*School of Software, Sun Yat-sen University, Guangzhou, China*
[2]*School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China*
Email: *lnslwj@mail.sysu.edu.cn*

Abstract

As the development of Wireless Sensor Network (WSN), software testing for WSN-based applications becomes more and more important. Simulation testing is an important approach to WSN-based software testing, and TOSSIM is the most widely used simulation testing tool targeted at TinyOS which is the most popular operating system nowadays. However, simulation testing tools such as TOSSIM can not reveal program errors about communication detail or timing, and lack accurate power consumption model and even can not support power consumption estimation. In this paper, a hybrid testbed H-TOSSIM is proposed, which extends TOSSIM with physical nodes. H-TOSSIM uses three physical nodes, of which, one shares the simulated environment with all virtual nodes to test the WSN program, and the other two bridge the real world and the simulated environment. H-TOSSIM combines the advantages of both the simulation in physical node and the simulation testing tools in WSN software testing. Through experiments, we show that H-TOSSIM really reveals program errors which the pure simulation testing can not capture, and can support power consumption estimation for large WSN with high accuracy and low hardware cost.

Keywords: Wireless Sensor Network, Sink, Principal Node, Superior Node, Network Lifetime

1. Introduction

1.1. Background

Recent advances in electronic technology and the need of practical applications enable the rapid development of Wireless Sensor Network (WSN), which consists of many resource-limited sensor nodes, and can monitor the phenomena in the physical world. WSN can be applied in military surveillance, environmental monitoring, health diagnostics, home automation, etc [1]. One of the primary challenges in the researches on WSN is software testing. A sensor network is self-configuration, and its nodes are low-power embedded devices, which make its software testing challenging.

Simulation testing and hardware-in-the-loop (HIL) testing are the main approaches to WSN software testing. There exist many simulation tools for sensor networks. In simulation testing, the sensor network environment is simulated through pure PC software, which is controllable, convenient and low-cost. HIL testing is one kind of important means for embedded software testing. Commonly, HIL testing tools for WSN consist of dozens of physical sensor nodes. In HIL testing, the program under test runs in the physical sensor nodes with some assistant middleware. Compared with simulation testing, HIL testing can reveal more defects; however, it is high-cost and not so convenient.

TOSSIM [2–4] is one of the most widely used simulators for WSN, which is designed for TinyOS [5–7] programs. TinyOS is the most popular operating system for WSN nowadays, which supports almost all popular sensor nodes; and its latest release is TinyOS 2.x, which is corresponding to TOSSIM 2.x. In this paper, only TOSSIM 2.x instead of TOSSIM 1.x is considered. TinyOS is component-based and event-driven, and TOSSIM simulates the sensor network through replacing some low-level components and introducing a discrete event queue. As a TinyOS WSN simulator, TOSSIM is accurate and scalable. With its help, developer can test the program before TinyOS application is deployed.

1.2. Motivation

Simulation testing tools such as TOSSIM have some problems in WSN software testing. Firstly, they are difficult to reveal program defects and faults related with communication details or timing. Secondly, they are hard

*Supported by the National Natural Science Foundation of China under Grant No. 60673050.

to include an accurate power consumption model. And these problems are mainly caused by pure software simulation.

Taking TOSSIM as the representative, it has the following defects and inadequacy. The first is that TOSSIM can not reveal length setting error of message sending. In a TinyOS program, message sending is a common operation, and when sending a message, its length must be set correctly. However, even if the length of a message is set to be less than its intended size, TOSSIM can not reveal this fault when testing the program. Yet exceptions will occur for such program to run in the physical sensor network because messages will be partly lost.

The second is that TOSSIM can not reveal task calculation overload problem. Task is a deferred procedure call in TinyOS, which is used to complete some calculation. For example, a TinyOS program sends a message every 100 ms, and posts a task which includes 200 thousands multiplication before each sending. In the simulation testing of TOSSIM, such program works fine. However, when running in the physical sensor network, the task calculation overload problem will occur: the number of messages a node sends per second is much less than the expectation (about 10 messages, in this case).

There is also an inadequacy in TOSSIM; it does not support the power consumption estimation of sensor nodes, which is an important issue in WSN software testing because most sensor nodes are power-limited. Though PowerTOSSIM [8], a pure software extension to TOSSIM, can estimate the power consumption of sensor node, yet it is not so accurate and supports only one kind of sensor node. HIL testing can also estimate the power consumption of sensor nodes through digital multimeter; however, its hardware cost is too high because it needs dozens of physical sensor nodes.The problems existing in the simulation testing tools such as TOSSIM impede the comprehensive testing for WSN software, which may increase the cost of the application development. And these problems are difficult to solve by pure software extension.

2. Related Work

There exist many testing tools dedicated to WSN software testing. In the following discussion, ns-2 [9], SensorSim [10], EmTOS [11], PowerTOSSIM, AMETU [12] and avrora [13] belong to simulation testing tools; TOSHILT [14], MoteLab [15] and DSN [16] belong to HIL testbed.

Ns-2 is a universal network simulator which has been popular for many years. SensorSim is an extension to ns-2, which integrates some WSN features. Both ns-2 and SensorSim can not support TinyOS program directly.

EmTOS is an extension to EmSim [17] which is designed for EmStar [17], another WSN operating system. EmTOS can simulate heterogeneous WSN, which supports both EmStar and TinyOS programs. PowerTOSSIM is an extension to TOSSIM, and supports the power consumption estimation of the node; however, its error rate can be up to 13% and it supports only one kind of sensor node.

AMETU and avrora are both fine-grained TinyOS program simulators, and they both simulate the WSN in instruction level. The differences between them are mainly the synchronization strategy for different nodes.

Because of the simplification of the network layer, these simulators are difficult to reveal program faults related to communication details. And they also can not reveal program faults related to timing since they do not model the practical capability of sensor nodes.

TOSHILT, MoteLab, and DSN are HIL testbeds, all of which consist of dozens of physical sensor nodes. The differences of them are mainly the connection type between sensor nodes and the console. All of them can reveal more faults than simulation testing; however, their hardware cost is too high and they are not convenient when testing WSN programs.

3. Proposed Solution

As discussed above, pure software extension is not the solution to solve the problems existing in simulation testing tools such as TOSSIM. Instead, physical nodes are considered here because they are the target platforms for WSN software and may capture more problems. So, the solution which combines the physical nodes and the simulated environment is proposed in this paper. This solution is called H-TOSSIM, which extends TOSSIM with physical nodes. In H-TOSSIM, not all TinyOS programs run in the physical nodes, because that costs too much and is inconvenient. H-TOSSIM is a hybrid testbed. In fact, there will be just only one physical node in the

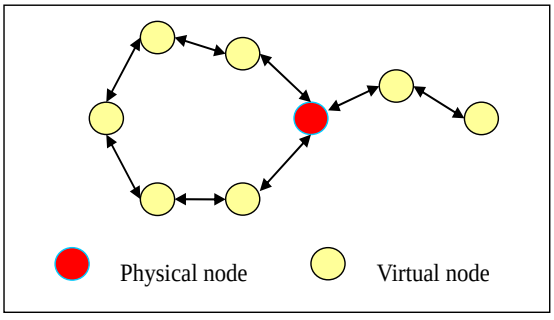


Figure 1. An example of the tested WSN topology in H-TOSSIM.

tested WSN topology of H-TOSSIM, and others are all virtual nodes. The only physical node can be configured to be a neighbor of any virtual node. As shown in Figure 1, in H-TOSSIM, one physical node interacts with other virtual nodes so as to test the TinyOS program, and all nodes run the same program. So the potential faults which pure simulation testing tools can not reveal will be captured through the interaction between the physical node and other virtual nodes. Another advantage of H-TOSSIM is that the power consumption of a node in a large WSN can be estimated through digital multimeter with low hardware cost.

In H-TOSSIM testbed, the physical node runs in the real world, and the virtual nodes run in the simulated environment of PC, so two extra physical nodes are needed as dual base stations to bridge the physical node and the virtual nodes. It means that H-TOSSIM totally needs three physical nodes.

4. Design of H-TOSSIM

4.1. Overview of the Architecture

Figure 2 shows the overview of the architecture of H-TOSSIM, which consists of a PN, a pair of DBS, two SFs, an MTTS, an ESECT and a GNB. PN is a physical sensor node which runs the TinyOS application under test. DBS consists of two base stations, which bridges the PN and the PC side of H-TOSSIM. SF is a tool provided by TinyOS to support serial communication, of which, one end connects the serial port and communicates with one of the DBS, and the other end may communicate with any PC program through socket. MTTS provides the services of messages transformation and transfer, and its both ends are separately two SFs and ESECT. ESECT is an extension to TOSSIM which aims to implement the

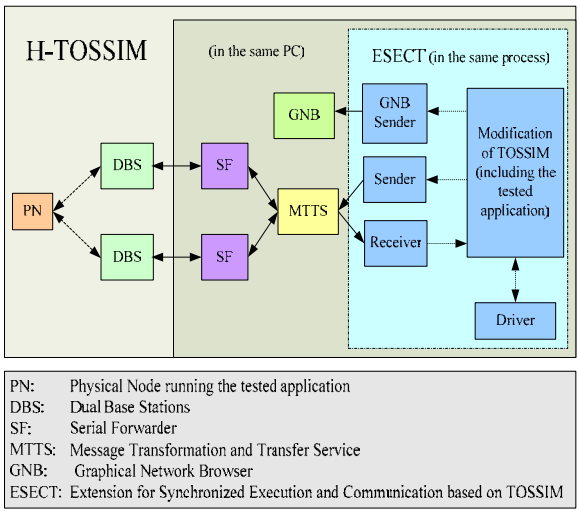


Figure 2. The architecture of H-TOSSIM.

synchronized execution and communication for the virtual nodes with the only physical node. ESECT includes five parts: the modification of TOSSIM, a driver, a receiver, a sender, and a GNB sender. GNB is a graphical network browser, which displays the network interaction situation in a GUI (Graphical User Interface).

In the following of this section, DBS, MTTS, ESECT and GNB will be introduced in detail. PN will be referred in the design of DBS. And since SF is a tool from TinyOS, it is discussed briefly in the design of MTTS.

4.2. DBS

DBS is mainly used to transfer the messages between PN and the serial communication. Herein, we first explain why DBS but not single BS is used. There are two reasons. The first reason is to make messages sending from the virtual nodes to PN become concurrent. In H-TOSSIM, PN may have several virtual neighbors; however, all messages sent from the virtual nodes to PN are serialized in ESECT. Yet if we want to finds out more program faults through PN, we should test the case that PN receives messages concurrently. So DBS is adopted. With DBS, messages sent in high rate from the virtual nodes will be forwarded to each of the DBS alternatively, which will bring concurrency because of the relatively low-speed DBS.

The second reason is that wireless radio rate is approximately as twice as serial rate. There are many kinds of physical sensor nodes, and the radio rates of some of them can be up to 250 Kbps [18,19]. However, the BS connects with the PC through serial communication, of which the rate is just up to 115 Kbps. When PN sends messages in a high rate, there will be blocking between the BS and PC if only one single BS is used. That is why DBS is used in H-TOSSIM. With DBS, the transfer rate between PN and PC can be up to 230 Kbps, which is high enough because the serial messages are usually shorter than the corresponding radio messages.

DBS physically consists of two BS's; however, it is not the simple combination of two BS's in software. The main differences between DBS and BS are reflected on the direction from PN to PC. In the direction from PC to PN, each of the DBS transfers every message received from serial to radio. However, in the other direction, each of the DBS only transfers half of the messages it receives from PN, and drops every message from the other BS. In order to make each of the DBS transfer half of the messages from PN, every message sent from PN is flagged to designate which of the DBS should deal with it. In H-TOSSIM, the source field of the message is chosen as the location of the flag because it can be retrieved in ESECT. The work of flag setting is done by a modified low component of TinyOS in PN, and the tested.

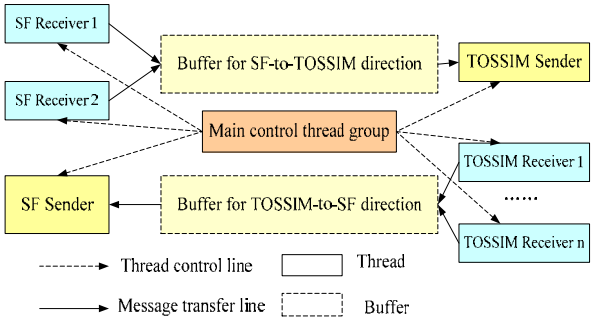


Figure 3. Threads relations of MTTS.

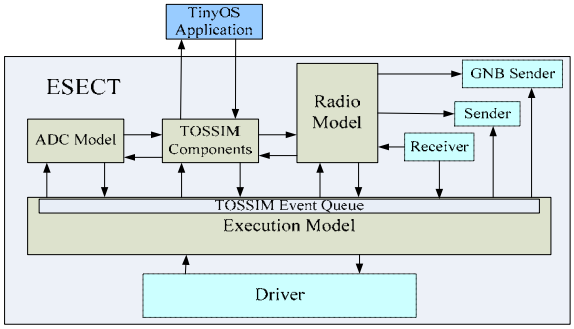


Figure 4. The architecture of ESECT.

application does not need any modification. In fact, DBS should also deal with message format transformation between radio messages and serial messages; however, it is rather trivial with the help of TinyOS components.

4.3. MTTS

MTTS connects two SFs with ESECT. An SF communicates with a BS through a serial port in one end, and provides a socket server in the other end. MTTS connects to two SFs as a client and provides a socket server for ESECT. Figure 3 shows the threads composition of MTTS.

In one end of MTTS, there are two SF receivers which will get SF messages from two SFs and put them into the buffer for SF-to-TOSSIM direction, and there is also an SF sender which handles with sending messages from the buffer for TOSSIM-to-SF direction to two SFs alternatively. And at the other end, there will be at least one TOSSIM receiver which gets TOSSIM messages from ESECT and put them into the buffer for TOSSIM-to-SF direction, and there will be only one TOSSIM sender which is used to send messages from the buffer for SF-to-TOSSIM direction to any clients connected with MTTS. In short, messages from two SFs are aggregated to any MTTS client in SF-to-TOSSIM direction, and messages from any MTTS client are dispatched to two SFs alternatively in TOSSIM-to-SF direction. And there is also a main control thread group which manages all the senders and receivers.

Besides messages transfer, MTTS should also take charge of message formats transformation between SF message and TOSSIM message. SF message format is similar to that of serial message except an extra field called as AM type is added at the head of SF message. And TOSSIM message format is the same as that of serial message when just considering the header and the data region of the message. Though there are other parts in TOSSIM message, only the header and the data region are considered in MTTS because ESECT will manage the other parts of TOSSIM messages. So, message formats transformation in MTTS is mainly implemented by the adding or removing of the AM type fields. And except AM type fields, network byte order of some other fields in the message may also be changed since network byte order can be different between both ends. All this transformation is done by each direction's buffer.

4.4. ESECT

Figure 4 shows the architecture of ESECT. In ESECT, execution model, radio model, ADC model and TOSSIM components belong to the original TOSSIM; driver, receiver, sender and GNB sender are newly introduced.

Execution model is the foundation of TOSSIM, and it is based on a discrete TOSSIM event queue. In TOSSIM, the simulated WSN is driven by TOSSIM events. A TOSSIM event is different from a TinyOS event which is a kind of procedure call; however, a TOSSIM event is a structure which is associated with a virtual clock. All TOSSIM events in the event queue are ordered ascendingly according to the virtual time. And the running of TOSSIM is in accordance with the ordered TOSSIM events.

Radio model and ADC model are separately used to model the radio environment anc d the sensing environment. TOSSIM components are used to replace those low-level and hardware-specific components. All these components establish the simulated environment.

In the following, the newly introduced parts will be discussed, which enable the simulated environment to interact with the physical sensor node normally.

Driver The driver builds the simulated environment which is shared by all nodes, drives the simulated WSN, and synchronizes the single physical node and the virtual nodes.

The driver first creates the topology of the network and the noise of each node based on a configuration file. In order to make PN share the same simulated environment with all virtual nodes, a virtual agent which represents the single physical node is created in the simulated environment. Figure 5 shows the interaction between PN and the virtual nodes through the virtual agent. In ESECT, messages sent from the virtual nodes to PN are first sent to the virtual agent, and then sent to PN by the sender of ESECT; however, messages from PN are di-

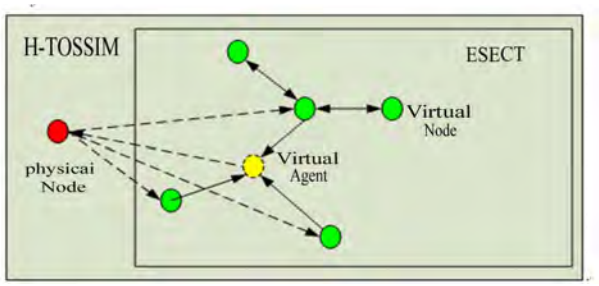


Figure 5. Interaction between PN and the virtual nodes.

rectly sent to the virtual nodes. In fact, the virtual agent here is only a stub which does not run the TinyOS program under test. After building the simulated environment, the driver will also establish all connections to MTTS and GNB if possible.

In order to drive the simulated environment, the driver sets the boot-up time for all virtual nodes (excluding the virtual agent) by inserting corresponding TOSSIM events into the discrete event queue. When the simulation starts, the driver gets the latest TOSSIM event continually from the discrete event queue, and then runs the event handler. Because every TOSSIM event is related to a node, the execution of the event handler causes the corresponding node to take some actions, which may produce some more TOSSIM events such as to assure the running of ESECT.

The time associated with a TOSSIM event is virtual, but the time in PN is real. They are very different. Therefore synchronization is essential to maintain a correct interaction between the virtual nodes and PN. Generally speaking, the virtual clock ticks faster than the real clock when simulating not too large WSN applications. So when fetching the latest TOSSIM event, the driver checks whether the virtual time is faster than the real time; if so, the driver will sleep until the real time is equal with the virtual time, and otherwise it will execute the event handler immediately.

Receiver The receiver in ESECT is used to receive messages from MTTS and forward them to the neighbors of PN. The receiver is not controlled by the driver; instead, it inserts new TOSSIM events about message receiving for the driver. When the receiver receives a message, it creates a new TOSSIM message according to the one received and the ID of PN, and then deliver it to the message list of any virtual node next to PN. The receiver also creates a TOSSIM event for every virtual node next to PN when sending a message to it.

Sender The sender is controlled by the driver and starts each time the event handler of the virtual agent is executed. In fact, there are only events about message receiving in all TOSSIM events of the virtual agent because it does not run actually. So, the occurrence of a TOSSIM event of the virtual agent means that a virtual node sends a message to PN. Meanwhile, the sender will fetch the message in the message list of the virtual agent, and send it to MTTS.

GNB Sender The GNB sender manages sending network interaction information to GNB, and starts every time a TOSSIM event about message receiving occurs. If a TOSSIM event is about message receiving, it records both the source and the destination of the message. And when the GNB sender starts, it creates a short message which includes the source and the destination of the message according to the information of the occurring TOSSIM event, and then sends it to GNB.

4.5. GNB

GNB is a graphical browser for the tested WSN, showing the network interaction dynamically. Figure 6 shows the graphical interface of GNB, which is displaying the interaction of an 8-node network. In GNB, each circle represents a node, and the color of the single physical node is different from others. An array represents a message from the rear of the array to the head of the array.

GNB consists of two threads, of which, one is used to show and update the interface, and the other is used to receive short messages from ESECT. The positions of the nodes in GNB can be random or designated by the user. When ESECT starts, GNB collects the interaction information continuously, and updates the interface periodically. Through GNB, the tester can get an overall sight of the WSN application under test, which is helpful for revealing some defects and faults in the program.

5. Evaluation

In this section, we show that H-TOSSIM really solves the problems existing in pure simulation testing tools

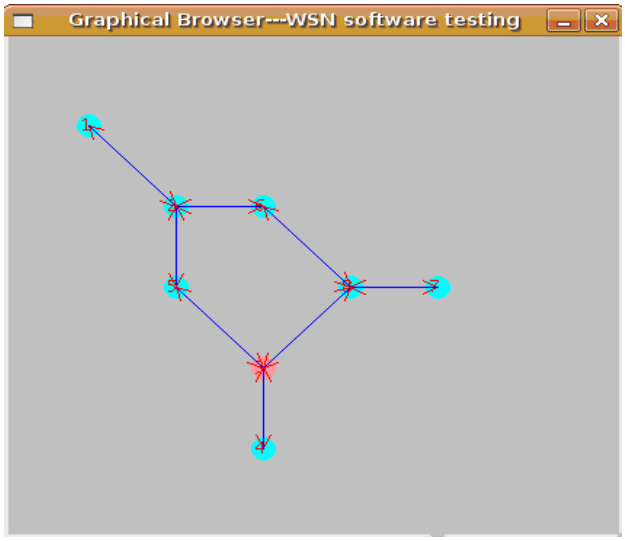


Figure 6. The graphical interface of GNB.

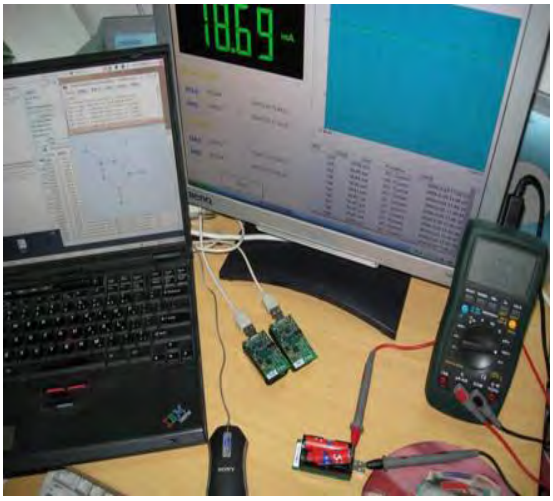


Figure 7. The testbed of H-TOSSIM.

such as TOSSIM. Figure 7 shows the testbed of H-TOSSIM. On the left, there is a laptop running two SFs, MTTS, ESECT, and GNB; in the center, there are three physical nodes, of which the two side-by-side nodes are used as DBS, and the remaining one is PN; on the right, there is a digital multimeter which is used to record the current of PN, and its result is saved to the desktop. The digital multimeter and the desktop are an option for estimating the power consumption of PN.

In the rest of this section, the applications under test we choose are *typical*, which means: 1) they were developed by the scholars who designed and implemented the TinyOS and TOSSIM, and distributed along with the TinyOS 2.x Package; 2) many researches on WSN testing also use these applications for evaluating their simulators. Besides, for better understanding of the advantages of H-TOSSIM against TOSSIM, comparisons of testing the same application are made.

5.1. Revealing the Length Setting Error of Message Sending

In the nesC programming, before sending out a message via the radio, the code must explicitly depicts the length of the package. Hence it has a chance that the declared length and the actual length of the package are not corresponding. In the real network, the radio component of a mote sends out the data according to the declared length, so the above case possibly leads to a incomplete package and unpredictable errors. However, TOSSIM, for the consideration of scalability, simulates the package sending by delivering a pointer to the package in the computer memory from the source node to the destination node, instead of the entire package, and this mechanism makes it can't reveal the length setting error of message sending. H-TOSSIM has a physical network, which behaves identical to any node in the real network, so it has the ability to reveal this error.

```
typedef nx_struct RadioToBlinkMsg2 {
    nx_uint16_t nodeid;
    nx_uint8_t group;
    nx_uint8_t value;
} RtoBGroupMsg_t;

RtoBGroupMsg_t message
```

```
typedef nx_struct RadioToBlinkMsg {
    nx_uint16_t nodeid;
    nx_uint16_t counter;
    nx_uint8_t flag;
} RtoBFlagMsg_t;

RtoBFlagMsg_t message
```

Figure 8. The structures of the two types.

In the following experiments, the program called as RadioToBlink is tested. This program sends two types of messages periodically, and these two types are separately called as RtoBGroupMsg_t and RtoBFlagMsg_t. Figure8 shows the structures of the two types. RtoBGroupMsg_t message is 4 bytes, and RtoBFlagMsg_t message is 5 bytes

In the two types of messages, the nodeid field is the source id of the message; the counter field is the value of a variable kept in the program which will increase by 1 whenever a RtoBGroupMsg_t message is sent; the group field and the value field are separately the high byte and the low byte of the counter; the flag field is the value of the lowest 3 bits of the counter.

We implant a fault in this program: the length of **RtoBFlagMsg_t** message is set to be 4 bytes when sending it out. In such a case, this type of message will be partly lost. We test the program in TOSSIM and H-TOSSIM. Figure 9 and 10 show the tested network topologies in TOSSIM and H-TOSSIM.

Figure 11 and 12 give the testing results, which show the messages received by node 2. From Figure 11, it can be shown that all **RtoBFlagMsg_t** messages received are normal. It means that TOSSIM can not reveal the length setting error of the program. However, from Figure 12,

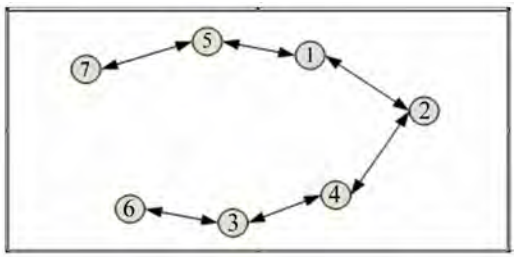


Figure 9. The tested network topology in TOSSIM.

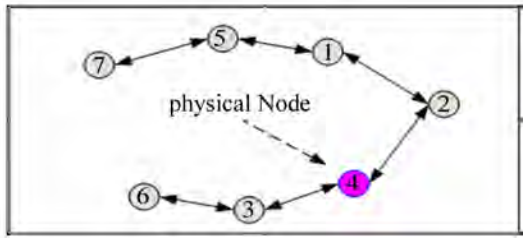


Figure 10. The tested network topology in H-TOSSIM.

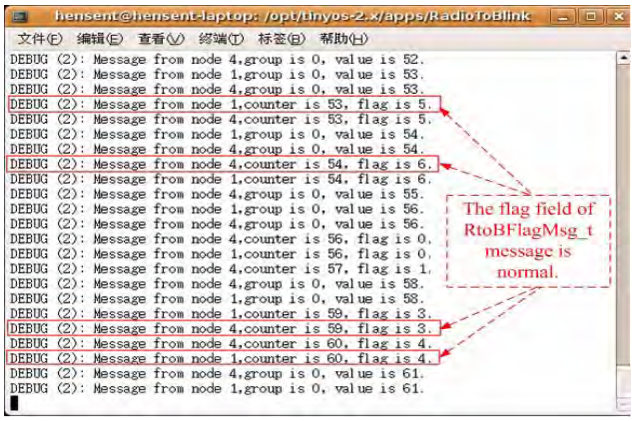


Figure 11. The debugging information for RadioToBlink in TOSSIM.

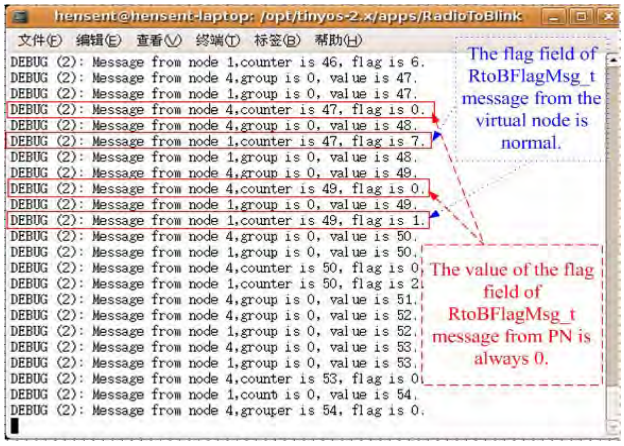


Figure 12. The debugging information for RadioToBlink in H-TOSSIM.

we can see that the flag fields of **RtoBFlagMsg_t** messages received from node 4 keep being 0, which means that the flag field is lost. That is to say, H-TOSSIM can reveal the length setting error through the comparison of PN and other virtual nodes.

5.2. Revealing Calculation Overload Problem in a Task

In the real network, while the mote is processing a heavy task, it possibly ignores program interrupts because there is not enough CPU resource to handle the interrupt. Consequently, it leads some unexpected errors, such as loss of package. So the programmer needs to test whether his application will has a defect causing the mote into a calculation overload status. However, events in TOSSIM, by the mechanism of discrete event, are considered as completion in a snap in the simulated virtual environment. As a result, the calculation overload problem never occurs in TOSSIM. However, this situation exists in the physical node of H-TOSSIM, which is helpful for the developer to find out his application's defect.

In the following experiments, the program called as **BlinkToRadio** will be tested. This program sends a **BtoRFlagMsg_t** message every **T** time, and posts a task to do some processing before each message sending. The structure of the **BtoRFlagMsg_t** message is the same with that of **RtoBFlagMsg_t** message. And the **counter** variable in the program will increase by 1 when sending a message. Here **T** is set to be 200 ms, and there is 300000 times multiplication in a task.

We test this program in both TOSSIM and H-TOSSIM for 10 seconds. And each node is expected to send 50 messages totally. The testing network topologies in TOSSIM and H-TOSSIM are the same with that of the previous testing.

Figure 13 and 14 give the testing results, which focus on the messages received by node 2. From Figure 13, it can be shown that the value of the counter field is approximately 50 finally, which is as expected. However, this does not mean that the program is correct when it is

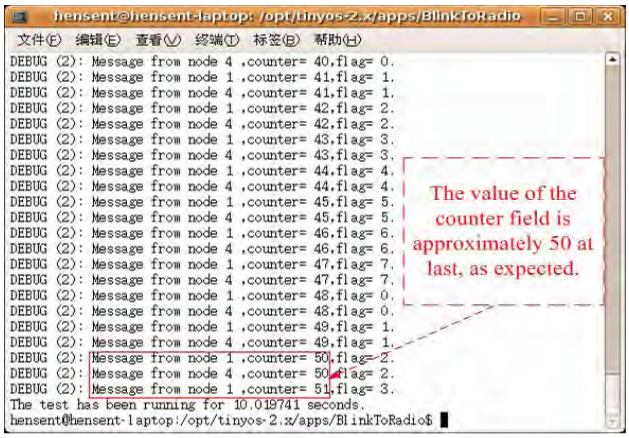


Figure 13. The debugging information for BlinkToRadio in TOSSIM.

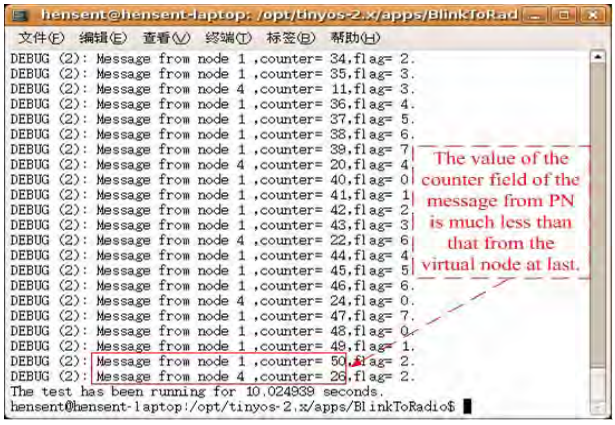


Figure 14. The debugging information for BlinkToRadio in H-TOSSIM.

deployed. From the result of H-TOSSIM, we can see that the value of the counter field of the message from PN is much less than that of the virtual node, which indicates that the calculation of the task in the program is overladen. That is to say, H-TOSSIM can reveal the calculation overload problem in a task.

5.3. Estimating the Power Consumption

H-TOSSIM needs a digital multimeter when it estimates the power consumption of a node; however, its advantage is that it supports the power consumption estimation of single node in a large WSN with low hardware cost. In this section, we first justify that H-TOSSIM is necessary and useful; and then we evaluate the accuracy of H-TOSSIM; finally, to show the advantage of H-TOSSIM, we use it to estimate the power consumption of a single node in different size of WSN.

In the following experiments, a program called SensorToRadio is used. This program reads sensing result every second and sends it out as a message. When the program receives a message, it will do some processing, and then forwards it if it is a new value. The program will be tested for 150 seconds every time. Because the voltage of PN can be kept 3V for a time, average current is used to measure the power consumption.

We test the program in three different sizes of physical sensor networks and estimate the power consumption of a node in the networks. Figure 15 shows the average current of a node in these three networks. We can see that the power consumptions of a node in different sizes of networks are different. So H-TOSSIM is necessary and useful, we can use it to estimate power consumption for different sizes of WSN with only three physical nodes.

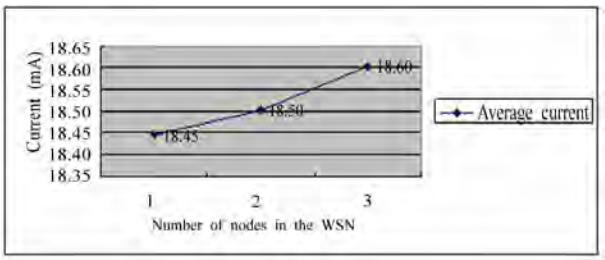


Figure 15. Average current of A node in different size of WSN.

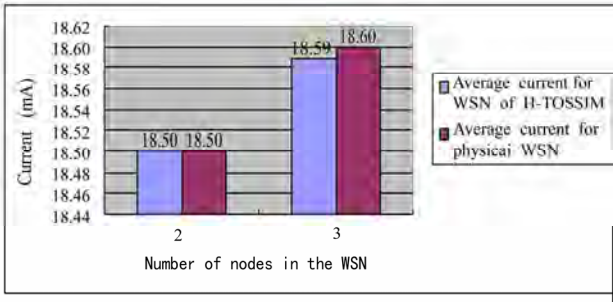


Figure 16. Power consumption estimation: H-TOSSIM Vs Physical WSN.

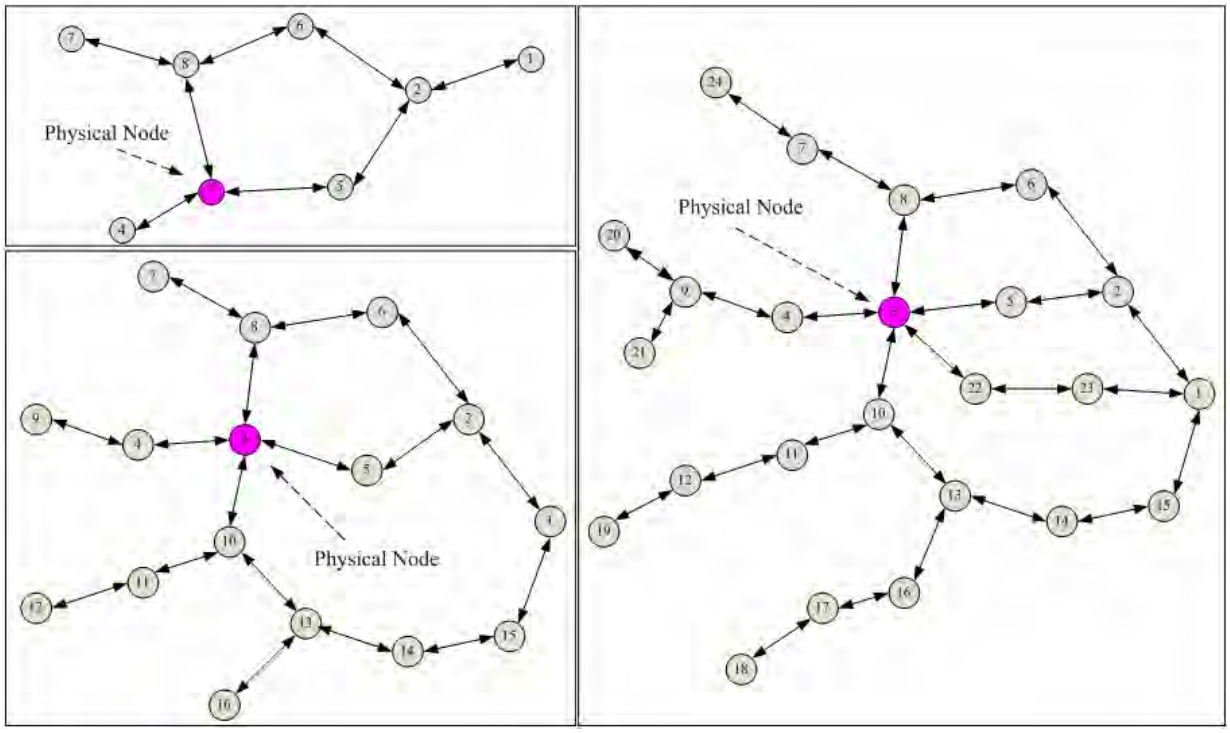


Figure 17. The three different sizes of network topologies.

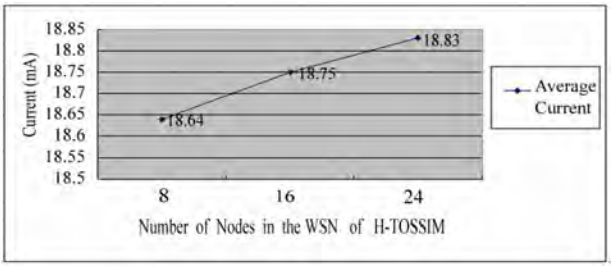


Figure 18. The estimation results for three different sizes of network topologies.

In order to evaluate the accuracy of H-TOSSIM, we compare H-TOSSIM and physical sensor network in power consumption estimation. Because of the limit of the number of physical nodes, we compare the sensor networks with 2 and 3 nodes only. Figure 16 shows the estimation results. We can see that the results are the same for the network with 2 nodes, and for the network with 3 nodes the results are subequal. That is to say, it is accurate for H-TOSSIM to estimate the power consumption of a node in the network.

Finally, we test the program with H-TOSSIM in three different sizes of WSN, and estimate the power consumption of PN. Figure 17 shows the testing network topologies. And the estimation results are shown in Figure 18. The average currents of PN for these three different topologies are separately 18.64 mA, 18.75 mA and 18.83 mA. Through these testing, we show the advantage of H-TOSSIM that it can estimate power consumption for large WSN with low hardware cost.

6. Conclusions and Future Work

In this paper, we first analyze the problems existing in pure simulation testing tools such as TOSSIM. Then we propose H-TOSSIM, a hybrid testbed, which extends TOSSIM with physical nodes. In H-TOSSIM, a physical node shares the same simulated environment with all virtual nodes so as to test a WSN program. H-TOSSIM combines the advantages of both the physical node and the simulated environment in software testing. Through experiments, we show that H-TOSSIM solves the problems existing in pure simulation testing tools with low hardware cost.

For the future work of H-TOSSIM, it uses only one kind of combination pattern between the physical nodes and the simulated environment; however, there are other combination patterns which are worth considering.

The first consideration is to use the physical nodes to provide signal gains between different nodes for the simulated environment. Signal gains are designated by user now. If these data can be acquired from real world through the physical nodes, the accuracy for H-TOSSIM to estimate the power consumption for large WSN can be improved.

The second consideration is to use the physical nodes to provide sensing data for the virtual nodes. Sensing results are produced randomly in the current version of H-TOSSIM. If the virtual nodes can get sensing data from the real world through the physical nodes, the software testing can be more sufficient and more practical.

7. References

[1] I. Akyildiz, W. Su, *et al*., "Wireless sensor networks: A survey," Computer Networks, Vol. 38, No. 4, pp. 393–422, March 2002.

[2] P. Levis, N. Lee, *et al*., "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Los Angeles, CA, pp. 126–137, November 2003.

[3] P. Levis and N. Lee, "TOSSIM: A simulator for TinyOS networks, October 2007, http://www.cs.berkeley.edu/~pal/pubs/nido.pdf.

[4] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in Proceedings of the Sixth International Conference on Information Processing in Sensor Networks, Cambridge, Massachusetts, pp. 21–30, April 2007.

[5] TinyOS. http://www.tinyos.net/.

[6] J. Hill, R. Szewczyk, *et al*., "System architecture directions for networked sensors," ACM SIGPLAN Notices, Vol. 35, No. 11, pp. 93–104, 2000.

[7] D. Gay, P. Levis, *et al*., "The nesC language: A holistic approach to networked embedded systems," ACM SIGPLAN Notices, Vol. 38, No. 5, pp. 1–11, May 2003.

[8] V. Shnayder, M. Hempstead, *et al*., "Simulating the power consumption of large-scale sensor network applications," in Proceedings of the Second ACM Conference on Embedded Networked Systems, Baltimore, MD, pp. 188–200, November 2004.

[9] The Network Simulator: ns–2. http://www.isi.edu/nsnam/ns.

[10] S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: A simulation framework for sensor networks," in Proceedings of the Third ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Boston, Massachusetts, pp. 104–111, August 2000.

[11] L. Girod, T. Stathopoulos, *et al*., "A system for Simulation, emulation, and deployment of heterogeneous sensor networks," in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems, Baltimore, MD, pp. 201–213, November 2004.

[12] J. Pollet, D. Blazakis, *et al*., "ATEMU: A fine-grained sensor network simulator," in Proceedings of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, pp. 145–152, October 2004.

[13] B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: Scalable sensor network simulation with precise timing," in Proceedings of the Fourth International Conference on Information Processing in Sensor Networks, Los Angeles, CA, pp. 477–482, April 2005.

[14] D. Jia, G. H. Krogh, and C. Wong, "TOSHILT: Middleware for hardware-in-the-Loop testing of wireless sensor networks," October 2007. http://www.ece.cmu.edu/~webk/sensor_networks/pub/ipsn05_hilt.pdf.

[15] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A wireless sensor network testbed," in Proceedings of the Fourth International Conference on Information Processing in Sensor Networks, Los Angeles, CA, pp. 483–488, April 2005.

[16] M. Dyer, J. Beutel, *et al*., "Deployment support network: A toolkit for the development of WSNs," in Proceedings of the Fourth European Workshop on Sensor Networks, Berlin, pp. 195–211, January 2007.

[17] L. Girod, J. Elson *et al*., "EmStar: A software environment for developing and deploying wireless sensor networks," in Proceedings of the USENIX Technical Conference, San Diego, CA, pp. 24–37, June 2004.

[18] J. Hill, M. Horton, *et al*., "The platforms enabling wireless sensor networks," Communications of the ACM, Vol. 47, No. 6, pp. 41–46, June 2004.

[19] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in Proceedings of the Fourth International Conference on Information Processing in Sensor Networks, Los Angeles, CA, pp. 364–369, April 2005.

Wireless Sensor Network, **2009, 1, 334-339**
doi:10.4236/wsn.2009.14041 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Distributed Video Coding Using LDPC Codes for Wireless Video

P. APARNA, Sivaprakash REDDY, Sumam DAVID*
Department of Electronics and Communication Engineering,
National Institute of Technology Karnataka, Surathkal, India
Email: *sumam@ieee.org*
Received May 19, 2009; *revised June* 16, 2009; *accepted June* 23, 2009

Abstract

Popular video coding standards like H.264 and MPEG working on the principle of motion-compensated predictive coding demand much of the computational resources at the encoder increasing its complexity. Such bulky encoders are not suitable for applications like wireless low power surveillance, multimedia sensor networks, wireless PC cameras, mobile camera phones etc. New video coding scheme based on the principle of distributed source coding is looked upon in this paper. This scheme supports a low complexity encoder, at the same time trying to achieve the rate distortion performance of conventional video codecs. Current implementation uses LDPC codes for syndrome coding.

Keywords: Syndrome Coding, Cosets, Distributed Source Coding, Distributed Video Coding (DVC).

1. Introduction

With the proliferation of various complex video applications it is necessary to have advanced video and image compression techniques. Popular video standards like ISO MPEG and ITU-H.26x have been successful in accomplishing the requirements in terms of compression efficiency and quality. However these standards are pertinent to downlink friendly applications like video telephony, video streaming, broadcasting etc. These conventional video codecs work on the principle of motion compensated prediction which increases the encoder complexity due to the coexistence of the decoder with the encoder. Also motion-search algorithm makes the encoder computationally intensive. The downlink friendly architectures belong to the class of *Broadcast* model, where in high encoder complexity is not an issue. The encoder of a *Broadcast* model resides at the base-station where power consumption and computational resources are not an issue. However this *Broadcast* model of video is not suitable for uplink friendly applications like mobile video cameras, wireless video sensor networks, wireless surveillance etc which demands a low power, low complexity encoder. These uplink friendly applications which belong to *wireless-video* model demands a simple encoder since the power and the computational resources are of primary concern in the wireless scenario. Based on the information theoretic bounds established in 1970's by Slepian-Wolf [1] for distributed lossless coding and by Wyner-Ziv [2] for lossy coding with decoder side information, it is seen that efficient compression can also be achieved by exploiting source statistics partially or wholly at the decoder. Video compression schemes that build upon these theorems are referred as distributed video coding which befits uplink friendly video applications. Distributed video coding shifts the encoder complexity to the decoder making it suitable for wireless video model. Unlike conventional video codecs distributed coding exploits the source statistics at the decoder alone, thus interchanging the traditional balance of complex encoder and simple decoder. Hence the encoder of such a video codec is very simple, at the expense of a more complex decoder. Such algorithms hold great promise for new generation mobile video cameras and wireless sensor networks. In the design of a new video coding paradigm, issues like compression efficiency, robustness to packet losses, encoder complexity are of prime importance in comparison with conventional coding system. In this paper we present the simulation results of distributed video coding with syndrome coding as in PRISM [3], using LDPC codes for coset channel coding [4].

2. Background

2.1. Slepian-Wolf Theorem for Lossless Distributed Coding [1]

Consider two correlated information sequences X and Y.

*SMIEEE.

Encoder of each source is constrained to operate without the knowledge of the other source while the decoder has access to both encoded binary message streams as shown in Figure 1. The problem that Slepian-Wolf theorem addresses is to determine the minimum number of bits per source character required for encoding the message stream in order to ensure accurate reconstruction at the decoder. Considering separate encoder and the decoder for X and Y, the rate required is $R_X \geq H(X)$ and $R_Y \geq H(Y)$ where $H(X)$ and $H(Y)$ represents the entropy of X and Y respectively. Slepian-Wolf [1] showed that good compression can be achieved with joint decoding but separate encoding.

For doing this an admissible rate region is defined [6] as shown in Figure 2 given by:

$$R_X + R_Y \geq H(X,Y) \quad (1)$$

$$R_X \geq H(X/Y),\ R_Y \geq H(Y) \quad (2)$$

$$R_X \geq H(X),\ R_Y \geq H(Y/X) \quad (3)$$

Thus Slepian-Wolf [1] showed that Equation (1) is the necessary condition and Equation (2) or Equation (3) are the sufficient conditions required to encode the data in case of joint decoding. With the above result as the base, we can consider the distributed coding with side information at the decoder as shown in the Figure 3. Let X be the source data that is statistically dependent to the side information Y. Side information Y is separately encoded at a rate $R_Y \geq H(Y)$ and is available only at the decoder. Thus as seen from Figure 2 X can be encoded at a rate $R_X \geq H(X/Y)$.

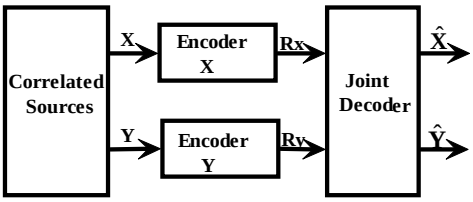


Figure 1. Compression of correlated sources by separate encoder but decoded jointly.

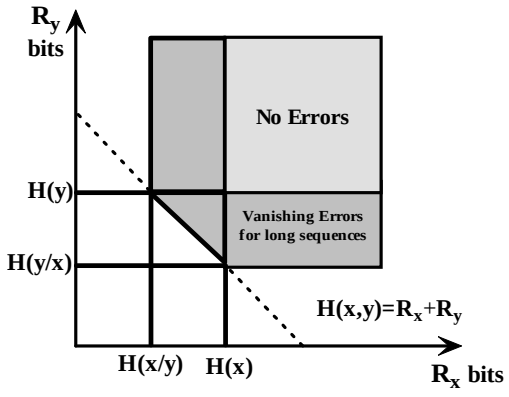


Figure 2. Admissible rate region [5].

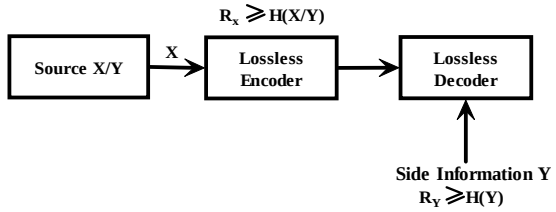


Figure 3. Lossless decoder with side information.

2.1. Wyner-Ziv Rate Distortion Theory[2,6]

Aaron Wyner and Jacob Ziv [2,6] extended Slepian-Wolf theorem and showed that conditional Rate-MSE distortion function for X is same whether the side information is available only at the decoder or both at encoder and decoder; where X and Y are statistically dependent Gaussian random processes. Let X and Y be the samples of two random sequences representing the source data and side information respectively. Encoder encodes X without access to side information Y as shown in Figure 4.

Decoder reconstructs $\hat{X}$ using Y as side information. Let D = E [d ($\hat{X}$, X)] is the acceptable distortion. Let $R_{X/Y}(D)$ be the rate required for the case where side information is available at the encoder also and $R_{X/Y}^{WZ}(D)$ represent the Wyner-Ziv rate required when encoder doesn't have access to side information. Wyner-Ziv proved that Wyner-Ziv rate distortion function $R_{X/Y}^{WZ}(D)$ is the achievable lower bound for the bitrate for a distortion D

$$R_{X/Y}^{WZ}(D) - R_{X/Y}(D) \geq 0 \quad (4)$$

They also showed that for Gaussian memoryless sources

$$R_{X/Y}^{WZ}(D) - R_{X/Y}(D) = 0 \quad (5)$$

As a result source sequence X can be considered as the sum of arbitrarily distributed side information Y and independent Gaussian Noise.

Distributed video coding is based on these two fundamental theories, specifically works on the Wyner-Ziv coding considering a distortion measure. In such a coding system the encoder encodes each video frame separately

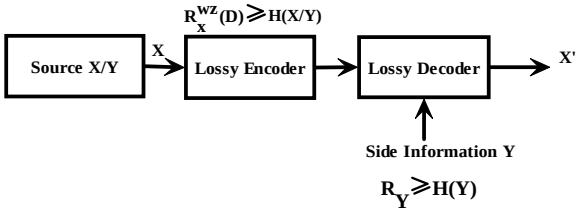


Figure 4. Lossy decoder with side information.

with respect to the correlation statistics between itself and the side information. The decoder decodes the frames jointly using the side information available only at the decoder. This video paradigm is as opposed to the conventional coding system where the side information is available both at the encoder and decoder as shown in Figure 5.

2.2. Syndrome Coding [5]

Let X be a source that is to be transmitted using least average number of bits. Statistically dependent side information *Y*, such that $X = Y + N$ is available only at the decoder. The encoder must therefore encode *X* in the absence of *Y*, whereas the decoder jointly decodes *X* using *Y*. Distributed source encoder compresses *X* in to syndromes S with respect to a Channel code *C* [7]. Decoder on receiving the syndrome can identify the coset to which *X* belongs and using side information *Y* can reconstruct back *X*.

2.3. Correlation Channel and the Channel Codes [4]

The performance of the channel codes is the key factor of the distributed video coding system in both error correcting and data compression. Turbo and LDPC codes are two advanced channel codes which have astonishing performance near the Shannon Capacity limit. The use of LDPC codes for syndrome coding was first suggested by Liveris in [4], where the message passing algorithm was modified to take syndrome information in to account.

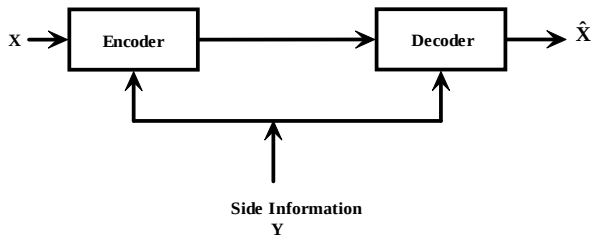


Figure 5. Lossless decoder with side information.

The correlation between binary sources $X = [X_1, X_2.....,X_n]$ and $Y = [Y_1, Y_2, ..., Y_n]$ is modeled using a binary symmetric channel. We consider X_i and Y_i to be correlated according to $Pr\ [X_i \neq Y_i] = p < 0.5$. The rate used for *Y* is its entropy $R_Y = H(Y)$, therefore the theoretical limit for lossless compression of *X* is given by

$$nR_x \geq nH(X_i/Y_i) = nH(p) = n(-plog_2p-(1-p)log_2(1-p)) \quad (6)$$

The compressed version of *X* is the syndrome *S* which is the input to the channel. The source *Y* is assumed to be available at the decoder as side information. Using a linear (n,k) binary block code, it is possible to have 2^{n-k} distinct syndromes, each indexing a set of 2^k binary words of length *n*. This compression results in mapping a sequence of n input symbols into $(n-k)$ syndrome symbols.

3. Implementation

3.1. Encoder

The encoder block diagram is shown in the Figure 6. The video frames are divided into blocks of 8x8 and each block is processed one by one. Block DCT (Discrete Cosine Transform) is applied to each 8x8 block (or 16x16) and the DCT coefficients are zig-zag scanned so that they are arranged as an array of coefficients in order of their importance. Then the transformed coefficients are uniform quantized with reference to target distortion measure and desired reconstruction quality. After quantization a bitplane is formed for each block as shown in Figure 7 [3]. Main idea behind distributed video coding is to code source *X* assuming that the side information *Y* is available at the decoder such that $X = Y + N$, where *N* is Gaussian random noise. This is done in the classification step where bitplane for each coefficient is divided into different levels of importance. Classification step strongly rely on the correlation noise

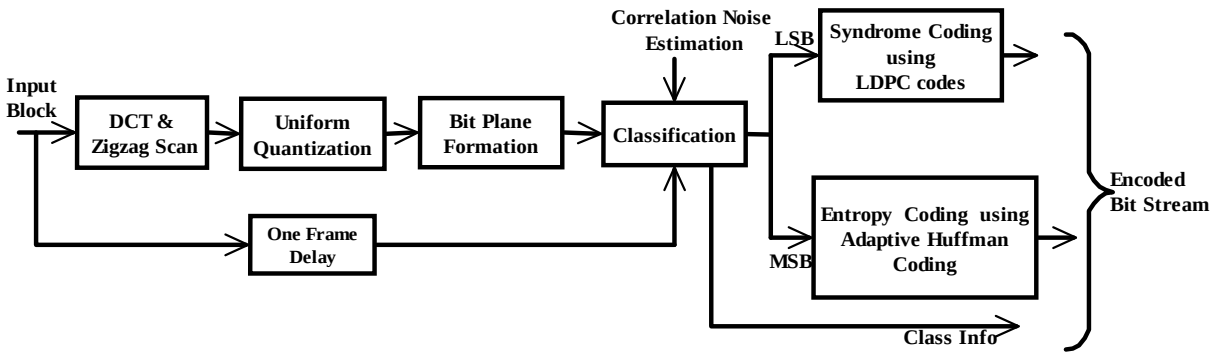


Figure 6. Video encoder.

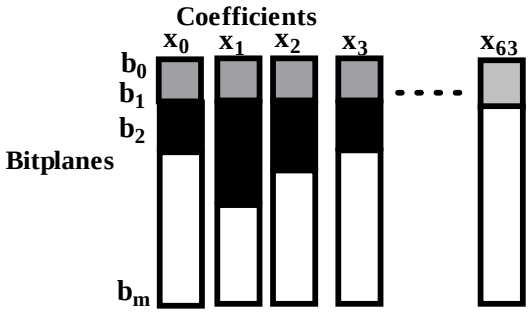


Figure 7. Bit planes for each coefficient blocks.

structure *N* between the source block *X* and the side information block *Y*. Less is the correlation noise between *X* and *Y*, more is the similarity and hence less number of bits of *X* can be transmitted to the decoder. In order to classify the bitplanes offline training is done for different types of video files without any motion search. On the basis of offline process 16 types of classes are formed, where each class considers different number of bitplanes for entropy coding and syndrome coding for each coefficient in the block. In the classification process, MSE (mean square error) for each block is computed with respect to the zero motion blocks in the previous frame. Based on the MSE and the offline process appropriate class for that particular block is chosen. As a result some of the least significant bit planes are syndrome coded and some of the bitplanes that can be reconstructed from side information are totally ignored. The syndrome coding bitplanes shown in black and gray in Figure 7 and skip planes shown in white in Figure 7. Skip planes can be reconstructed back using side information at the decoder and hence need not be sent to the decoder. The important bits of each coefficient that cannot be determined by side information has to be syndrome coded [3]. In our implementation we code two bitplanes using coset channel coding and the remaining syndrome bitplanes using Adaptive Huffman coding. Among the syndrome coding bitplanes we code the most significant bit planes using Adaptive Huffman coding. The number of bitplanes to be syndrome coded is directly used from class information that is hard coded. Hence we need not send four-tuple data (run, depth, path, last) as in PRISM [3]. Rest of the least significant bitplanes is coded using coset channel coding. This is done by using a parity check matrix *H* of a (*n*,*k*) linear channel code. Compression is achieved by generating syndrome bits of length (*n*-*k*) for each n bits of data. These syndrome bits are obtained by multiplying the source bits with the parity check matrix *H* such that

$$S = Hb_X$$

where *S* represents the syndrome bits. *H* represents the parity check matrix of linear channel code. b_X represents the source bits.

These syndromes identify the coset to which the source data belongs to. In this implementation we have considered two biplanes for coset coding marked gray in the Figure 7. We have implemented this using irregular 3/4 rate LDPC coder [4].

3.2. Decoder

The Decoder block diagram is shown in the Figure 8. The entropy coded bits are decoded by an entropy decoder and the coset coded bits are passed to the LDPC decoder. In this implementation, previous frame is considered as the side information required for syndrome decoding. Once the syndrome coded bits are recovered they identify the coset to which X_i belongs and hence using the side information Y_i we can correctly decode the entire bits of X_i. The quantized codeword sequence is then dequantized and inverse transformed to get the original coefficients.

4. Simulation Results

Video Codec is designed for a single camera scenario which is an application to wireless network of video camera equipped with cell phones. The video codec is simulated and tested with a object oriented approach

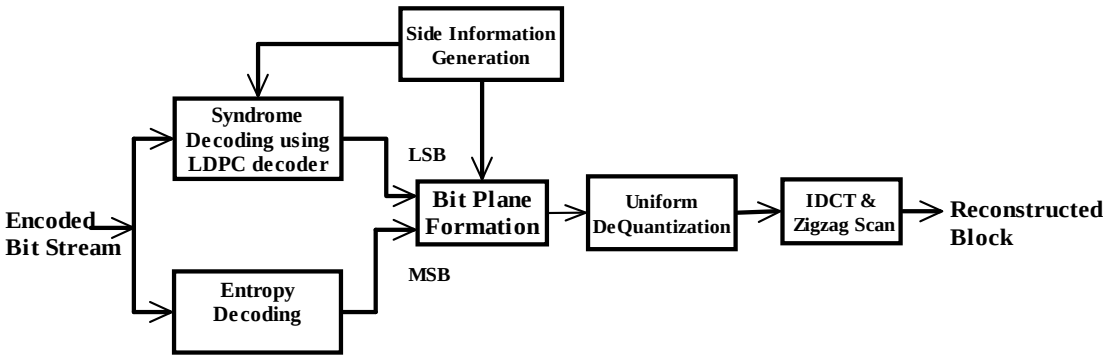


Figure 8. Video decoder.

Table 1. Filename: foreman. QCIF, frame rate=30fps.

BitRate (Mbps)	Luma PSNR (dB) for different Methods		
	DVC Implementation	H.263+Predictive Coder	IntraCoder (Motion JPEG)
2.57	31.357	34.72	30.092
2.67	33.554	35.03	32.863
3.55	35.534	35.86	34.92

Table 2. Filename: football. QCIF, frame rate=30fps.

BitRate (Mbps)	Luma PSNR (dB) for different Methods		
	DVC Implementation	H.263+ Predictive Coder	IntraCoder (Motion JPEG)
3.52	30.724	25.62	30.07
3.67	31.834	25.76	30.92
4.87	34.005	26.59	33.80

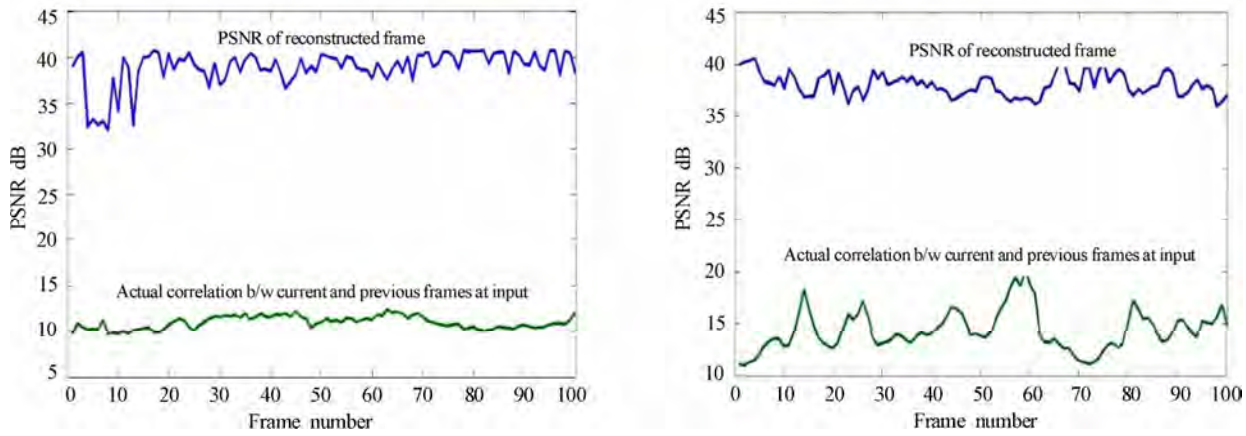


Figure 9. a) Error resilience characteristics of DVC, 4th, 10th, 20th frames are lost for football; b) Error resilience characteristics of DVC, 4th, 10th, 20th frames are lost for foreman.

using C++ in gcc. The program processes frames one by one and within each frame, block wise processing is done. The input to the encoder is a QCIF video file (Quarter Common Intermediate Format). Encoder allows the storage of one previous frame. Objective performance evaluation of the system is done by measuring the Compression Ratio (CR), MSE and the Peak Signal to Noise Ratio (PSNR) between the original and the reconstructed video. The PSNR and CR for various video sequences is computed. These are compared with that of H.263+ Intra and H.263+ Predictive video codec [8]. The encoder and decoder block as shown in Figure 6 and Figure 8 respectively are implemented and some preliminary simulation results are presented in this paper for two video files *Football* and *Foreman* in *QCIF resolution* with a frame rate of 30 *fps*. The rate distortion performance and the error resilience characteristics of the distributed video coder is presented in this paper. As seen from the Table 1, for the same bitrate distributed video coder has better PSNR than DCT based intraframe coder and but is slightly inferior to H.263+ predictive coder [8] for *Foreman* file. As seen from Table 2 distributed video coder has better PSNR than DCT based intraframe coder and H.263+ predictive coder for *Football* file. With some enhancements to the current coding scheme such as accurate modeling of correlation statistics between the source data and the side information, proper motion search module for side information generation etc, better rate-distortion performance can be achieved with a low complexity encoder model.

Error Resilience characteristics of Distributed video scheme is as shown in Figure 9a for *Football* and Figure 9b for *Foreman*. Effect on the quality of the reconstructed video sequence is seen by dropping 4th, 10th, 20th frames at the decoder in our implementation. It is seen that distributed video coder recovers quickly. In Distributed video scheme, decoding is dependent on the side information Y that is universal for all source data X as long as correlation structure is satisfied.

5. Conclusion

In this paper we have tried PRISM [3] like implementation using LDPC coset channel coding. By proper modeling of correlation structure of source and the side information for video we can achieve better compression performance with better quality of reconstructed video sequence. However the main aim of distributed video coding scheme is to reduce encoder complexity to conform with *wireless-video* model, which seems to be satisfied. Distributed codec is more robust to packet /frame

loss due to the absence of prediction loop in the encoder. In a Predictive coder accuracy of decoding is strongly dependent on a single predictor from the encoder, loss of which results in erroneous decoding and error propagation. Hence Predictive coder can recover from packet or frame loss by only some extent. The quality of the reconstructed signal for the same CR can be improved by performing more complex motion search. However it is seen that the current implementation operates well in high quality (PSNR of order of 30dB) regime. The extension to lower bit rates without any compromise in the quality so that it is comparable with the conventional codecs will be the next part of the work.

6. References

[1] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," IEEE Transactions on Information Theory, Vol. IT-19, pp. 471–480, July 1973.

[2] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," IEEE Transactions on Information Theory, Vol. IT-22, No. 1, pp. 1–10, January 1976.

[3] R. Puri, A. Majumdar, and K. Ramachandran, "PRISM: A video coding paradigm with motion estimation at the decoder," IEEE Transactions on Image Processing, Vol. 16, No. 10, October 2007.

[4] A. D. Liveris, "Compression of binary sources with side information at the decoder using LDPC codes," IEEE Communication Letters, Vol. 6, No. 10, October 2002.

[5] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," Proc. IEEE Data Compression Conference, Snowbird, UT, pp. 158–167, March 1999.

[6] A. D. Wyner, "Recent results in the Shannon theory," IEEE Transactions on Information Theory, Vol. 20, No. 1, pp. 2–10, January 1974.

[7] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," Proc. Allerton Conference on Communication, Control and Computing, Allerton, IL, October 2002.

[8] G. Cote, B. Erol, M. Gallant, and F. Kosssentini, "H.263+: Video coding at low bitrates," IEEE Transactions. Circuits Sys. Video Technology, Vol. 8, No. 7, pp. 849–866, November 1998.

[9] B. Girod, A. M. Aaron, S. R. and D. Rebollo-Monedero, "Distributed video coding," Proceedings of the IEEE, Vol. 93, No. 1, pp. 71–83, January 2005.

Wireless Sensor Network, **2009, 1, 340-349**
doi:10.4236/wsn.2009.14042 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Dynamic Hierarchical Communication Paradigm for Wireless Sensor Networks: A Centralized, Energy Efficient Approach

Suraiya TARANNUM[1], S. Srividya[2], D. S. Asha[2], K. R. Venugopal[2]
[1]*Department of Telecommunication Engineering, AMC Engineering College, Bangalore, India*
[2]*Department of Computer Science and Engineering, Bangalore University, Bangalore, India*
Email: *ssuraiya@gmail.com*

Abstract

A Wireless Sensor Network (WSN) consists of a large number of randomly deployed sensor nodes. These sensor nodes organize themselves into a cooperative network and perform the three basic functions of sensing, computations and communications. Research in WSNs has become an extensive explorative area during the last few years, especially due the challenges offered, energy constraints of the sensors being one of them. In this paper, the need for effective utilization of limited power resources is emphasized, which becomes pre-eminent to the Wireless Sensor Networks. Organizing the network to achieve balanced clusters based on assigning equal number of sensors to each cluster may have the consequence of unbalanced load on the cluster heads. This results in an unbalanced consumption of energy by the nodes, cumulatively leading to minimization of network lifetime. In this paper, we put forth a Sink administered Load balanced Dynamic Hierarchical Protocol (SLDHP) to balance the load on the principal nodes. Hierarchical layout of the sensors endows the network with considerable minimization of energy consumption of nodes leading to an increased lifespan. Simulation results indicate significant improvement of performance over Base station Controlled Dynamic Clustering Protocol (BCDCP).

Keywords: Wireless Sensor Network, Sink, Principal Node, Superior Node, Network Lifetime

1. Introduction

A Wireless Sensor Network (WSN) is an ad-hoc wireless telecommunication network which embodies a number of tiny, low-powered sensor nodes densely deployed either inside a phenomenon or close to it [1]. The multifunctioning sensor nodes operate in an unattended environment with limited sensing and computational capabilities. The advent of wireless sensor networks has marked a remarkable change in the field of information sensing and detection. It is a conjunction of sensor, distributed information processing, embedded and communication techniques. WSNs may in the near future be equally prominent by providing information of the physical phenomena of interest and ultimately being able to detect and control them or enable us to construct more meticulous models of the physical world.

WSNs are easier, faster and cheaper to deploy than other forms of wireless networks as there are no predetermined positions for the sensors. They have higher degree of faulttolerance than other wireless networks and are self-configuring or self-organizing [2]. Sensors are deployed randomly and are expected to perform their mission properly and efficiently. Another unique feature of sensor networks is the co-operative effort of sensor nodes to achieve a particular task.

A WSN is envisioned to consist of a large number of sensors and many base stations. The sensors are equipped with transceivers to gather information from the environment and pass it on to one of the base stations. A typical sensor node consists of four major components: a data processor unit; a sensor; a radio communication subsystem that consists of transmitter/receiver electronics, antennas, an amplifier; and a power supply unit [3]. The sensors are compact in size which make them extremely energy restrained. Further more, replacing batteries in large scales in possibly harsh terrain becomes infeasible. Hence, it is well accepted that the key challenge in unlocking the potential of such networks is maximizing their post-deployment active lifetime. The lifetime of the wireless sensors may be prolonged by ensuring that all aspects of the system achieve energy efficiency. Since communications in wireless sensor networks consume significant amount of energy, the designed algorithms must ensure that nodes expend minimum amount of energy for transmitting and receiving data.

A web of sensor nodes can be deployed to gather productive information from the sensor field. Some of the benefits of using WSNs are extended range of sensing, fault-tolerance, improved accuracy and lower cost. As a consequence, the sensor networks are expected to find extensive use in a variety of applications including remote climate monitoring, seismic, acoustic, medical and intelligence data-gathering [4,5]. Hence, they are suitable for a wide range of applications like military, health, education, commerce and so on. Military applications may range from tracking enemy movement in the battlefield to guiding targeting system. Bio-sensors are used for monitoring patients blood sugar level. Commercial applications may range from tracking postal packages or office equipment to monitoring product quality on an assembly line. Environmental applications include forest-fire detection, flood detection, tracking movements of birds etc. Sensors are also used to simulate home automation and to build smart environments.

Efficient utilization of energy is crucial to the WSNs. Wireless microsensor network protocols should therefore be selfconfiguring to enable ease of deployment of nodes, latency aware, qualitative, robust and to extend system lifetime. The sensors are extremely energy bounded, hence the network formed by these sensors are also energy constrained. The communication devices on these sensors are small and have limited power and sensing ranges. A routing protocol coordinates the activities of individual nodes in the network to achieve global goals and does it in an efficient manner. The simplest is the Direct Communication Routing Protocol, where each node transmits the sensed information directly to the base station. The nodes consume considerable amount of energy if the communication path is long. This results in the early death of distant nodes. To overcome this drawback, the technique of Minimum Transmission Energy utilizes a multhop routing scheme. In this scheme, the nodes that are close to the base station drain their energy rapidly as they are involved in transmission of messages on behalf of others.

Hierarchical routing groups sensors in the entire network into clusters. It aims at reduction of energy consumption by localizing data communication within a cluster and aggregating data to decrease transmissions to the base station. The first attempt in this regard, was made by Low Energy Adaptive Cluster Hierarchy (LEACH). The operation is framed in iterations and each iteration comprises of a setup and a data transmission phase. During the setup phase, nodes organize themselves into clusters with predetermined number of nodes serving as cluster heads. In the data transmission phase, the self-elected cluster heads aggregate data received from the nodes in their cluster before forwarding to the base station. The role of cluster heads is randomly rotated among all the nodes in the network. This technique serves as a basic model for other hierarchical routing protocols. A centralized version of the adaptive approach comprises a hierarchical structure in which the base station has control over the cluster formation. The base station uses the location and energy information sent by the nodes to select the predetermined number of cluster heads. Efficient clustering is achieved as the base station possess the global knowledge of the network. This technique exhibits improvement over the adaptive approach.

In Power Efficient Gathering in Sensor Information Systems (PEGASIS), the nodes function co-operatively to optimize network lifetime. A greedy algorithm is used to configure the network into chains. In each iteration, a randomly chosen leader node directs the aggregated data to the base station. A centralized energy efficient routing protocol called Base Station Controlled Dynamic Clustering Protocol (BCDCP), was proposed which widened the area for research in hierarchical routing. Here, much of the functionalities like formation of clusters and routing paths are performed by the high energy base station which lightens the load of sensor nodes. This protocol configures the network into balanced clusters where each cluster head serves an approximately equal number of member nodes. Cluster head-to-cluster head multihop routing is employed in this protocol to transfer the data to the base station.

Motivation: Efficient management of energy deserves much of the attention in the WSNs. Routing protocols designed for WSNs must therefore effectively tackle this issue in order to enhance the lifetime of the network. Hierarchical routing techniques are preferable in this direction. The arrangement of the nodes in the form of a load balanced hierarchy proves to be beneficial.

Contribution: In our paper, we propose a energy efficient hierarchical routing protocol, SLDHP to increase the lifetime of homogeneous as well as heterogeneous WSNs. SLDHP achieves a load balanced hierarchical arrangement of nodes in the network which performs better than the other hierarchical routing protocols.

Organization: The rest of the paper is organized as follows. In section II, we discuss the related work. In section III, the underlying model is described and the problem is defined in section IV. Our proposed algorithm, SLDHP is presented in section V. Performance analysis is presented in section VI and section VII contains the conclusion.

2. Related Work

Hierarchical routing aims to efficiently maintain the energy consumption of sensor nodes by involving them in multihop communication within a particular cluster and

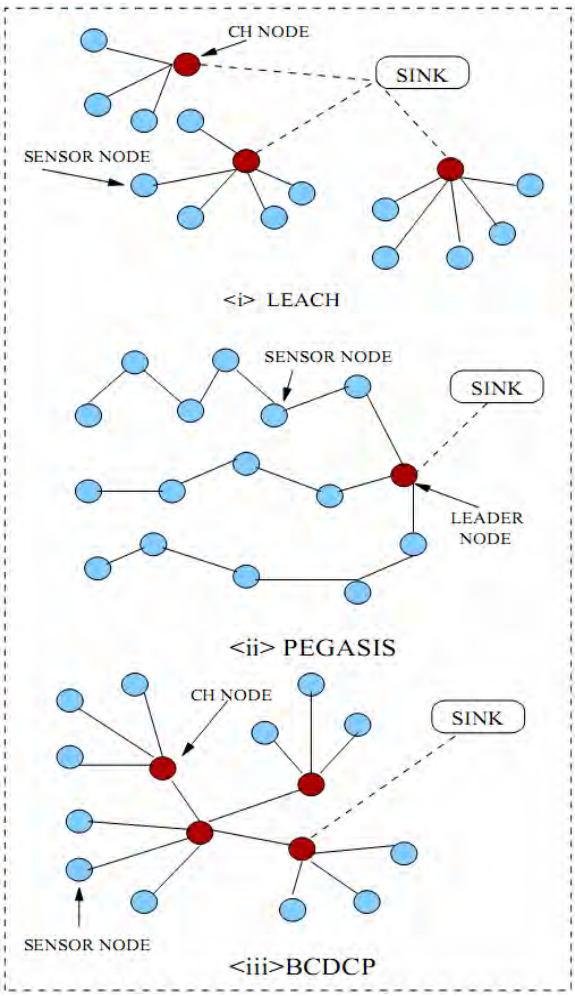


Figure 1. Three main topologies of hierarchical routing protocols.

by performing data aggregation and fusion to decrease the number of transmitted messages to the base station.

Heinzelman *et al.* [6] proposed an adaptive clustering protocol. This approach employs the technique of randomly changing the role of cluster head among all nodes in the sensor network. The operation of this protocol is organized into different iterations where each iteration consists of a setup phase and a transmission phase. During setup phase, nodes organize themselves into clusters in which cluster head is elected locally within each cluster. During transmission phase, the self elected cluster heads aggregate data received from all the nodes within its cluster, applies a data fusion technique before sending it directly to the base station. In this method, the decision is made per iteration and it is assumed that we have a knowledge of the total residual energy of the network.

In [7], a centralized algorithm for routing is described. This protocol uses the base station for centralized computation of cluster heads. The base station upon receiving the location and energy level information from the sensor nodes during the setup phase, locates a predetermined number of cluster heads and configures the entire network into clusters. The cluster heads are chosen in such a way that nodes consume minimum energy for transmitting their data. The shortcoming may be that they drain their energy rapidly as they have to communicate directly with the base station irrespective of their positions. The results in [7] show improvement over [6].

A chain based protocol is presented in [8]. In this protocol, each node communicates only with a close neighbour and takes turns to transmit to the base station, thus reducing the amount of energy spent per iteration. For constructing a chain, it is assumed that all nodes have global knowledge of network. A greedy algorithm is employed to ensure that nodes already on the chain need not be revisited. Here, even though the forwarding node has capability of taking more load, it is not assigned if it is already on the chain.

A centralized clustering based routing protocol is discussed in [9]. According to this protocol, energy intensive tasks such as cluster setup, cluster head selection, routing path formation and TDMA schedule creation are performed by the base station which is assumed to have unlimited power supply. This protocol configures the network into balanced clusters, i.e., the number of nodes in each cluster are same. Such equal clustering results in an unequal load on the cluster head.

Guangyan *et al.* [10] have reviewed the energy efficiency of cluster based routing protocols with extended conditions of general complexity of data fusion algorithm, general data compressing ratio and long distances. They present three discoveries, first of which is that data fusion algorithm is computed based on applications. Secondly, multihop scheme not used by earlier works could sometimes prove beneficial. Thirdly, when network area is larger than 200mx200m, the number of high energy dissipating nodes are more which accelerates the death of nodes. These findings guide in improving the routing protocols and hence to extend their application ranges.

Geographic and energy aware routing algorithm developed by Yan Yu *et al.* [11], propagates a query to the appropriate geographical region without flooding. The protocol uses energy aware and geographically informed neighbor selection to route a packet towards the target region. To disseminate the packet inside the destination region, a recursive geographic forwarding or restricted flooding algorithm is used. The protocol exhibits noticeably longer network lifetime as compared to non-energy aware geographic routing algorithms.

A novel algorithm proposed by Andrea in [12], performs three main functions of configuring the network into optimum number of clusters, decentralized cluster head selection and cluster formation. The value of optimum number of clusters depends on total number of sensors in the network, on the path-loss exponent (α), on

dimensions of the network and distance of the broadcast packets. They use an adaptive strategy for cluster head selection. The algorithm for cluster formation uses total path energy dissipation instead of energy lost in path from the node to its cluster head. The algorithm optimizes system lifetime in a large range of applications and situations.

A cost based comparison of homogeneous and heterogeneous clustered sensor networks has been presented. It first considers single hop clustered sensor networks and use adaptive clustering protocol. It also takes into account sensor-network with two types of nodes as representative single hop heterogeneous networks. For multihop homogeneous network Vivek *et al*. [13] propose and analyze a multihop variant of the adaptive approach. They consider communication radius for in-cluster communication and size of clusters. This algorithm exhibits better energy efficiency in many cases, but does not give expected performance if the heterogeneity is due to the operation of the network.

An energy efficient distributed clustering approach for adhoc sensor network is developed in paper [14]. In this approach, cluster heads are chosen randomly based on their residual energy and nodes participate in cluster operation such that the communication cost is minimized. The protocol does not make any assumptions regarding the distribution density of nodes. The clustering process takes a fixed number of iterations and does not depend on network topology. This protocol acheives only a two-level hierarchy.

Alan *et al*. [15] have derived a load balancing heuristic for wireless adhoc networks in order to extend the lifespan of a cluster head to as large an extent as possible before another node becomes the cluster head. Two cluster head load balancing heuristics are described. The first approach is for cluster election heuristics that favour the election of cluster head based on *node-id*, and the second approach is based on the degree of connectivity.

In [16], a cluster based query protocol is illustrated for wireless sensor networks using self-organized sensor clusters to register queries, process queries and disseminate data within the network. This protocol uses cluster heads as data storage and aggregation points. Instead of sending large amounts of raw data over a network to reply to a query, each cluster head collects and filters data from its member sensors. This is achieved using the information about the cluster location. With this protocol, energy efficiency is achieved by reducing the number of data transmissions over the network during the course of the data collection and query processing.

A stable election protocol is described in [17] which is a heterogeneous-energy-aware protocol. It is based on weighted election probabilities of each node to become cluster head based on the remaining energy of each node. In this approach every sensor node in a heterogeneous two-level hierarchical network independently elects itself as a cluster head based on its initial energy relative to that of other nodes. The protocol does not demand any global knowledge of energy at every election iteration and also does not consider as to how nodes could be assigned optimally to cluster heads.

In [18], a balanced *k*-clustering algorithm, for clustering sensor nodes into *k* number of clusters is described. Each cluster is balanced and the total distance between sensor nodes and the head nodes is minimized. Minimizing the total distance helps in reducing the communication overhead and hence energy dissipation. The algorithm demonstrates that the balanced *k*-clustering problem can be solved optimally using network flow, but assumes the number of nodes as a multiple of *k* at all times, which may not be practical.

A cluster based routing algorithm is depicted in [19] to extend the lifetime of the sensor networks and hence to maintain a uniform consumption of the energy by the nodes. This is obtained by the addition of a slot in a frame, which enables the exchange of residual energy messages between the base station, cluster heads and nodes. The algorithm takes into account the residual energy of the nodes during cluster head selection, resulting in balanced energy consumption of the sensor nodes. The protocol performs better than the adaptive approach.

In [2], the authors focus on the design criteria for routing protocols and issues and challenges of cluster-based routing in WSNs. The characteristics and the general routing models for protocols in sensor networks are studied here. Yunfeng et al. [20] have devised a protocol called energy balancing multipath routing, the basic idea being that instead of source-initiated or destination-initiated route discovery, it is the base station that finds multiple paths to the source of the data and selects one of them to be used during communication. It is based on the assumption that the base stations are typically many orders of magnitude more powerful than common sensor nodes. It adopts a scheme similar to the well known software architecture client server model.

Energy aware routing that uses sub-optimal paths occasionally to provide substantial gains is designed by Rahul et al. [22]. It emphasizes that using lowest energy paths may not be always optimal from the point of view of network lifetime and long-term connectivity. The protocol is suitable for low energy and low bitrate networks. The key concept is to send traffic through different routes which helps in using the node resources more evenly. It sends the traffic on multiple paths without adding much complexity by using a probabilistic forwarding technique. According to this method, the nodes burn their energy uniformly across the network ensuring a more graceful degradation of service with time.

The problem of energy-aware routing in networks with renewable energy sources is adressed by Longbi *et al*. [21]. They present a simple, static multi-path routing approach that is optimal in the large system limit. The

proposed static routing scheme utilizes the knowledge on the traffic patterns and energy consumption, and does not demand the instantaneous information about the node energy. For the distributed computation of the optimal policy, they outline the possible approaches and propose heuristics to build the set of pre-computed paths. This scheme outperforms leading dynamic routing algorithms, and is close to an optimal solution when the energy claimed by each packet is relatively small compared to the battery capacity.

3. Model

3.1. The Nomenclature

The terminology used in our study are,

HmNt Homogeneous Network consists of sensors possessing a uniform initial energy.

HtNt Heterogeneous Network comprises of sensors with different initial energies.

$\mathbb{N}$ Set of all the sensor nodes deployed in the sensor field of the network.

E_{avg} This is defined as the average energy of the wireless sensor network.

$$E_{avg} = \frac{1}{n}\sum_{k=1}^{n} E_k \tag{1}$$

where n is the number of the sensors and E_k is theenergy of the k^{th} sensor.

$\mathbb{P}$ Set consisting of sensor nodes with energy equal to or greater than E_{avg}, and is a subset of set $\mathbb{N}$, which is a set of all the sensor nodes deployed in the network.

PrNd Principal Node is a node which receives the sensed data from other nodes in its hierarchy, aggregates it to forward either to another principal node or to the *Superior Node*. This functions as the root of the hierarchy and sends the aggregated message to the sink.

n_{min} Minimum energy node.

3.2. Radio Power Model

A typical sensor node is depicted in Figure 2 and consists of four major components: a data processor unit; a micro-sensor; a radio communication subsystem that consists of transmitter/receiver electronics, antennas and an amplifier; and a power supply unit. [23]. Although energy is dissipated in all of the first three components of a sensor node, energy dissipations associated with the radio component is considered since the core objective of this study is to develop an energy-efficient network layer protocol to improve the network lifetime. In addition to this, energy dissipated during data aggregation in the cluster heads is also accounted.

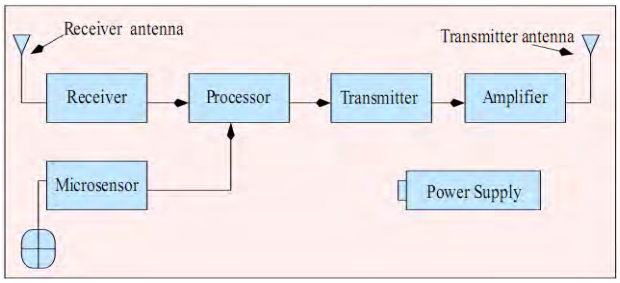


Figure 2. A typical sensor node.

The radio energy model [9] employed in our study is described in terms of the energy dissipated in transmitting k-bits of data between two nodes separated by a distancer meters and also the energy spent for receiving at the destination sensor node and is given by,

$$E_T(k,r) = E_{Tx} * k + E_{amp}(r) * k \tag{2}$$

$$E_{amp}(r) = \varepsilon_{FS} * r^2 \tag{3}$$

The energy cost incurred in the receiver is given by,

$$E_R(k) = E_{Rx} * k \tag{4}$$

where E_{amp} denote energy dissipated in the transmitter of the source node is required to maintain an acceptable signal-to-noise ratio for reliable transfer of data messages. We use free space propagation model and hence the energy dissipation of the amplifier is given by:

$$E_{amp}(r) = \varepsilon_{FS} * r^2 \tag{5}$$

where ε_{FS} denotes the transmit amplifier parameter corresponding to free space.

The assumed values for the various parameters is as given below.

$$E_{Tx} = E_{Rx} = 50\text{n}J / bit$$

$$\varepsilon_{FS} = 10\,pJ / bit / m^2$$

The energy spent for data aggregation is,

$$E_{DA} = 5\text{nJ / bit / message.}$$

4. Problem Definition

A sensor network is described by means of an edge-weighted graph, $G_{WSN}(\mathbb{N}, \mathbb{D}, \text{Sink})$, $\mathbb{N} = \{n_1, n_2 .... n_n\}$ is a set of sensor nodes and $\mathbb{D} = \{d_1, d_2 .... d_n\}$ contains the inter-node distances.

The objectives of our work are:

1) To develop an energy-efficient hierarchical routing algorithm which minimizes the energy consumption of the network.

2) To maximize the network lifetime.

4.1. Assumptions

- WSN consisting of a fixed sink with unlimited supply of energy and *n* wireless sensor nodes having limited power resources.
- The wireless sensor network can be either homogeneous or heterogeneous in nature.
- The nodes are equipped with power control capabilities to vary their transmitted power.
- Each node senses the environment at a fixed rate and always has data to send to the sink.
- The sensor nodes are aware of their geographic position as each of them are equipped with a Global Positioning System (GPS).

5. Sink Administered Load Balanced Dynamic Hierarchical Protocol (SLDHP)

This section focuses on design details of our proposed protocol SLDHP, which is a hierarchical wireless sensor network routing protocol. Here the sink with unrestrained energy plays a vital role by performing energy intensive tasks thereby bringing out the energy efficiency of the sensors and rendering the network endurable. The pattern of the hierarchy varies dynamically as it is based on energy levels of the sensors in each iteration.

SLDHP functions in two phases namely:

1) Network Configuring Phase

2) Communication Phase.

The algorithm steps are described in Table 1.

5.1. Network Configuring Phase

The goal of this phase is to establish optimal routing paths for all the sensors in the network. The key factors considered are balancing the load on the principal nodes and minimization of energy consumption for data communication. In this phase, the sink probes the sensors to send the status message that encapsulates information regarding their geographical position and current energy level. The sink upon receiving this, stores the information in its data structures to facilitate further computations. To construct the routing path, first the sink traces the node with minimum energy, n_{min} from the set N. The minimum energy node n_{min} will be alloted to the principal node, which will be selected based on the following criteria:

- The sink reckons the set $\mathbb{P}$, that contains nodes with energy above E_{avg}, which is a subset of set $\mathbb{N}$.
- It then computes the distance between n_{min} and each of the nodes in $\mathbb{P}$. Consider any two nodes with respective *x* and *y* positions given by (x_1, y_1) and (x_2, y_2). The Euclidean Distance between these two nodes is given by:

$$\sqrt{(|x_1 - x_2|)^2 + (|y_1 - y_2|)^2} \quad (6)$$

- The node in the set P which has minimum distance to n_{min} is selected as the principal node.

To aid further calculations, the amount of energy spent by the principal node on receiving and aggregating message sent from n_{min} is reduced virtually. The minimum energy node is then removed from the set $\mathbb{N}$. This phase repeats until all the nodes in the network are assigned to principal nodes. The last node that remains in set $\mathbb{N}$ is the node with maximum energy which serves as a superior node and has the job of sending the aggregated message to the sink.

The protocol gives prime importance to balance the load on the principal nodes. The minimum energy nodes will be assigned to a principal node as long as it has the capability to handle them. Once the energy of the principal node falls below E_{avg}, it will be treated as a normal node and hence will be assigned to another principal node. In this way, multihop minimal spanning tree is constructed without a need for running a separate *minimal spanning tree algorithm*. Figure 3 depicts the hierarchical setup of our protocol.

SLDHP eliminates the necessity of knowing the optimum number of clusters in the network. The load is evenly balanced depending upon the capacity of the principal nodes. The protocol starts with a chaining setup and ends in a hierarchical model. In this way, multihop, load balanced network is achieved. The concluding task of this phase is to determine the Time Division Multiple Access (TDMA) slots for all the nodes within the hierarchy. Once all the computations are over, the sink sends messages to all the sensors indicating their principal nodes and the TDMA slots.

5.2. Communication Phase

The sensors send their sensed data to their respective principal nodes. Each of the principal nodes gather data from the nodes down in their hierarchy, fuses and then forwards either to another principal node or to the sink. This phase inturn comprises of three activities.

- *Data gathering*: utilizes a time-division multiple access scheduling scheme to minimize collisions between sensor nodes trying to transmit data to the principal node.
- *Data fusion or aggregation*: Once data from all sensor nodes have been received, the principal node combines them into a target entity to greatly reduce the amount of redundant data sent to the sink.
- *Data routing*: Transfers the data along the principal node-to-principal node routing to the superior node, which transmits the fused data to the sink.

Table 1. SLDHP algorithm.

I *Network Configuring Phase*
(i) Initialize
- The $Sink$ queries all nodes regarding their status.
- Nodes reply by sending the status message.

(ii) Main Processing
- Sink computes the average energy of the network.
- begin
  - In addition to computing the average energy, the sink also does the following operations.
    - It traces n_{min}, which is the node with minimum energy. It then computes the set $\mathbb{P}$ which contains the node-ids of all the nodes with energy above E_{avg}.
    - Finds distance between n_{min} and elements of $\mathbb{P}$.
    - An element of $\mathbb{P}$, p_i having minimum distance to n_{min} is assigned as the principal node.
    - Checks whether p_i is still eligible to be in set $\mathbb{P}$. If not, it is eliminated from the set.
    - n_{min} is discarded from $\mathbb{N}$.
  - Repeat this until all the nodes have been assigned to principal nodes.
  - Schedules $TDMA$ slots for all the nodes.

(iii) Finalize
- Sink sends messages to all the nodes indicating their principal nodes and the $TDMA$ slots.

II *Communication Phase*
- Data Gathering
- Data fusion or aggregation
- Data routing

6. Performance Analysis

6.1. The Simulation Test-Bed

A homogeneous sensor network was set up with the simulation environment comprising 100 nodes, with all nodes possesing the same initial energy of 2*J*. The simulations were carried out using the Objective Modular Network Testbed in *C* + + (OMNeT++) simulator [24]. The sensor nodes were deployed randomly in a sensor field of a grid size of 500mx500m. The simulations were carried out several times, for different network configurations in order to obtain consistent results. The performance metrics considered are Average Energy Consumption by the nodes and Network Lifetime. The proposed protocol was compared with BCDCP and it was found that SLDHP performed significantly better in all simulation runs.

6.2. Average Energy Consumption of the Sensor Network

Figure 4 shows the Average Energy Consumption of the sensor network, as a variation with reference to number of iterations of the network. The simulation environment is setup with the initial battery energy of all nodes being 2J and a message length of 4 kbits/packet. We observe that the protocol greatly reduces the energy consumed and hence outperforms others in terms of battery efficiency. This is due to the minimum-spanning tree hierarchical structure formed by SLDHP as compared to the cluster-based structure which consists of equal num

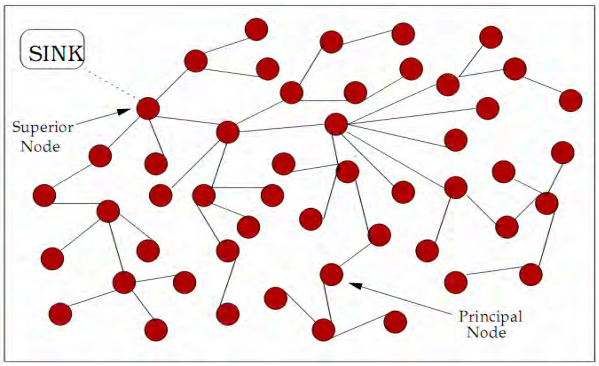


Figure 3. Hierarchical setup of SLDHP.

ber of member nodes with unequal distribution of energy. BCDCP achieves balancing by assigning equal number of nodes to each of the clusters which results in overloading the already overloaded cluster-heads to drain out much of their energy on receiving, aggregating and transmitting the data at a much faster rate. In comparison, our proposed algorithm comprises of unequal member nodes within the hierarchy, but load balanced in terms of energy resources, which contributes significantly to the increased energy efficiency of the algorithm. Hence the packet transmission time in our algorithm is predominantly short as compared to others. From the plot, it is observed that initially when the number of iterations is less, energy consumption in both the schemes is found to be almost the same, with no conspicuous results. This is due to the fact that the hierarchical structure at this point of time seems almost the same. The real advantage comes to light when the number of iterations increases, with the hierarchical structure adapting itself dynamically to the changing scenario. The superior performance offered by SLDHP enables to achieve a reduction of energy consumption by about 21% as compared to the earlier algorithms.

6.3. Sensor Network Lifespan

The energy consumption rate can directly influence the lifes-pan of the sensor nodes as the depletion of battery resources will eventually cause failure of the nodes. Hence the wireless engineer is always entrusted with the task of prolonging the lifespan of the network by improving the longevity of the sensor nodes. The simulation results of number of nodes alive over a period of time are presented in Figure 6. The simulation environment is the same, i.e., initial energy of nodes being 2J, message length being 4 kbits/packet and the initial node density being 100. Both the protocols are based on a hierarchical structure in which all the nodes rotate to take responsibility for being the cluster-head and hence no particular sensor is unfairly exploited in battery consumption. Due to the hierarchical structure, it is found that till the 806th iteration, the number of nodes that are alive is almost the same in both schemes and equals 100.

This implies that the time duration between the first exhausted node and the last one is quite short or the difference in energy levels from node to node does not vary greatly for lower number of iterations. After this critical point, both the curves in the Figure drop indicating the fall in the number of alive nodes. It is evident from the plot that the number of alive nodes is significantly more in our protocol as compared to other and which agrees with the results obtained in the previous simulation. This algorithm can extend the lifespan of the network by about 34% as compared to the earlier algorithm. It is observed that the number of alive nodes in earlier algorithm is a maximum of 100, dropping at a steady rate till none of the nodes are found to be alive at the 1800th iteration. In comparison, the nodes of SLDHP are very much live and active even for a little beyond the 2000th iteration, once again indicating the superior performance of the algorithm. The reason for this is again the same, the difference in hierarchical structure, plus the added advantage of dynamically having a load balancing scheme.

6.4. Average Energy Consumption for Varying Message Lengths

Figure 5 shows the average energy consumption of the network when SLDHP is run with the data communication phase transmitting data at varying message lengths of 4kbits/packet and 8kbits/packet respectively. From the plot, it is observed that when the message length is 4 kbits/packet, the behaviour is exactly similar to the one depicted in Figure 4 for SLDHP due to the similarities of the simulation environment set up. When the message length is doubled, the average energy consumption of the sensor network is much more as observed from the simulation results. This is quite obvious because of greater overhead involved in aggregating and transmitting a larger sized message. From the plot, it is seen that at the end of the 2000th iteration, the energy consumed for transmitting a smaller message is close to 2J while the same energy level is reached in the 1620th iteration itself, for a larger message transmission. A message length of 4 kbits/pkt seems ideal as lesser length message may not be in a position to carry out the desired task and a larger length may unnecessarily contribute to additional overhead which can degrade the performance of the network.

The plots in Figure 7 show the average energy consumption of the network with proposed algorithm run for two different message lengths. The simulation environment is set up with all the nodes equipped with a uniform initial energy of 2J. The node density is varied to account for scalability of the WSN and at the same time will aid in understanding the behaviour of the network especially in terms of energy management of the network for varying node densities. For comparatively lower value of node density, the average energy consumption of the network is smaller being a little less than 0.06J for a

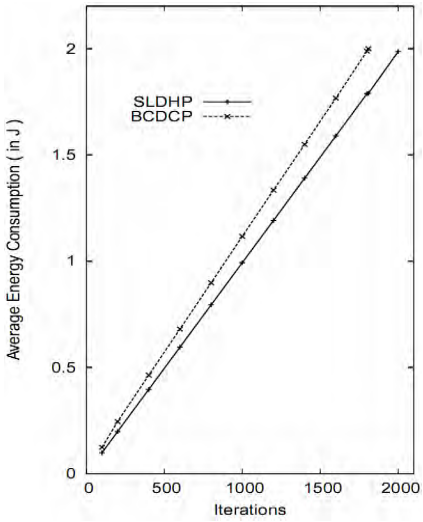


Figure 4. Comparison of average energy consumption.

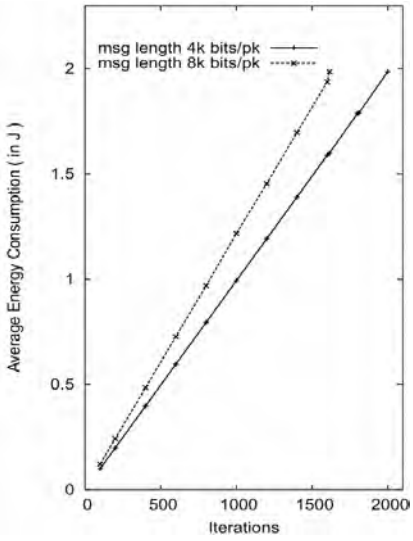


Figure 5. Average energy consumption (SLDHP) with variable size of the packet.

smaller message length, increasing steadily to about 0.09J for a node density of 100. In comparison, it is found that the energy consumption is relatively more for a larger sized message, varying from 0.078J for 40 nodes reaching a value of 0.12J for 100 nodes. This behavior is much the same as for a smaller message, the difference being that obviously more energy is consumed for a larger message size. As the number of nodes increase, the complexity of the network configuring phase also increases proportionately leading to an increased overhead on the sink to dynamically form load balanced hierarchical structures. The complexity of the data communication phase is no less, with more number of nodes being involved in data communications and with the complexity increasing with increasing nodes. The energy consumption of the network increases in proportion to the number of nodes and the same analogy holds good for

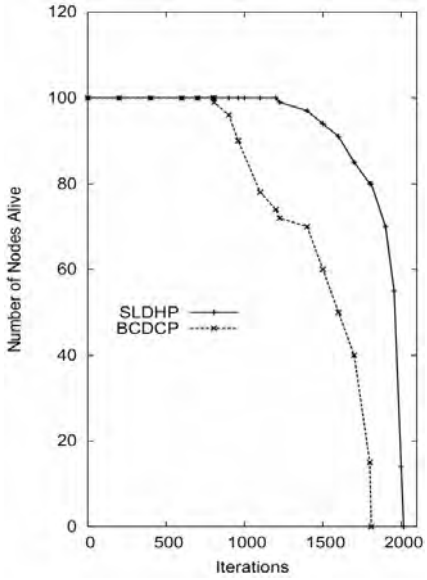


Figure 6. Comparison of lifespan of the wireless sensor network.

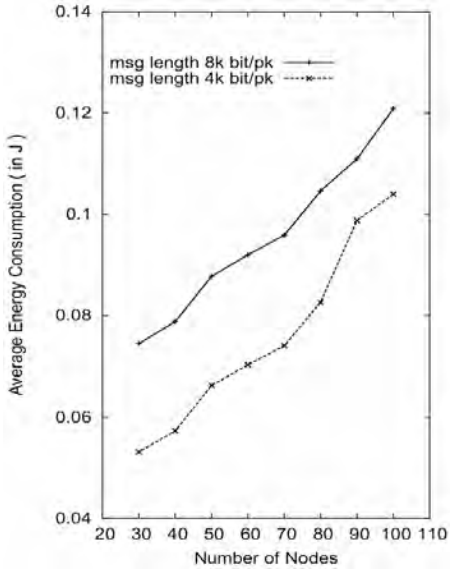


Figure 7. Average energy consumption (SLDHP) for different packet lengths.

different message lengths, the consumption being much more for larger sized messages.

6.5. Network Lifespan for Varying Size of Packet

Figure 8 shows another performance run when communications in SLDHP, take place by transmitting varying length messages of 4 kbits/packet and 8 kbits/packet. The simulations are carried out under similar conditions. As seen from the plot, when the message length is 4 kbits/packet, larger number of nodes are alive and the same is confirmed by the results obtained in Figure 6. When the message length is doubled, the saturation of the network takes place at a faster rate due to increased overhead on the sensor nodes and the principal nodes in particular. This manifests in nodes consuming larger energy, resulting in a larger transmission cost, leading to a shorter lifespan of the network. The smaller the message length, greater is the lifespan of the network with the number of live nodes prolonging the network lifespan to as long as the 2000^{th} iteration. Till the 1400^{th} iteration, the number of alive nodes in both cases seems exactly the same, but drops abruptly to zero at the 1635th iteration, for a larger message length. The reason for this is the same as described for Figure 6 and hence the same inference can be drawn here as well. Hence it is inferred that 4kbits/packet is apt for the present scenario.

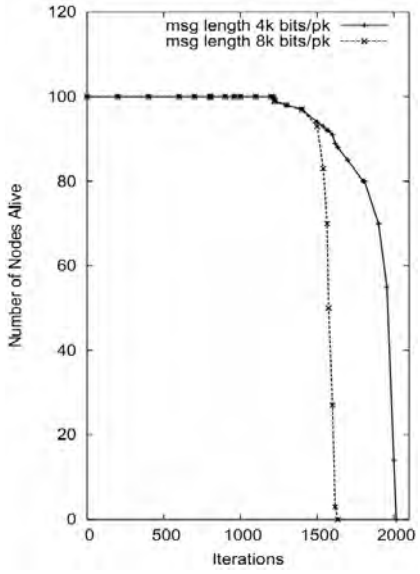


Figure 8. Lifespan of the wireless sensor network (SLDHP) with variable size of packet.

7. Conclusions

A WSN is composed of tens to thousands of sensor nodes which communicate through a wireless channel for information sharing and processing. The sensors can be deployed on a large scale for environmental monitoring and habitat study, for military surveillance, in emergent environments for search and rescue, in buildings for infrastructure health monitoring, in homes to realize a smart environment. SLDHP manages to balance the load on the principal nodes and hence the sensor nodes are relieved from the energy intensive tasks such as formation of hierarchy and scheduling of slots to send their sensed data. This job is effectively accomplished by the high powered sink. The simulation results indicate that the network lifetime is elevated to a large extent when

compared to other hierarchical routing protocols. The future work includes applying our protocol to a distributed wireless sensor network and hence to improve the network performance as in present scenario.

8. References

[1] E. Shih, S. H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," Seventh Annual ACM SIGMOBILE Conference on Mobile Computing and Networking, July 2001.

[2] J. Ibriq and I. Mahgoub, "Cluster-based routing in wireless sensor networks: Issues and challenges," SPECTS, pp. 759–766, 2004.

[3] I. F. Akylidiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor network: A survey on sensor networks," IEEE Communications Magazine, Vol. 40, No. 8, pp. 102–114, August 2002.

[4] M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," Twenty-First Annual Joint Conference of the IEEE Computer and Communications Society, INFOCOMM, 2002.

[5] J. Agre and L. Clare, "An integrated architecture for cooperative sensing networks," IEEE Computer Magazine, pp. 106–108, May 2000.

[6] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," Proc. 33rd Hawaii Int'l. Conf. Sys. Sci., January 2000.

[7] W. B. Heizelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660–670, October 2002.

[8] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," IEEE Trans. Parallel and Distrib. Sys., Vol. 13, No. 9, pp. 924–935, September 2002.

[9] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," IEEE Communications Magazine, Vol. 43, pp. 8–13, March 2005.

[10] G. Huang, X. Li, and J. He, "Energy-efficiency analysis of cluster-based routing protocols in wireless sensor networks," IEEE Aerospace Conference, March 2006.

[11] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, pp. 159–169, May 2001.

[12] A. Depedri, A. Zanella, and R. Verdone, "An energy efficient protocol for wireless sensor networks," December 2003.

[13] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous sensor networks: A comparative study," Proceedings of International Conference on Communications (ICC 2004), June 2004.

[14] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Transactions on Mobile Computing, Vol. 3, No. 4, December 2004.

[15] A. D. Amis and R. Prakash, "Load-balancing clusters in wireless ad hoc networks," Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00), 2000.

[16] Z. Zhang and G. Zheng, "A cluster based query protocol for wireless sensor networks," The 8th International Conference on Advanced Communication Technology, Vol. 1, pp. 140–145, February 2006.

[17] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A stable election protocol for clustered heterogenous wireless sensor networks," The 8th International Conference on Advanced Communication Technology, 2004.

[18] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, "Optimal energy aware clustering in sensor networks," Sensors, Vol. 2, pp. 258–269, July 2002.

[19] U. P. Han, S. E. Park, S. N. Kim, and Y. J. Chung, "An enhanced cluster based routing algorithm for wireless sensor networks," International Conference on Parallel and Distributed Processing Techniques and Applications, Vol. 1, June 2006.

[20] Y. Chen and N. Nasser. "Energy-balancing multipath routing protocol for wireless sensor networks," Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Vol. 191, 2006.

[21] L. Lin, N. B. Shroff, and R. Srikant, "Energy-aware routing in sensor networks: A large systems approach," WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services, pp. 159–169, January 2006.

[22] R. C. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," WCNC 2002 Conference, March 2002.

[23] V. Raghunathan *et al*., "Energy aware wireless microsensor networks," IEEE Signal Processing Magazine, Vol. 1, No. 2, pp. 40–50, March 2002.

[24] A. Vargas, OMNeT++ Discrete Event Simulator System, version 2.3 edition, 2003.

Wireless Sensor Network, **2009, 1, 350-357**
doi:10.4236/wsn.2009.14043 Published Online November 2009 (http://www.scirp.org/journal/wsn).

Scientific Research

Minimization of Collision in Energy Constrained Wireless Sensor Network

Moses Nesa SUDHA[1], Muniappan Lakshapalam VALARMATHI[2], George RAJSEKAR[1], Michael Kurien MATHEW[1], Nagarajan DINESHRAJ[1], Sivasankaran RAJBARATH[1]
[1]*Karunya University, Coimbatore, India*
[2]*Government College of Technology, Coimbatore, India*
Email: *nesasudha@yahoo.com, michaelmathew.87@gmail.com*
Received May 13, 2009; *revised July* 20, 2009; *accepted July* 27, 2009

Abstract

Wireless Sensor Networks (WSNs) are one of the fastest growing and emerging technologies in the field of Wireless Networking today. The applications of WSNs are extensively spread over areas like Military, Environment, Health Care, Communication and many more. These networks are powered by batteries and hence energy optimization is a major concern. One of the factors that reduce the energy efficiency of the WSN is collision which occurs due to the high density of data packets in a typical communication channel. This paper aims at minimizing the effects of congestion leading to collision in the network by proposing an effective algorithm. This can be done by optimizing the size of the contention window by introducing parameters like source count and α. If the contention window of a node is low, it results in collision. If the size of the contention window of a node is high then it results in a medium access delay. Thus minimizing collision and medium access delay of data packets conserve energy.

Keywords: Energy, Collision, Contention Window, Wireless Sensor Networks

1. Introduction

Wireless Sensor Networks (WSNs) are a typical type of wireless networks consisting of a large number of sensor nodes. WSNs are undoubtedly one of the largest growing types of networks today. They are fast becoming one of the largest growing networks today and, as such, have attracted quite a bit of research interest. They are used in many aspects of our lives including environmental analysis and monitoring, battlefield surveillance and management, emergency response, medical monitoring and inventory management. These networks also play a significant role in areas like agriculture and industries as well. Their reliability, cost-effectiveness, ease of deployment and ability to operate in an unattended environment, among other positive characteristics, make sensor networks the leading choice of networks for these applications.

Much research has been done to make these networks operate more efficiently including the application of data aggregation. A wireless network normally consists of a large number of distributed nodes that organize themselves in an ad-hoc fashion. Each node has one or more sensors, embedded processors and low power radios which are normally battery operated. Unlike other wireless networks, it is generally difficult or impractical to charge/replace exhausted batteries. That is why the primary objective in wireless sensor networks design is maximizing node/network lifetime, leaving the other performance metrics as secondary objectives. Various factors like concurrent transmissions, buffer overflows and dynamically time varying wireless channel conditions lead to the concept of Congestion [1]. Collision has the following drawbacks: 1) increase energy dissipation rates of sensor nodes, 2) causes a lot of packet loss, which in turn diminish the network throughput and 3) hinders fair event detections and reliable data transmissions [2,3]. Congestion control or congestion avoidance has thus become very crucial for effective transmission of data packets [4]. The main reason of congestion in WSN, is allowing sensing nodes to transfer as many packets as they can [2]. Hence it can be inferred that, congestion in wireless networks leads to collision between the packets transmitted. Collision occurs when two nodes send data at the same time, over the same transmission medium or channel. Medium Access Control (MAC) Protocols have been developed to assist each node to decide when and how to access the channel [1–10]. However, the medium- access decision within a dense network composed of nodes with low duty-cycles

is a challenging problem that must be solved in an energy-efficient manner. Keeping this in mind, emphasis is first given to the peculiar features of sensor networks, including reasons for potential energy wastage at medium-access communication and how they can be minimized.

2. Priority Based MAC Protocol

2.1. Configuration Requirements

First the topology of the entire Wireless Sensor Network (WSN) is set as required. The MAC type used here is 802.11. For transmission and reception of data packets to take place in a WSN, there is a need to have a source node, the transmission paths to be followed, and a sink node. Source nodes can vary but sink nodes are fixed once the transmission of data packets occur. The final collection of the transmitted data occurs at the sink node. This collected data is taken and used according to the application needed. Apart from this various other parameters like transmission range, packet size, sink location, data rate, simulation time and initial energy are all given as initial settings along with the topology formation.

2.2. Calculation of Sensing Nodes (Ns)

N_s is the approximate number of nodes within the sensing radius of a particular event. We consider a network of N sensing nodes, deployed with uniform random distribution over an area A. The Node density is defined as $\rho = N/A$. And N_s is calculated as,

$$N_s = \pi \rho R_s^2 \quad (1)$$

where, R_s is the sensing range of each node. A single sink node in the network placed anywhere within the terrain is taken into consideration. Mobile sensors which form a dynamic ad-hoc network are not considered. All sensing nodes considered are static and the network is homogeneous i.e., all nodes have the same processing power and equal sensing and transmission range. Data generation rate of each sensing node is also assumed to be equal.

2.3. Priority Based Source Count

Source Count value of any node i, denoted as SC_i, is defined as the total number of nodes to which it is able to forward data. In other words, it is the number of downstream nodes for a particular node, which responds to the advertisement of the node. Since a downstream node requires knowing its Source Count (SC) value whenever it has some data packets to send, it is sufficient to propagate SC value along with the data packet. While transmitting data packets, each upstream node inserts its SC value in the packet header and the downstream node can easily obtain its SC value. An upstream node learns the SC value of its downstream by snooping packets transmitted by the latter. Note that, a transient state exists between the event occurrence and the stabilization of SC values of all downstream nodes. SC value of a downstream node is stabilized whenever it receives at least one packet from all of its upstream nodes and therefore the network enters into steady state when the sink node receives at least one packet from each source node. Since the duration of transient state is very short (less than a second in our simulation), the effectiveness of the proposed protocol is not hampered. It is notable that, SC values of each node along the routing path are updated without transferring any additional control packets. This SC parameter works as a driving entity for all schemes of our proposed protocol. Thus the SC values for all the nodes that are involved in transmission and receptions of data packets are calculated. With the help of these values the priority of transmission is assigned to each node. This helps in minimizing the collision in the Wireless Sensor Network.

2.4. Calculation of Contention Window

Contention Window is a parameter which depends on time [1]. It determines the rate of flow of data packets and medium access delay. Now the Contention Window value is calculated for each node that is involved in the communication process. It is calculated as follows:

$$W(i) = CWmin \times (N_s/Sc_i) \quad (2)$$

where, W (i)-Contention Window value for any node i, CWmin-Minimum Contention Window value, N_s- Approximate number of nodes within the sensing radius of a particular event, Sc_i-Source Count value of any node i.

2.5. Inclusion of α Parameter

The parameter α is a scaling factor that is introduced in Equation (2) to optimize effects of collision and medium access delay. It ranges from 0.1 to 2 based on channel contention.

$$W(i) = CWmin \times (N_s / Sc_i) \times (1/\alpha) \quad (3)$$

If the number of contending neighbors of a transmitting node is very low, lower value of α simply increases the medium access delay and reduces the network throughput. On the other hand, if the number of contending neighbors of a transmitting node is very high, a higher value of α increases the collision probability and thereby increases packet loss. The value of α is initialized to 1, which nullifies its effect. Later on, to ensure efficient medium utilization, the value of α is set carefully. A sharp increase or decrease of the value of α may

also hinder the throughput of the network. Sections 3.1 and 3.2 describe the variation of α.

2.6. Idealization of Contention Window

The limitation of Equation (2) is that the contention window cannot be varied for different number of data packets. But we know that window size is directly proportional to packet size and inversely proportional to data rate. Hence we have another equation:

$$W(i) = (\text{Packet size} \times \text{No. of data packets})/\text{Data rate} \quad (4)$$

Equation (4), is used to vary the α value in Equation (3) and thereby an optimized contention window is obtained in order to minimize the effects of collision and delay, in the process of communication, simultaneously.

2.7. Idle Listening

In the above sections, the effect of collision and some parameters associated with it have been analyzed. In this section, another factor has been taken into account which leads to some amount of energy loss in MAC protocols - idle listening [11]. Since a node does not know when it will be the receiver of a message from one of its neighbors, it must keep its radio in receive mode at all times. So it loses energy as long as it is ON. Hence, the nodes which do not take part in the communication, loses energy due to idle listening. Here in this paper, this phenomenon has been considered in Sections 2.8 and 3.3. As a result, a particular amount of energy is conserved and better energy efficiency is obtained for each node in the network.

2.8. Evaluation with Idle Listening

In this paper, each node in the network is enabled when it receives or transmits data packets. If a node does not involve in communication, it is disabled, whereby no further transmission or reception of data packets take place. The amount of energy lost by keeping a node in the ON state is approximately 50–100% of the receiving energy. In the scenarios explained in Sections 3.1 and 3.2, the energy loss due to the node being in the ON state is 66% of the receiving energy. When the nodes that are not involved in the communication process are disabled, this energy is saved and we obtain higher energy efficiency.

3. Energy Analysis

In the above sections, we have discussed about the phenomenon of contention window and idle listening. In this section, we will discuss two scenarios in which the contention window is varied to consider the effects of collision and medium access delay.

3.1. Scenario 1

In the first scenario, we have eight nodes placed in a randomly chosen topology. The simulation would be carried out according to the parameters mentioned in the above table. Here we would consider the effect of collision for each node in the network.

Using the specifications given in Table 2, data packets are transmitted. Here the data packets are very high in number and so when they are transmitted, collision occurs. In order to minimize the collision, we vary the contention window by decreasing α value. By doing so the contention window size is increased and thereby collision is minimized. This variation of α is done by keeping the value of contention window obtained from Equation (3) as reference.

Table 1. Simulation parameters.

Parameter	*Value*
Total Area	500 x 500
Number of nodes	8
MAC Type	802.11
Initial Energy	5 Joule/Node
Transmission Energy	0.0005 Joule/Node
Reception Energy	0.0003 Joule/Node
ON-Time Energy	0.0002 Joule/Node
Data Rate	10 Bytes/s
Packet Size	64 Bytes
Initial α Value	1
Range of α Value	0.1 ~ 2
Simulation Time	150 ms

Table 2. Source count and No. of packets.

Node	*Source Count*	*No. Of Packets*
0	3	10
1	3	12
2	2	4
3	1	18
4	1	11
5	4	9
6	5	7
7	5	12

Using the specifications given in Table 2, data packets are transmitted. Here the data packets are very high in number and so when they are transmitted, collision occurs. In order to minimize the collision, we vary the contention window by decreasing α value. By doing so the contention window size is increased and thereby collision is minimized. This variation of α is done by keeping the value of contention window obtained from Equation (3) as reference.

Figure 1 is a diagrammatic representation which shows that, more number of data packets is sent within the limited time frame and as a result, collision is occurring.

$$\uparrow W(i) = Wmin \times (Ns/SCi) \times (1/\alpha)\downarrow \quad (5)$$

3.2. Scenario 2

In the second scenario, we have eight nodes placed in a randomly chosen topology. The simulation would be carried out according to the parameters mentioned in Table 3. Here, the effect of medium access delay for each node in the network is considered. The number of packets transmitted by each node would be lesser than the number considered in the first case of collision.

Using the above specification, data packets are transmitted. Since they are very less in number medium access delay occurs. In order to minimize this delay, we vary the contention window by increasing the α value. By doing so the contention window size is decreased and thereby medium access delay is minimized. This variation of α is done by keeping the value of contention window obtained from Equation (3).

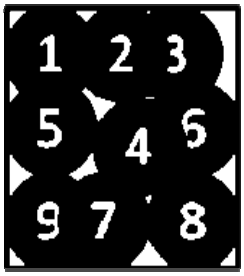


Figure 1. Collision of packets when contention window size is small.

Table 3. Source count and No. of packets.

Node	*Source Count*	*No. Of Packets*
0	3	2
1	3	2
2	2	3
3	1	6
4	1	6
5	4	1
6	5	1
7	5	1

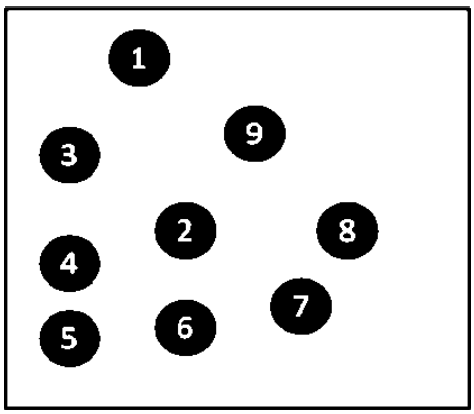


Figure 2. Medium access delay in the network when contention window size is large.

Figure 2 is a diagrammatic representation which shows that, less number of data packets is sent within the large time frame and as a result, medium access delay is occurring.

$$\uparrow W(i) = Wmin \times (Ns/SCi) \times (1/\alpha)\downarrow \quad \mathbf{(6)}$$

3.3. Conclusion of Energy Analysis

In Scenario 1, to minimize the effect of collision, the size of the contention window is increased by decreasing the α value. In Scenario 2, to minimize the effect of medium access delay, α value is increased. This results in minimizing the medium access delay.

Thus taking into account whether collision occurs or medium access delay occurs, α value is varied and thereby the contention window is also varied. Thus an idealized contention window required for the communication process is obtained which will minimize the effect of collision and medium access delay to a greater extent, producing high efficiency.

In the above 2 cases energy loss due to idle listening is reduced. This is achieved by disabling the nodes when transmission and reception of data packets do not occur. However, the energy conserved in the above scenarios was observed to be considerably less. Further analysis can be done on implementing a better algorithm to reduce the effect of idle listening.

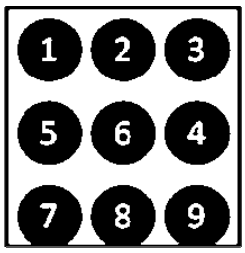


Figure 3. Optimized contention window in order to minimize both collision and medium access delay.

Figure 3 is a diagrammatic representation which shows the ideal condition that the data packets is sent within the idealized time frame and as a result collision and medium access delay is minimized to a great extent.

4. Results

Considering the above mentioned topology and simulation parameters two cases have been analyzed. In the first case the effect of collision has been minimized and in the second case, the phenomenon of medium access delay has been dealt with. Thus we try to obtain an ideal contention window size whereby both these effects are dealt with effectively. The implementation of the following scenarios have been done in Network Simulator-2 (Ns-2), version 2.28 [12].

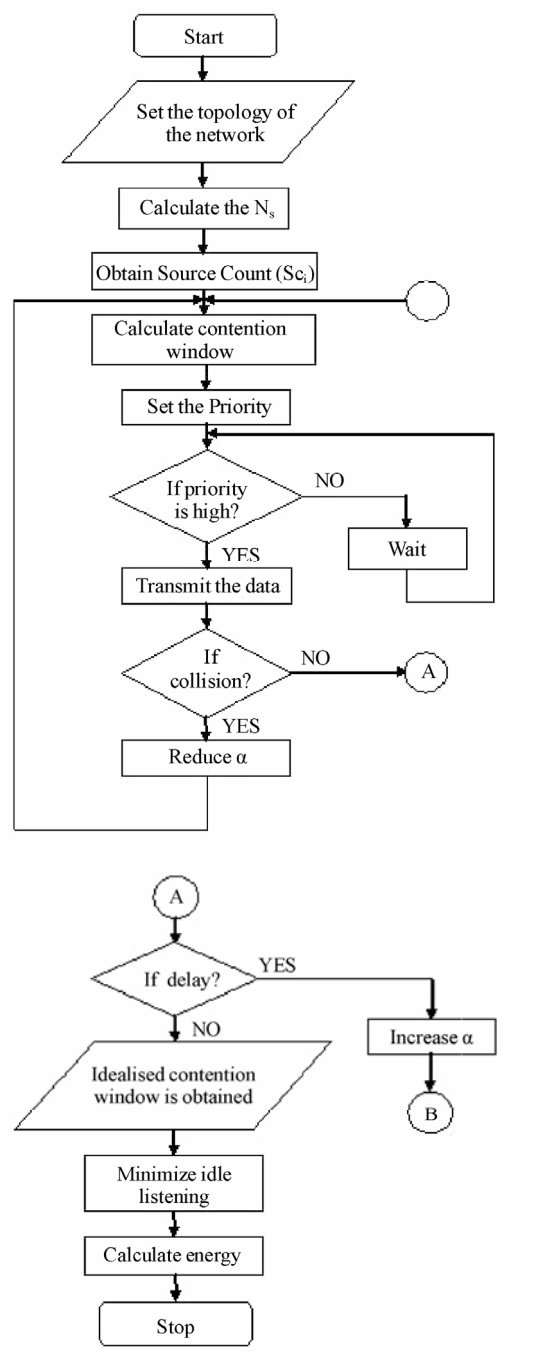


Figure 4. Flow-diagram which explains the flow of the entire process of the proposed protocol.

4.1. Scenario 1

Here in this scenario, consider that each node in the network has a higher number of packets to send to the downstream nodes in the network. Due to this, the contention window would be small and thus its size been increased. Thus it has been found that the collision is minimized effectively. As a result, the energy lost in each node has been reduced. Furthermore, a small amount of energy has been conserved by reducing idle listening.

It has been observed that, though the collision has been minimized, a very small medium access delay occurs with each node in the network.

Table 4. Results considering collision.

Node	Energy Remaining in Node			
	With-out SC	With SC	With SC & CW	With SC, CW and Idle listening
0	4.7049	4.7349	4.7899	4.7919
1	4.7329	4.7679	4.8329	4.8349
2	4.8634	4.9109	4.9159	4.9199
3	4.8574	4.8574	4.9324	4.9374
4	4.8384	4.8919	4.9219	4.9259
5	4.7529	4.7979	4.8479	4.8499
6	4.7849	4.8729	4.9129	4.9269
7	4.6874	4.8014	4.8714	4.8764

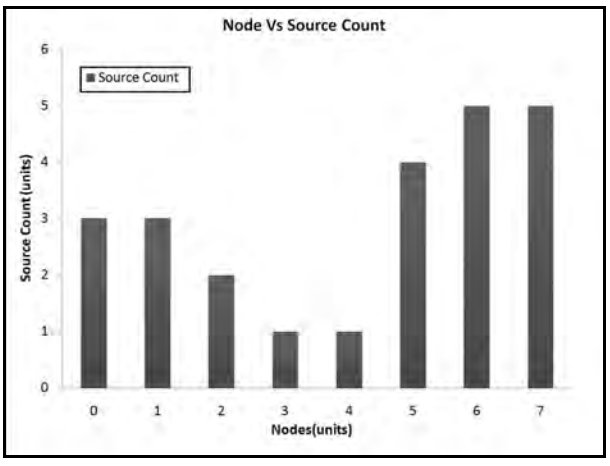


Figure 5. The source count value for each node.

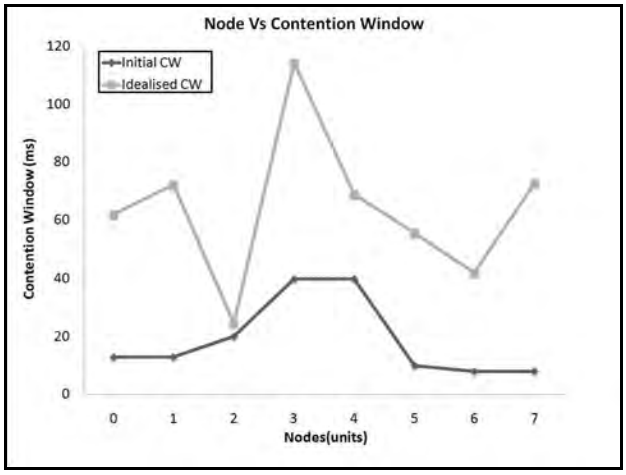


Figure 6. The initial contention window and idealized contention window for each node.

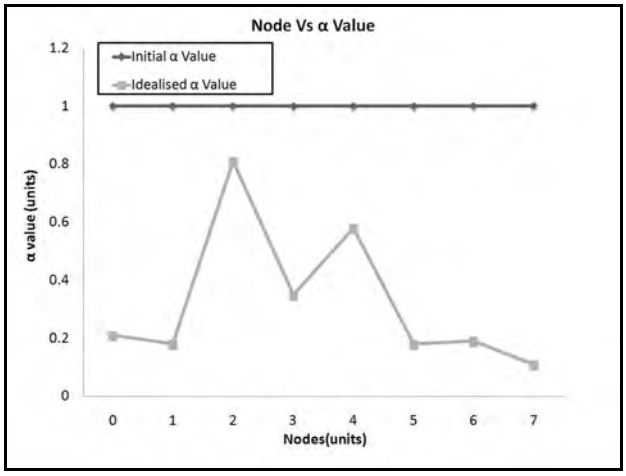


Figure 7. The α value which has been reduced due to the collision which occurs in each node.

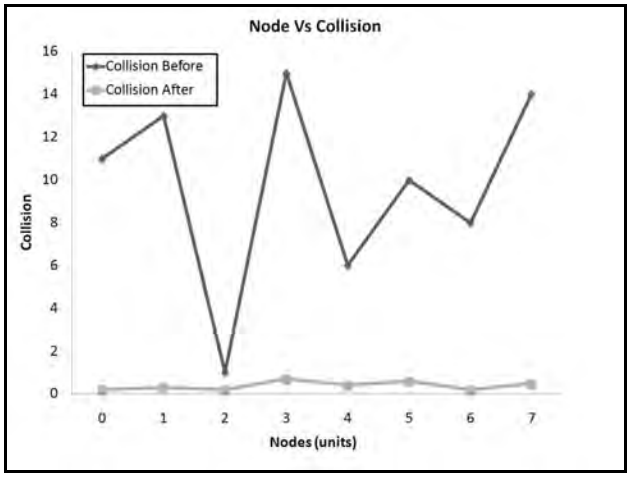


Figure 8. The collision occurring and the collision which has been minimized for each node.

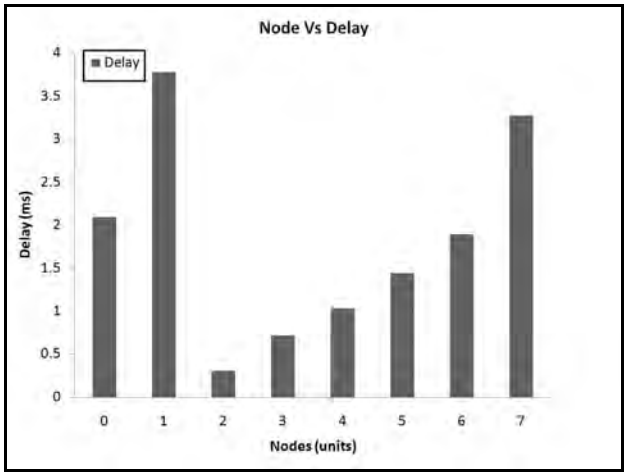


Figure 9. The delay occurring after obtaining the ideal size for the contention window for each node.

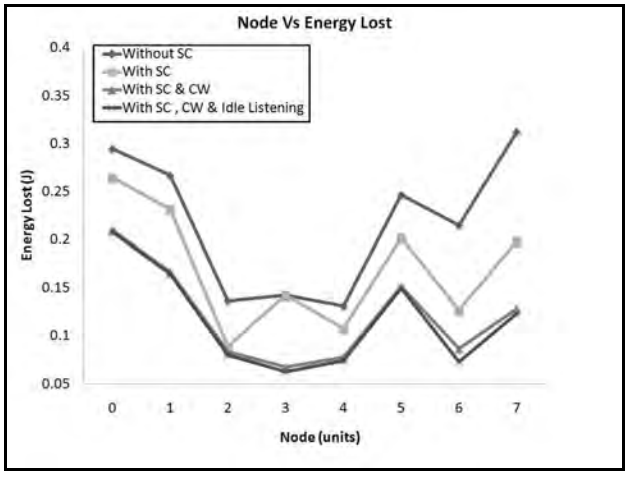


Figure 10. The energy lost comparison for each node after applying the various cases.

Table 5. Results considering medium access delay.

Node	Energy Remaining in Node			
	Without SC	With SC	With SC & CW	With SC, CW and Idle listening
0	4.7049	4.7899	4.7899	4.7919
1	4.7329	4.8329	4.8329	4.8349
2	4.8634	4.9159	4.9159	4.9199
3	4.8574	4.9329	4.9324	4.9374
4	4.8384	4.9219	4.9219	4.9259
5	4.7529	4.8479	4.8479	4.8499
6	4.7849	4.9129	4.9129	4.9269
7	4.6874	4.8714	4.8714	4.8764

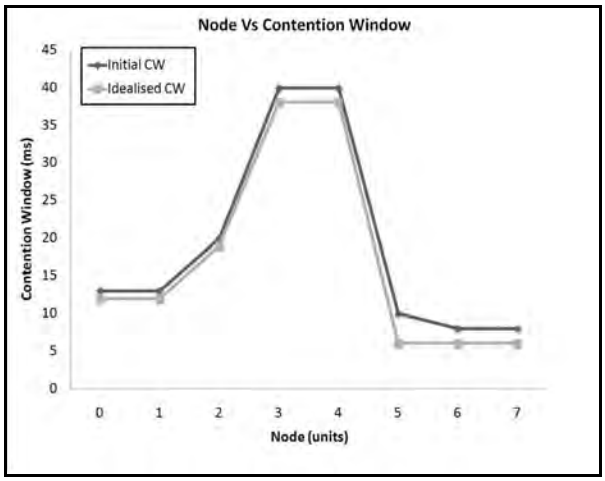


Figure 11. The graph above shows the initial contention window size and the idealized contention window for each node.

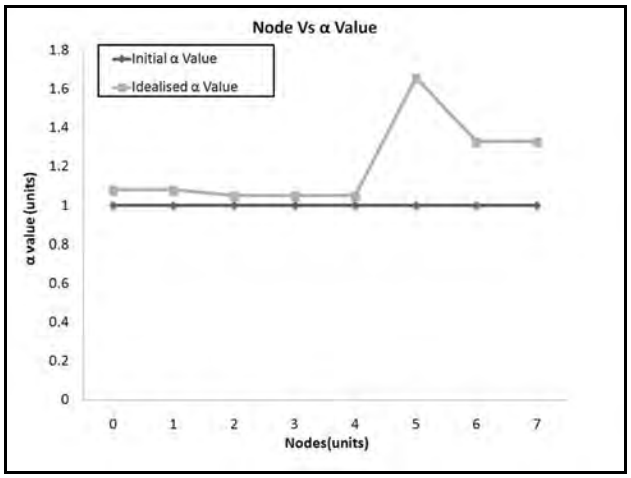


Figure 12. The α value which has been reduced due to the collision which occurs in each node.

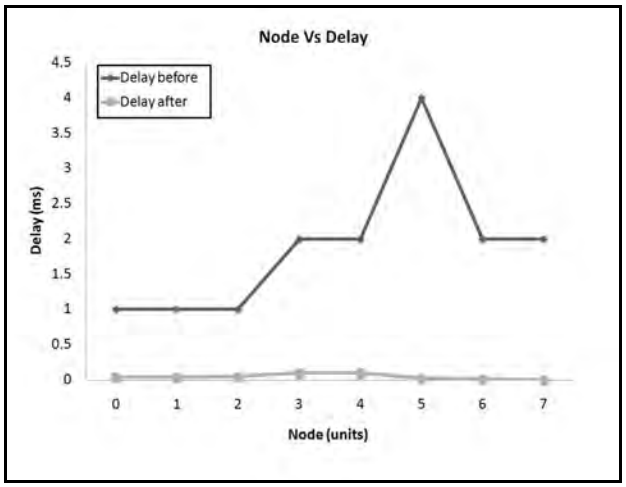


Figure 13. The graph above shows the initial delay and the minimized delay for each node.

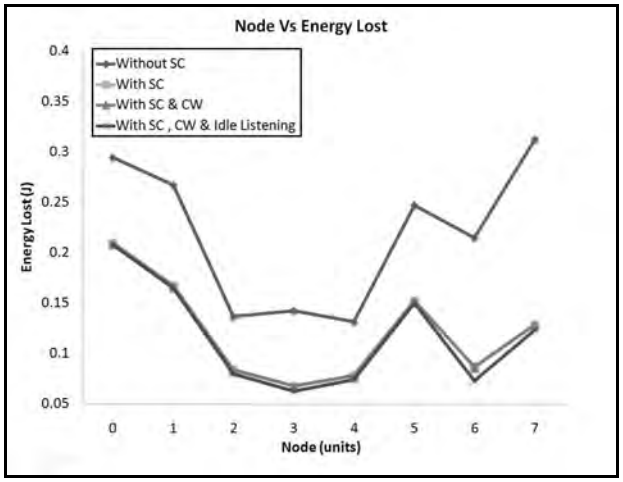


Figure 14. The energy lost comparison for each node after applying the various cases.

4.2. Scenario 2

In this scenario, consider that each node in the network has a smaller number of packets to send to the downstream nodes in the network. Due to this, the contention window would be larger than required and thus its size has been reduced. The contention window size is varied in such a way that the medium access delay becomes negligible and collision is avoided.

It has been observed that, medium access delay has been minimized and along with effects of collision. This has helped in increasing the throughput of the network. Furthermore the effect of idle listening and the energy loss caused by it has been dealt with.

5. Conclusions

In this paper, a novel method of minimization of collision and medium access delay is introduced to reduce the loss of energy of the nodes in the network. Here a Source Count value is considered for each node, in order to reduce the collision of packets in the network. The Source Count value prioritizes the transmission of packets from each node, whereby the collision of data packets and medium access delay associated with each node during communication process are minimized. The Contention Window size is calculated in accordance with Source Count values assigned for each node. The initial efficiency of the WSN was found to be around 70%. After applying the various parameters to minimize collision and medium access delay, the efficiency was increased to around 85-90%. Hence a substantial amount of energy can be saved through this method. Without optimizing on idle listening, the efficiency was found to be around 82%.

Though the idealized contention window size has been calculated, the chances of collision and medium access delay may still prevail in a minimal amount in the network. Hence it should be noted that a MAC protocol needs to be introduced which could eliminate the effects of collision and medium access delay simultaneously. Also the energy conserved by minimizing idle listening, can further be improved.

7. References

[1] Mamun-Or-Rashid, M. Alam, M. M. Razzaque and M. A. Hong, "Reliable event detection and congestion avoidance in wireless sensor networks," Networking Lab, Department of Computer Engineering, Kyung Hee University, South Korea, 2007.

[2] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," In the Proceedings of ACM SenSys, Los Angeles, ACM Press, New York, pp. 266–279, 2003.

[3] Y. Sankarasubramaniam, A. Ozgur, and I. Akyildiz, "ESRT Event-to-Sink Reliable Transport in wireless sensor networks," In the Proceedings of ACM Mobihoc, ACM Press, New York, pp. 177–189, 2003.

[4] S. Chen and N. Yang, "Congestion avoidance based on lightweight buffer management in sensor networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 17, No. 9, pp. 934–946, 2006.

[5] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Pumpslowly, fetch-quickly (PSFQ): A reliable transport protocol for sensor networks," IEEE Journal of Selected Areas in Communication, Vol. 23, No. 4, pp. 862–872, 2005.

[6] R. Stann and J. Heidemann, "RMST reliable data transport in sensor networks," In the Proceedings of IEEE SPNA, Anchorage, Alaska, IEEE Computer Society Press, Los Alamitos, pp. 102–112, 2003.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," In the Proceedings of ACM MobiCom 2000, Boston, MA, ACM Press, New York, pp. 56–67, 2000.

[8] J. Kang, Y. Zhang, and B. Nath, "TARA topology aware resource adaptation to alleviate congestion in sensor networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 18, No. 7, pp. 919–931, 2007.

[9] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, "A survey of transport protocols for wireless sensor networks," IEEE Network Magazine, Vol. 20, No. 3, pp. 34–40, 2006.

[10] A. Woo, and D. Culler, "A transmission control scheme for media access in sensor networks," In the Proceedings of ACM MobiCom, ACM Press, New York, pp. 221–235, 2001.

[11] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy Efficient, Collision Free, Medium Access Control for Wireless Sensor Networks," SenSys'03, Los Angeles, California, USA, November 2003.

[12] The Network Simulator–ns-2, http://www.isi.edu/nsnam/ ns/index.

Wireless Sensor Network, **2009, 1, 358-364**
doi:10.4236/wsn.2009.14044 Published Online November 2009 (http://www.scirp.org/journal/wsn).

An Energy-Efficient MAC Protocol for WSNs: Game-Theoretic Constraint Optimization with Multiple Objectives

Liqiang ZHAO, Le GUO, Li CONG, Hailin ZHANG
State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
Email: *lqzhao@mail.xidian.edu.cn*
Received March 23, 2009; *revised June* 2, 2009; *accepted June* 10, 2009

Abstract

In WSNs, energy conservation is the primary goal, while throughput and delay are less important. This results in a tradeoff between performance (e.g., throughput, delay, jitter, and packet-loss-rate) and energy consumption. In this paper, the problem of energy-efficient MAC protocols in WSNs is modeled as a game-theoretic constraint optimization with multiple objectives. After introducing incompletely cooperative game theory, based on the estimated game state (e.g., the number of competing nodes), each node independently implements the optimal equilibrium strategy under the given constraints (e.g., the used energy and QoS requirements). Moreover, a simplified game-theoretic constraint optimization scheme (G-ConOpt) is presented in this paper, which is easy to be implemented in current WSNs. Simulation results show that G-ConOpt can increase system performance while still maintaining reasonable energy consumption.

Keywords: Wireless Sensor Network, MAC, Energy Efficiency, Game Theory, Constraint Optimization

1. Introduction

As an emerging technology, Wireless Sensor Networks (WSNs) have a wide range of potential applications including environment monitoring, smart spaces, medical systems and robotic exploration. Performance analysis and optimization of WSNs, especially its Medium Access Control (MAC) protocols, have attracted much research interests. Traditional MAC protocols for wireless ad hoc networks are designed to maximize throughput and minimize delay. As sensor nodes are generally battery-operated, to design a good MAC protocol for WSNs, the first attribute that has to be considered is energy consumption [1]. Other important attributes (such as throughput and delay) are generally the primary concerns in traditional wireless ad hoc networks, but in WSNs they are secondary.

IEEE 802.11 Distributed Coordination Function (DCF), the basic MAC protocol in Wireless LANs (WLANs), is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), one of typical contention-based MAC protocols. CSMA/CA uses an acknowledgment (ACK) mechanism for verifying successful transmissions and optionally, an RTS/CTS handshaking mechanism for decreasing collisions overhead. In both cases an exponential backoff mechanism is used. Before transmitting, a node generates a random slotted backoff interval, and the number of the backoff slots is uniformly chosen in the range [0, *CW*-1]. At the first transmission attempt, the contention window, *CW*, is set equal to a value CW_{min} called the minimum contention window. After each unsuccessful transmission, *CW* is doubled up to the maximum value CW_{max}. Once *CW* reaches CW_{max}, it will remain at the value until the packet is transmitted successfully or the retransmission time reaches retry limit. While the limit is reached, retransmission attempts will cease and the packet will be discarded. Currently, CSMA/CA has been the de facto MAC standard for wireless ad hoc networks, widely used in almost all of the testbeds. Moreover, low-power, low-rate Wireless PANs (WPANs) such as IEEE 802.15.4 utilizes CSMA/CA too. However, the energy consumption using CSMA/CA is very high when nodes are in an idle mode. It is mainly called problem of idle listening. CSMA/CA-based S-MAC is explicitly designed for WSNs to solve this problem [2]. The basic idea of S-MAC is that used energy is traded for throughput and delay by introducing an active/sleep duty period. Some researchers are attempting to improve the performance of S-MAC [3–6]. To handle load variations in time and location, T-MAC introduces an adaptive duty cycle by dynamically ending its active part. This reduces

the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput [7].

Recently, game theory [8] becomes a very good tool to analyze and improve the performance of contention-based protocols. Game-theoretic approaches were proposed to solve the problem of security, query routing, and power control respectively in distributed sensor networks [9–12].

When using game theory in WSNs rather than mathematics or economics, much attention should be paid to the context of WSNs. For example, explicit cooperation among nodes is clearly impractical in WSNs as it causes additional energy and bandwidth consumption. We presented a novel concept of incompletely cooperative game theory to improve the performance of MAC protocols in WSNs without any explicit cooperation among nodes [13–14].

In this paper, the preliminary results presented in [13–14] will be substantially extended. The problem of energy-efficient MAC protocols for WSNs is modeled as game-theoretic constraint optimization with multiple objectives, e.g., energy consumption and QoS metrics.

2. Game-Theoretic Constraint Optimization

A node starts a game process when a new packet arrives at the node's transmission buffer and ends it when the packet is moved out of the buffer (i.e., transmitted successfully or discarded). Each game process includes many time slots and each time slot corresponds to one game state. In each time slot, each player (i.e., node) estimates the current game state based on its history. After estimating the game state, the player adjusts its own equilibrium strategy by tuning its local contention parameters. Then all the nodes take actions simultaneously, i.e., transmitting, listening, or sleeping. Although the player does not know which action the other nodes (i.e., its opponents) are taking now, it can predict its opponents' actions according to its history.

In the game, each player takes a distributed approach of detecting and estimating the current game state, and tuning its local contention parameters to the estimated game state.

In economics, normally, the optimal target of the player is to maximum its own profits. However, in WSNs, the target of each player is to maximum the system performance under certain limits, e.g., energy consumption and QoS requirements.

In the game for WSNs, the utility function of the player (i.e., node *i*) is represented by $\boldsymbol{\mu}_i = \boldsymbol{\mu}_i\left(s_i, \overline{s}_i\right)$. The parameters of the vector, $\mu_{i,j}$ correspond to its energy consumption and QoS requirements, e.g., bandwidth, delay, jitter, and packet-loss-rate. Obviously, there are some limits on its utility function, called $\boldsymbol{\mu}_i^{\max}$, e.g., the maximum energy consumption, the tolerant minimum bandwidth, maximum delay, jitter, or packet-loss-rate. If we do not consider its opponents, the strategy of the player, s_i, includes three possible actions: *transmitting*, *listening* or *sleeping*.

The strategy profile of its opponents (i.e., all the other *n* neighbors) is defined as $\overline{s}_i = \left(s_1, s_2, ..., s_{i-1}, s_{i+1}, ..., s_n\right)$. Similarly, we can get the utility function of its opponents that $\overline{\mu}_i = \overline{\mu}_i\left(\overline{s}_i, s_i\right)$. Also, there are some limits on the above utility function, called $\overline{\boldsymbol{\mu}}_i^{\max}$.

In many game-theoretic models, a player is a node contending for the channel. As there may be many nodes in a WSN and each node may contend for the channel repeatedly, a very complicated method is needed to determine the strategy. Hence, in the game, a player is not always a node. If we analyze the equilibrium strategy of node *i*, Player 1 is node *i*, and Player 2 (i.e., its opponents) is all the other *n*-1 nodes. In fact, it is possible for Player 1 to estimate Player 2's state, and difficult for Player 1 to estimate the states of each node in Player 2. In a formal description, we are looking for

$$\begin{cases} s_i^* = \arg\min\limits_{s_i} \sum\limits_j \left| \dfrac{\overline{\mu}_{i,j}^* - \overline{\mu}_{i,j}}{\overline{\mu}_{i,j}^*} \right| \left(\boldsymbol{\mu}_i < \boldsymbol{\mu}_i^{\max}\right) \\ \overline{s}_i^* = \arg\min\limits_{\overline{s}_i} \sum\limits_j \left| \dfrac{\mu_{i,j}^* - \mu_{i,j}}{\mu_{i,j}^*} \right| \left(\overline{\boldsymbol{\mu}}_i < \overline{\boldsymbol{\mu}}_i^{\max}\right) \end{cases} \quad (1)$$

Obviously, Player 1 adjusts its strategy s_i not to obtain its own optimal utility ($\boldsymbol{\mu}_i^*$), but to help Player 2 get the optimal utility ($\overline{\boldsymbol{\mu}}_i^*$); vice verse. Hence, it indicates that all the nodes play the cooperative game based on the estimated game states. On the other hand, the two players help each other get the optimal utility under their own limits respectively. It indicates that all the nodes play the constrained game.

As Player 2 includes all the other *n*-1 competing nodes except Player 1, collisions may happen among the *n*-1 competing nodes even not considering Player 1. So Player 2 includes four possible actions: *successful transmission*, *failed transmission*, *listening* or *sleeping*, even if we do not consider Player 1. Table 1 is the strategy table with 2 players (i.e., *n* nodes).

With regard to the payoff of Play 1 in a given time slot, there are four possibilities when considering the two players. Firstly, Player 1 sleeps with the probability of w_i, whose payoff is $c_{w,j}$, where *j* corresponds to the *j*-th parameter of the utility function. Secondly, Player 1 listens to the channel with the probability of $(1-w_i)(1-\tau_i)$, whose payoff is $c_{i,j}$. Here τ_i is the conditional transmission probability of Player 1. Thirdly, Player 1 fails to transmit its packets due to the collision between the two players with the probability of

Table 1. Strategy model with *n*+1 nodes.

Player 1 (node *i*)	Player 2/Opponent (all the other *n* nodes): Successful Transmission	Failed Transmission	Listening	Sleeping
Transmitting	$(c_f, \bar{c}_f)$		$(c_s, \bar{c}_i)$	$(c_f, \bar{c}_w)$
Listening	$(c_i, \bar{c}_s)$	$(c_i, \bar{c}_f)$	$(c_i, \bar{c}_i)$	$(c_i, \bar{c}_w)$
Sleeping	$(c_w, \bar{c}_f)$		$(c_w, \bar{c}_i)$	$(c_w, \bar{c}_w)$

$\tau_i(1-w_i)\left((1-\bar{w}_i)\bar{\tau}_i+\bar{w}_i\right)$, whose payoff is $c_{f,j}$. Here $\bar{w}_i$ and $\bar{\tau}_i$ are the sleeping probability and the conditional transmission probability of Player 2 respectively. Finally, Player 1 transmits successfully with the probability of $(1-w_i)\tau_i(1-\bar{w}_i)(1-\bar{\tau}_i)$, whose payoff is $c_{s,j}$.

With regard to the payoff of Player 2 in a given time slot, there are four possibilities too after considering the two players. Firstly, Player 2 sleeps with the probability of $\bar{w}_i$, whose payoff is $\bar{c}_{w,j}$. Secondly, Player 2 listens to the channel with the probability of $(1-\bar{w}_i)(1-\bar{\tau}_i)$, whose payoff is $\bar{c}_{i,j}$. Thirdly, Player 2 fails to transmit its packets due to the collisions between the two players or among the *n*-1 nodes within Player 2 with the probability of $(1-\bar{w}_i)\bar{\tau}_i\left((1-w_i)\tau_i+\bar{p}_i(1-w_i)(1-\tau_i)+w_i\right)$, whose payoff is $\bar{c}_{f,j}$. Finally, Player 2 transmits successfully with the probability of $(1-\bar{w}_i)\bar{\tau}_i(1-\bar{p}_i)(1-w_i)(1-\tau_i)$, whose payoff is $\bar{c}_{s,j}$. Here, $\bar{p}_i$ is the conditional collision probability of Player 2, which is the function of the probability $\bar{\tau}_i$ [14].

Hence, the optimal strategies of the two players under the given constraints are expressed as

$$\begin{cases} s_i^* = \arg\min\limits_{(w_i,\tau_i)} \sum\limits_j \left| 1 - \dfrac{(1-\bar{w}_i)\left((1-\bar{\tau}_i)\bar{c}_{i,j}+\bar{\tau}_i(1-\bar{p}_i)(1-w_i)(1-\tau_i)\bar{c}_{s,j}+\bar{\tau}_i\left((1-w_i)\tau_i+\bar{p}_i(1-w_i)(1-\tau_i)+w_i\right)\bar{c}_{f,j}\right)+w_i\bar{c}_{w,j}}{\bar{\mu}_{i,j}^*} \right| \left(\boldsymbol{\mu}_i < \boldsymbol{\mu}_i^{\max}\right) \\ \bar{s}_i^* = \arg\min\limits_{(\bar{w}_i,\bar{\tau}_i)} \sum\limits_j \left| 1 - \dfrac{\left((1-w_i)\left(\tau_i(1-\bar{w}_i)(1-\bar{\tau}_i)c_{s,j}+(1-\tau_i)c_{i,j}+\tau_i\left((1-\bar{w}_i)\bar{\tau}_i+\bar{w}_i\right)c_{f,j}\right)+w_i c_{w,j}\right)}{\mu_{i,j}^*} \right| \left(\bar{\boldsymbol{\mu}}_i < \bar{\boldsymbol{\mu}}_i^{\max}\right) \end{cases} \tag{2}$$

In general, the contention-based MAC protocol in WSNs is modelled as a game-theoretic constraint optimization with multiple objectives. Based on the estimated game state, each node achieves the global optima by adjusting its transmission and sleeping probability simultaneously.

3. A Simplified Game-Theoretic Constraint Optimization Scheme for WSNs

Unfortunately, the above problem has been proven to be NP-hard [15], so we cannot hope an algorithm that can find the theoretical optimum and runs in polynomial time. Hence, we present a simplified game-theoretic constraint optimization scheme (G-ConOpt) in this section. In G-ConOpt, we optimize the performance (e.g., the system throughput, delay, jitter, and packet-loss-rate) under the limited energy consumption.

In G-ConOpt, time is divided into super-frames and every super-frame has two parts: an active part and a sleeping part. During the active part, each node contends for the channel in the incompletely cooperative game. During the sleeping part, each node turns off its radio to preserve energy. The time length of the active and sleeping part is adjusted according to the estimated game state too.

In the game, firstly, a node estimates the current state of the game, e.g., the number of its opponents *n*-1. When the node is transmitting its frame, if any other node transmits at the same time slot, the frame will be collided. So the frame collision probability of the node *p* is obtained as follows:

$$p = 1-(1-\tau)^{n-1} \tag{3}$$

where τ is the frame transmission probability of the node.

If solving the above equation with respect to *n*, we obtain:

$$n = 1+\frac{\log(1-p)}{\log(1-\tau)} \tag{4}$$

Secondly, the node adjusts its equilibrium strategy, e.g., the minimum contention window (CW_{min}), to the estimated number of its opponents ($\hat{n}$), as follows [14]:

$$CW_{\min} = \left[\hat{n} \times rand(7,8)\right] \quad (5)$$

where rand (x, y) returns a random value between x and y, and [z] returns the floor function of z.

However, Vercauteren *et al* [16] showed that (4) is accurate only under saturated conditions (i.e., each node always has a packet waiting for transmission), and far from being accurate under unsaturated conditions if not filtered, e.g., for burst traffic. Bianchi and Tinnirello [17] presented two run-time estimation mechanisms, i.e., auto regressive moving average (ARMA) and Kalman Filters. The two mechanisms are very accurate even in unsaturated conditions. However, they are too complex to implement in sensor nodes.

We provided an auto degressive backoff mechanism to implement the game in current WLANs [14], which can be implemented easily in sensor nodes.

In the active part, after transmitting or discarding a packet, i.e., at the end of each game process, to maintain the current contention level, the player adjusts CW_{min} as

$$CW_{\min} = \begin{cases} \max\left(CW_{\min}, CW/2\right) & \text{The previous packet is transmitted successfully} \\ CW_{\max} & \text{The previous packet is discarded} \end{cases} \quad (6)$$

The parameter CW_{min}, CW_{max}, and CW at the right of (6) are the values of the nominal CW_{min}, CW_{max} and the final contention window used in the previous game process respectively. The parameter CW_{min} at the left of (6) is used in the current game process to transmit a new packet.

In CSMA/CA, a node starts a contention process always with the nominal CW_{min}, e.g., in IEEE 802.11b CW_{min}=32. So CSMA/CA has one main drawback: in a high load network the increase of the value of CW is obtained at the cost of continuous collision.

In G-ConOpt, after transmitting a packet, no matter it is transmitted successfully or not, the player does not start the next game process with the nominal CW_{min}, as shown in Figure 1. Given that the previous packet is transmitted successfully, the final value of CW is the optimal one. The best strategy for the player is to set $CW_{min}=CW/2$, to make use of the channel effectively. On the contrary, given that the previous packet is discarded, the best strategy for the player is to set $CW_{min}=CW_{max}$, to decrease collisions.

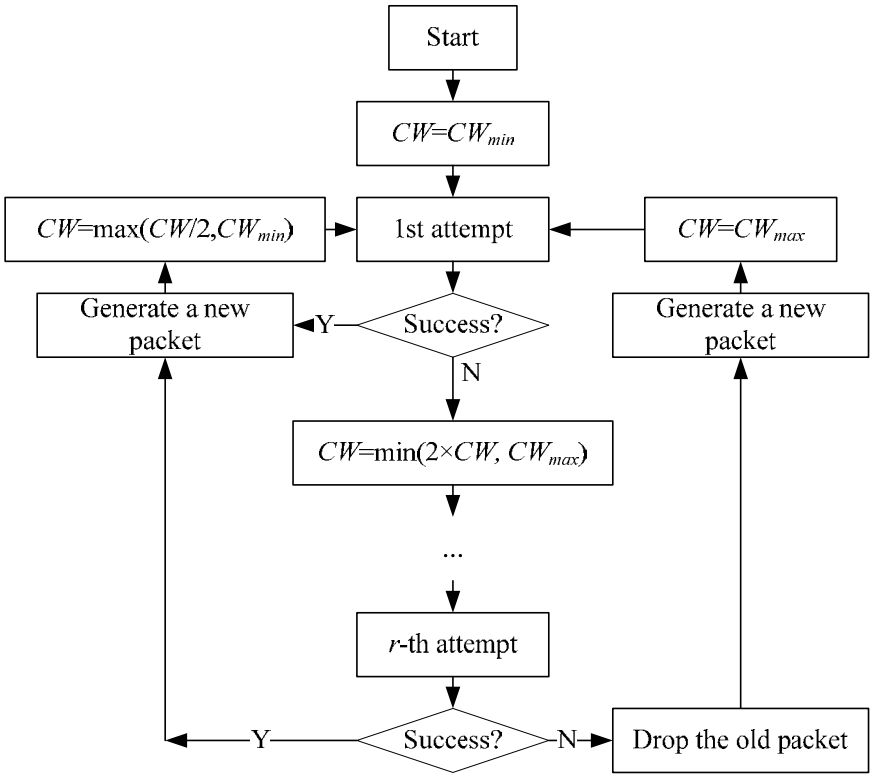


Figure 1. Auto degressive backoff mechanism.

Obviously, compared with the game, the most attractive feature of G-ConOpt is that it is simple to implement. Firstly, no estimation mechanism is needed. Secondly, it is not needed to compute the optimal value of CW_{min}. That is to say, G-ConOpt would not cause any more energy consumption.

Moreover, at the end of the active part, the node changes the length of the active part (T_{active}) and the next period (T_{next}), according to the estimated game state, as follows:

$$\begin{cases} T_{active}^{next} = \max\left(T_{active}^{current} + \alpha, T_{active,\max}\right) & T_{next} = 0.5 \times T^{current} \quad \hat{n} \geq n, T_{active}^{current} / T^{current} \leq 0.5 \\ T_{active}^{next} = \min\left(T_{active}^{current} - \alpha, T_{active,\min}\right) & T_{next} = 2 \times T^{current} \quad \hat{n} \leq n, T_{active}^{current} / T^{current} \geq 0.1 \\ T_{active}^{next} = T_{active}^{current} & T_{sleep}^{next} = T_{sleep}^{current} \quad else \end{cases} \tag{7}$$

where max(x, y) and min(x, y) return the larger value and the smaller value between x and y respectively. The parameters $T_{active}^{current}$ and $T^{current}$ are the time length of the active part and the period in the current period. $T_{active,max}$ and $T_{active,min}$, are the maximum and minimum length of the active part. T_{active}^{next} and T^{next} are used in the next period. α is a predetermined integer, n is the last estimated number of competing nodes, and $\hat{n}$ is the current estimated number of competing nodes.

At the end of the current active part, if the estimated number of competing nodes is larger than that in the last active part, it indicates many nodes still have packets to send. So the time length of the next active part equals to that of the current active part plus α but not longer than the maximum active part size. The time length of the next period is half that of the current period; thereby the nodes can wake up more frequently to reduce the delay of communication. On the other hand, if the estimated number of competing nodes is smaller than that in the last period, the time length of the next active part equals to that of the current active part minus α but not shorter than the minimum active part size. The time length of the next period is twice that of the current period, so the nodes need not wake up frequently.

4. Simulation Results

To evaluate the proposed protocol G-ConOpt, the following simulations are made in an ideal channel. The values of the parameters used to obtain numerical results for simulations are specified in IEEE 802.11b protocol, as shown in Table 2.

Table 2. Simulation parameters.

channel rate	aSlot Time	retry limit	MAC header	PHY header
1Mb/s	20μs	7	144μs	192μs
ACK	SIFS	DIFS	CW_{min}	CW_{max}
112μs	10μs	50μs	32	1024
Transmit Power	Receiving Power	Listen Power	Sleeping Power	
27.45mW	13.5mW	13.5mW	0.015mW	

The packets will be discarded only due to the retransmission time reaches the retry limit, and do not consider the delay limit. We set a star topology with one coordinator and 50 devices, where each device generates new fixed size packets under a Poisson process and transmit them to the coordinator. The packet arrival rate is initially set to be lower than the saturation case, and it is subsequently increased so that, at the end of the simulation time, all nodes are almost in saturation conditions [18].

CSMA/CA is considered as the worst case: it has no energy saving features at all. The radio of each node does not go into the sleep mode. It is either in the listening/receiving mode or transmitting mode. S-MAC is considered as the basic contention-based MAC protocol in WSNs. It includes the periodic active and sleeping time to achieve energy savings. For simplicity, the length of the active and sleeping part are fixed at 500ms in the following simulations. Compared with S-MAC, T-MAC can adapt to the load variations in time and location, and can end the active part according to the traffic loads.

Figure 2 shows that the four protocols have almost the same system throughput under light traffic loads, and under heavy traffic loads, the system throughput of G-ConOpt is a little higher than that of CSMA/CA, which is about 2 times that of S-MAC and a little higher than T-MAC.

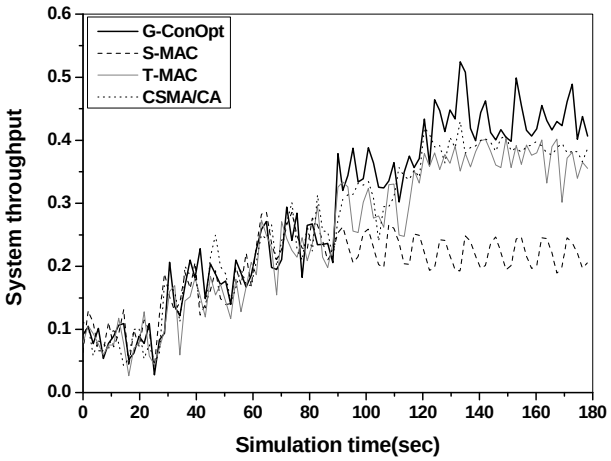


Figure 2. System throughput.

Figure 3 shows that delay in G-ConOpt, CSMA/CA and T-MAC are much lower than that in S-MAC. Under light traffic loads, delay in G-ConOpt is a little larger than that in CSMA/CA, which is due to the periodic active/ sleeping period in G-ConOpt. Under heavy traffic loads, delay in G-ConOpt is lower than that in CSMA/CA and T-MAC, which is due to the game in G-ConOpt.

Figure 4 shows that jitter in S-MAC is much higher than that in the other 3 protocols.

Figure 5 shows that packet-loss-rate in G-ConOpt almost keeps zero, which is much lower than that in S-MAC and CSMA/CA. Meanwhile, packet-loss-rate in G-ConOpt is a little lower than that in T-MAC, which is due to the game in G-ConOpt.

Figure 6 shows that the energy consumption in S-MAC is near to one half that in CSMA/CA, which is due to the periodic active/sleeping scheme. Energy consumption in T-MAC is a little lower than that in S-MAC under light traffic loads, for nodes in T-MAC sleep longer than that in S-MAC. However, energy consumption in T-MAC is larger than that in S-MAC under heavy traffic loads, since nodes in T-MAC sleep shorter than that in S-MAC. The energy consumption in G-ConOpt is the lowest one in the four protocols, which is due to the dynamic duty cycle strategy and the game in G-ConOpt.

As an energy-efficient MAC protocol, G-ConOpt considers not only energy consumption but also energy efficiency (i.e., the ratio of the successfully transmitted bit rate to energy consumption). Figure 7 shows that energy efficiency in G-ConOpt is much higher than that in S-MAC and CSMA/CA and T-MAC. As an energy-aware MAC protocol, S-MAC has higher energy efficiency than CSMA/CA under light traffic loads. However, the advantage of S-MAC over CSMA/CA decreases with the increasing of traffic loads. Under heavy traffic loads, energy efficiency in S-MAC is almost equal to that in CSMA/CA. Energy efficiency in T-MAC is always larger than that in S-MAC and T-MAC.

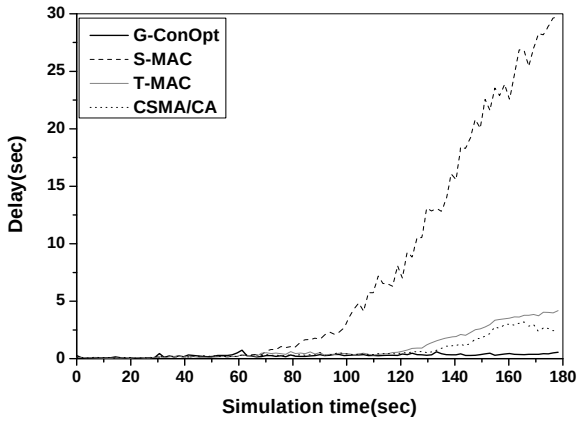


Figure 3. Delay.

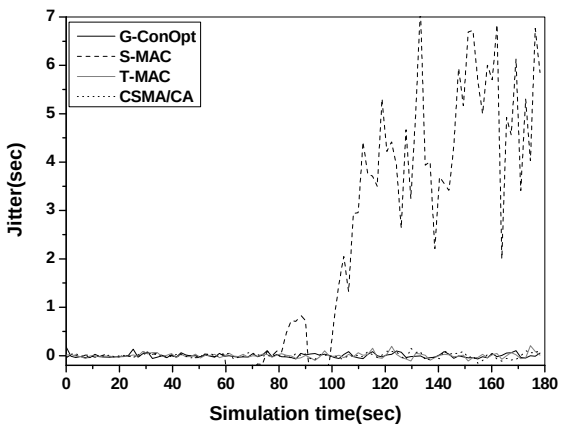


Figure 4. Jitter.

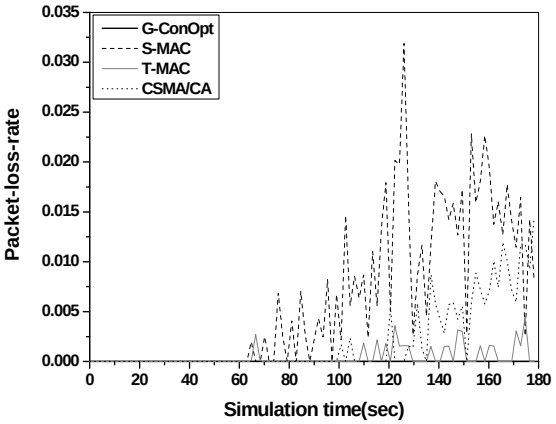


Figure 5. Packet-loss-rate.

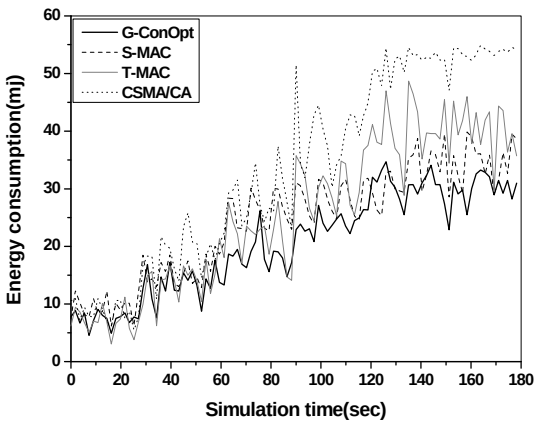


Figure 6. Energy consumption.

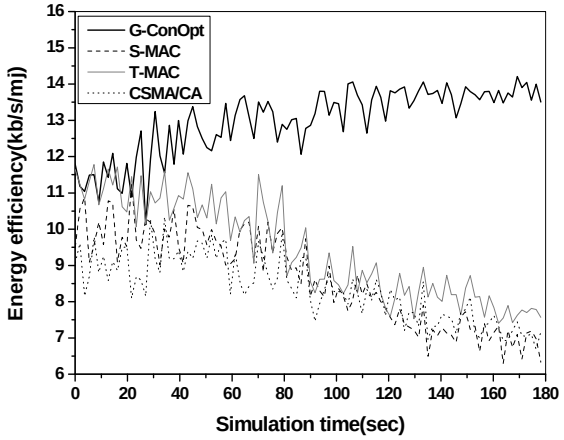


Figure 7. Energy efficiency.

5. Conclusions

In this paper, firstly, the incompletely cooperative game is used to model the MAC protocol of WSNs. Secondly, after considering the context of WSNs, e.g., the requirements on energy consumption, the problem of the MAC protocols of WSNs is modeled as a game-theoretic constraint optimization problem. Moreover, one simple formulation is presented for the problem. Finally, a simplified protocol, G-ConOpt is proposed, which can be easily implemented in current WSNs. Based on G-ConOpt, each nodes can achieve independently the optimal performance under limited energy consumption. The simulation results show that G-ConOpt is an appropriate tool to improve the performance of WSNs under certain constraints.

In this paper we only provide a simplified method to address the sleeping probability. We are developing an analytical model to obtain the optimal equilibrium of the sleeping probability.

Acknowledgement: This work is supported by the 111 Project (B08038), State Key Laboratory of Integrated Services Networks (ISN090105), Program for New Century Excellent Talents in University (NCET-08-0810), National Natural Science Foundation of China (No. 60772137), and UK-China Science Bridges: R&D on 4G Wireless Mobile Communications.

6. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, *et al.*, "Wireless sensor networks: a survey," Computer Networks, Vol. 38, No. 4, pp. 393–422, March 2002.

[2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," INFOCOM, New York, Vol. 3, pp. 1567–1576, June 2002.

[3] T. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," ACM SenSys, Los Angeles CA, November 2003.

[4] J. Polastre, J. Hill, and D. Celler, "Versatile low power media access for wireless sensor networks," ACM SenSys, USA, pp. 95–107, November 2004.

[5] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," ISCC, Egypt. pp. 244–251, June 2004.

[6] P. Lin, C. Qiao, and X. Wang, "Medium access control with dynamic duty cycle for sensor networks," WCNC, Atlanta, Georgia, March 2004.

[7] T. van Dam, K. Langendoen, "A adaptive energy-efficient MAC protocol for wireless sensor networks," ACM SenSys, USA, pp 171–180, November 2003.

[8] P. D. Straffin, "Game theory and strategy," The Mathematical Association of America, 1993.

[9] A. Agah, S. K. Das, and K. A. Basu, "Game theory based approach for security in wireless sensor networks," IPCCC, USA, pp. 259–263, April 2004.

[10] R. Kannan, S. Sarangi, and S. S. Lyengar, "Sensor-centric energy-constrained reliable query routing for wireless sensor networks," Journal of Parallel and Distributed Computing, Vol. 64, No. 7, pp. 839–852, July 2004.

[11] S. Sengupta and M. Chatterjee, "Distributed power control in sensor networks: A game theoretic approach," IWDC, India, pp. 508–519, December 2004.

[12] X. Zhang, Y. Cai, and H. Zhang, "A game-theoretic dynamic power management policy on wireless sensor network," ICCT, China, pp. 1–4, November 2006.

[13] L. Zhao, L. Guo, K. Yang, and H. Zhang, "An Energy-efficient MAC Protocol for WSNs: Game-theoretic constraint optimization," IEEE International Conference on Communication Systems, China, pp. 114–118, November 2008.

[14] L. Zhao, L. Guo, J. Zhang, and H. Zhang, "A Game-theoretic MAC protocol for wireless sensor network," Journal of IET Communications, Vol. 3, No. 8, pp. 1274–1283, August 2008.

[15] M. S. Garey and D. S. Johnson, "Computers and Intractability: Guide to the theory of NP-completeness," W. H. Freeman, New York, 1979.

[16] T. Vercauteren, A. L. Toledo, and X. Wang, "Batch and sequential bayesian estimators of the number of active terminals in an IEEE 802.11 network," IEEE Trans. on Signal Processing, Vol. 55, No. 2, pp. 437–450, January 2007.

[17] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," IEEE INFOCOM, Vol. 2, San Francisco, pp. 844–852, March 2003.

[18] G. Bianchi, "Performance Analysis of the IEEE 802.11 distributed coordination function," IEEE JSAC, Vol. 18, No. 3, pp. 535–547, March 2000.

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:

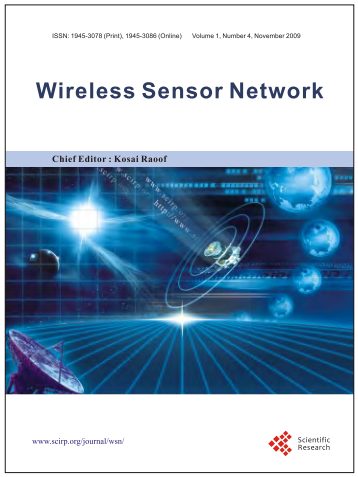
Paper Due: Feb. 28, 2010

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, 233-364
Published Online November 2009 in SciRP (http://www.SciRP.org/journal/wsn/).

TABLE OF CONTENTS

Volume 1 Number 4 **November 2009**

9771945307008 04