International Journal of

# Communications, Network and System Sciences

ISSN: 1913-3715

Volume 2, Number 5, August 2009

www.scirp.org/journal/ijcns/

Scientific Research

# JOURNAL EDITORIAL BOARD

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)
http://www.scirp.org/journal/ijcns/

## Editors-in-Chief

**Prof. Huaibei Zhou** Advanced Research Center for Sci. & Tech., Wuhan University, China
**Prof. Tom Hou** Department of Electrical and Computer Engineering, Virginia Tech., USA

## Editorial Board

**Prof. Dharma P. Agrawal** University of Cincinnati, USA
**Prof. Jong-Wha Chong** Hanyang University, Korea (South)
**Prof. Laurie Cuthbert** University of London at Queen Mary, UK
**Dr. Franca Delmastro** National Research Council, Italy
**Prof. Klaus Doppler** Nokia Research Center, Nokia Corporation, Finland
**Prof. Thorsten Herfet** Saarland University, Germany
**Dr. Li Huang** Stiching IMEC Nederland, Netherlands
**Prof. Chun Chi Lee** Shu-Te University, Taiwan (China)
**Prof. Myoung-Seob Lim** Chonbuk National University, Korea (South)
**Prof. Zhihui Lv** Fudan University, China
**Prof. Jaime Lloret Mauri** Polytechnic University of Valencia, Spain
**Dr. Lim Nguyen** University of Nebraska-Lincoln, USA
**Prof. Petar Popovski** Aalborg University, Denmark
**Dr. Kosai Raoof** University of Joseph Fourier, Grenoble, France
**Prof. Bimal Roy** Indian Statistical Institute, India
**Prof. Heung-Gyoon Ryu** Chungbuk National University, Korea (South)
**Prof. Rainer Schoenen** RWTH Aachen University, Germany
**Dr. Lingyang Song** Philips Research, Cambridge, UK
**Prof. Boris S. Verkhovsky** New Jersey Institute of Technology, USA
**Prof. Guoliang Xing** Michigan State University, USA
**Dr. Hassan Yaghoobi** Mobile Wireless Group, Intel Corporation, USA

## Editorial Assistants

**Xiaoqian QI** **Li ZHU** Wuhan University, China

## Guest Reviewers

Resul Das
Der-Rong Din
Zahir Hussain
Anjan Biswas
Xiao-Hui Lin
Yudong Zhang
X. Perramon
Hui-Kai Su
Zafer Iscan
Jing Chen
Xi Chen
Yen-Lin Chen
Burcin Ozmen
Wei-Hung Lin
Yansong Wang
K. Thilagavathi
Haitao Zhao
Nicolas Burrus
Rashid A. Saeed
Marco Castellani
Mingxin Tan
Sophia G. Petridou
Abed Ellatif Samhat
Zahir M. Hussain
Krishanthmohan Ratnam
Abed Ellatif Samhat
Luiz Henrique Alves Monteiro

*Int. J. Communications, Network and System Sciences*, **2009, 5, 325-458**
Published Online August 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

# TABLE OF CONTENTS

**Volume 2 Number 5 August 2009**



Copyright © 2009 SciRes. 

# International Journal of Communications, Network and System Sciences (IJCNS)

## Journal Information

### SUBSCRIPTIONS

The *International Journal of Communications, Network and System Sciences* (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 2 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

Copyright© 2009 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

### PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: ijcns@scirp.org

*Int. J. Communications, Network and System Sciences*, 2009, 5, 325-329
doi:10.4236/ijcns.2009.25035 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Performance Analysis of the D-STTD Communication System with AMC Scheme

**Jeonghwan LEE[1], Gilsang YOON[1], Namgil LEE[2], Sangjin RYOO[3], Cheolwoo YOU[4], Intae HWANG[1]**
[1]*School of Electronics & Computer Engineering, Chonnam National University, Gwangju, Korea*
[2]*Department of Information & Communication System, Kochang Korea Polytechnic College, Gochang, Korea*
[3]*Department of Computer Media, Hanyeong College, Yeosu, Korea*
[4]*Department of Communications Engineering, Myongji University, Seoul, Korea*
*Email*: *sjryoo@empal.com, sjryoo@hanyeong.ac.kr*
*Received April* 6, 2009; *revised May* 7, 2009; *accepted June* 10, 2009

## ABSTRACT

In this paper, we propose a Double-Space Time Transmit Diversity (D-STTD) communication system with Adaptive Modulation and Coding (AMC) scheme and analyze its performance using simulation experiments. The simulation results show that the probability of selecting a high Modulation and Coding Scheme (MCS) level increased as the Signal to Noise Ratio (SNR) improved. Furthermore, the D-STTD communication system with AMC scheme provided a more uniform throughput distribution throughout the entire SNR range compared to its counterpart which did not apply AMC scheme. Also, the maximum throughput of the D-STTD communication system with AMC scheme was twice that of a conventional AMC communication system or a Space Time Transmit Diversity (STTD) communication system with AMC scheme.

**Keywords:** AMC, MCS Level, STTD, D-STTD

## 1. Introduction

With the rapidly increasing demand for high-speed data transmission, next-generation mobile communication is expected to adopt a number of new technologies, including AMC, Orthogonal Frequency Division Multiplexing (OFDM), Multiple Input Multiple Output (MIMO), and Hybrid-Automatic Repeat Request (H-ARQ).

The AMC scheme varies the coding rate and modulation scheme according to the channel status to improve its transmission rate and has been employed in various high-speed wireless communication applications including Wireless Broadband (Wibro) Internet and High Speed Downlink Packet Access (HSDPA). And through continued technical developments, it is expected to be adopted by numerous wireless communication standards of the future. The D-STTD scheme complements the conventional STTD scheme in terms of throughput and its use is being widely discussed regarding various wireless communication standards.

## 2. AMC Scheme and D-STTD Scheme

The AMC scheme selects an optimal channel coding rate and modulation scheme based on the channel response data for signal transmission, creating a balance between error rate and throughput to improve the overall system throughput and transmission quality. Channel status has a significant effect on signal transmission and reception in the wireless mobile communication environment. Accordingly, a process referred to as link adaptation modifies transmission parameters to compensate for the losses caused by the changes in channel status. The AMC scheme is a type of link adaptation, and its structure is shown in Figure 1 [1−3].

Under a typical wireless communication environment, data is transmitted based on the processes of channel coding, interleaving, and modulation. Signal received via the communication channel is subject to the process of estimating the channel status according to SNR, and the original data is recovered through the reverse process in

Copyright © 2009 SciRes. 

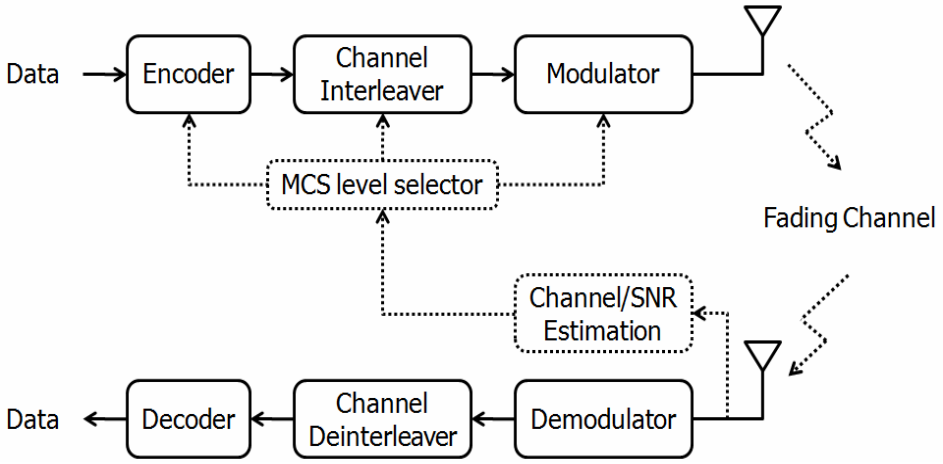



**Figure 1. Structure of AMC scheme.**

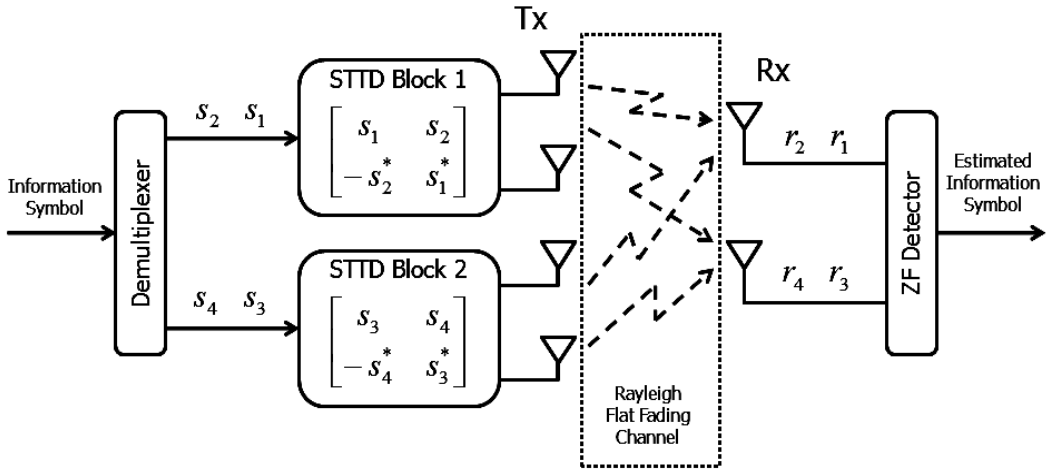



**Figure 2. Structure of D-STTD scheme.**

the transmitter. The estimated channel information is returned to the transmitter, which determines the MCS level based on this information, adjusts channel coding, interleaving, and the modulation scheme suitable for the channel status to carry out transmission. Like above, the AMC scheme selects the MCS level according to the channel status to find a balance between error rate and throughput to help improve the overall system throughput and transmission quality.

The D-STTD scheme is a type of MIMO scheme and expands from the STTD scheme proposed by Alamouti [4–6]. Whereas conventional STTD scheme only provides diversity gain, D-STTD scheme offers multiplexing gain in addition to diversity gain, making it a very effective MIMO scheme. The overall transmission pattern of the D-STTD scheme is shown in Figure 2. The received signal of the D-STTD scheme can be expressed in a simple matrix equation as follows:

$$\begin{bmatrix} r_1 \\ r_2^* \\ r_3 \\ r_4^* \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{12}^* & -h_{11}^* & h_{14}^* & -h_{13}^* \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{22}^* & -h_{21}^* & h_{24}^* & -h_{23}^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} \quad \textbf{(1)}$$

where $r$ denotes the received signal and $h$, $s$, and $n$ represent channel response, transmitted signal, and noise, respectively. Channel response $h_{ij}$ denotes a channel response between $j$-th transmit antenna and $i$-th receive antenna. Each channel response is independent and identically distributed (i.i.d.) and displays a complex Gaussian distribution with an average of zero. Noise $n$ is an Additive White Gaussian Noise (AWGN) with an average of zero and a distribution of $\sigma^2$I.

From Equation (1) and Figure 2, it can be verified that the D-STTD scheme transmits four symbols during two time slots, which corresponds to twice the throughput

Copyright © 2009 SciRes. 

and the multiplexing gain of the STTD scheme.

## 3. D-STTD Communication System with AMC Scheme

As a conventional D-STTD communication system uses a fixed channel coding rate and modulation scheme regardless of the channel response status, it can only select and use a low channel coding rate and a low-level modulation scheme in an environment that does not offer a certain level of SNR. On the other hand, a D-STTD communication system with AMC scheme selects a MCS level based on SNR information to flexibly apply the channel coding rate and the modulation scheme according to the channel response status, securing an optimal throughput at any given SNR.

Figure 3 shows the structure of a D-STTD communication system with AMC scheme. It can be seen that AMC scheme is incorporated into the D-STTD scheme explained in Chapter 2 to complement the system structure. The initially created data undergoes coding and interleaving processes based on the channel selected by the MCS level selector and is modulated with the modulation scheme also selected by the MCS level selector. The modulated signal is encoded by the D-STTD encoder for transmission over the channel and converted to an estimation of the original signal by the D-STTD decoder at the receiver. The signal then goes through the reverse process of the transmitter, and the original signal is restored.

## 4. Simulation Results

This chapter comparatively analyzes throughput of a conventional system and the D-STTD communication system with AMC scheme. Tables 1 and 2 list the MCS level selection criteria and the environmental parameters for the simulation experiment, which were prepared according to HSDPA and 3G Long Term Evolution (LTE) standards [7−9].

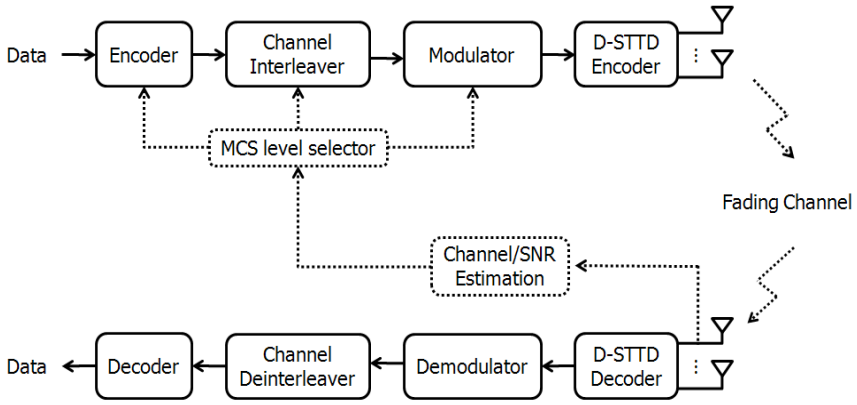



**Figure 3. Structure of D-STTD communication system with AMC scheme.**

**Table 1. MCS levels.**

| MCS level | Date Rate (kbps) | Number of bits per frame | Code Rate | Modulation | Throughput (15 Codes) |
|---|---|---|---|---|---|
| 1 | 180 | 1,800 | 1/3 | QPSK | 2.7 Mbps |
| 2 | 360 | 3,600 | 1/2 | QPSK | 5.4 Mbps |
| 3 | 536 | 5,360 | 1/2 | 16QAM | 8.0 Mbps |
| 4 | 720 | 7,200 | 1/2 | 64QAM | 10.8 Mbps |

**Table 2. Simulation Parameters.**

| Parameter | Value |
|---|---|
| Modulation Scheme | QPSK, 16QAM, 64QAM |
| Channel Coding Scheme | Turbo Code |
| Coding Rate | 1/3, 1/2 |
| Number of Transmit Antenna | 4, 2, 1 |
| Number of Receive Antenna | 2, 1 |
| Channel Environment | Rayleigh Flat Fading |
| Detection Algorithm | Zero Forcing (ZF) |

Copyright © 2009 SciRes. 

Figure 4 shows the throughput of a D-STTD 4×2 communication system for a fixed MCS level in the Rayleigh Flat Fading channel. As mentioned in Chapter 3, data is transmitted without loss for MCS level 1 (QPSK, Turbo Code 1/3) with the low SNR of roughly 2dB. However, at MCS level 4 (64QAM, Turbo Code 1/2), the data is transmitted without loss when SNR is 18dB or higher. If we assume that AMC scheme is not applied, a single coding rate and a specific modulation scheme must be selected, the coding rate and the modulation scheme that correspond to MCS levels 3 or 4 are inevitably excluded from the outset because they both have high probabilities of frame errors in low SNR range.

Figure 5 shows the MCS level selection probability of a D-STTD 4×2 communication system with the AMC scheme under the Rayleigh Flat Fading channel converted to an overall probability of 1. A low MCS level is selected for a low SNR range, and as the channel status improved along with the SNR, the probability of selecting a higher MCS level increased.

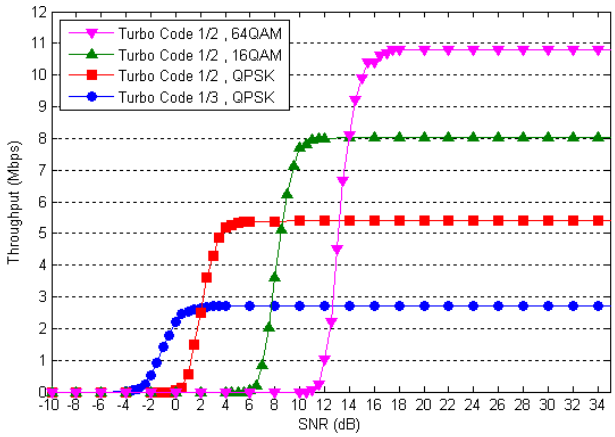



**Figure 4. Throughput of a D-STTD 4×2 communication system for a fixed MCS level in the Rayleigh Flat Fading channel.**

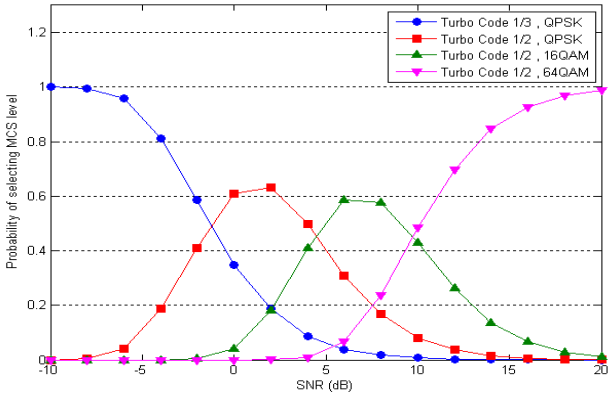



**Figure 5. Probability of MCS level selection in a D-STTD 4×2 communication system with AMC scheme.**

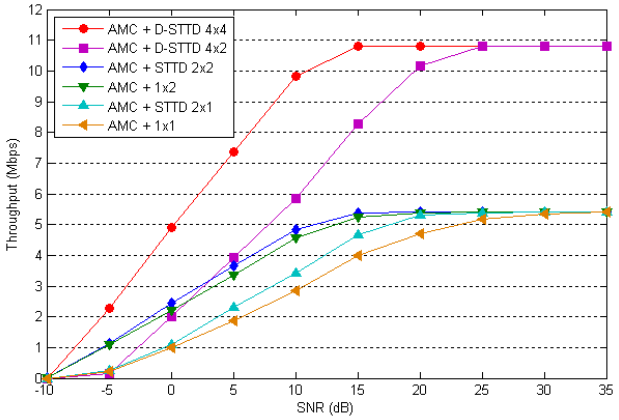



**Figure 6. Throughput of each system in the Rayleigh Flat Fading channel.**

Figure 6 indicates the throughput performance of each communication system in the Rayleigh Flat Fading channel. The conventional communication system applying only the AMC scheme showed the lowest throughput. Although the average throughput of the STTD 2×1 communication system with AMC scheme was somewhat improved, its maximum throughput was identical to that of the conventional system with AMC scheme. The average throughput of the STTD 2×2 communication system with AMC scheme was improved in the whole SNR ranges due to the receive diversity effect.

It was confirmed that the maximum throughput of the D-STTD 4×2 communication system with AMC scheme was twice that of the system that only applied AMC scheme or the STTD communication system with AMC scheme, which is the result of the multiplexing gain of D-STTD scheme explained in Chapter 2. Moreover, com- parison of Figures 4 and 6 confirms that the D-STTD 4×2 communication system with AMC scheme shows a more uniform throughput over the entire SNR ranges than the D-STTD 4×2 communication system that uses a fixed coding rate and modulation scheme without AMC scheme.

In the case of the D-STTD 4×4 communication system with AMC scheme, the average throughput improved because the receive diversity effect was added to the D-STTD 4×2 system with AMC scheme.

## 5. Conclusions

This paper proposed a D-STTD communication system with AMC scheme, examined its structure and characteristics and comparatively analyzed the performances. The results of the simulation experiments indicated that the proposed system yielded improved throughput in most SNR ranges from the conventional AMC communication system as well as the STTD communication system with AMC scheme. The maximum throughput improved by

Copyright © 2009 SciRes. 

100%. Moreover, the benefits of the AMC scheme provided a more uniform throughput throughout the SNR range as compared to using a single coding rate and a single modulation scheme. Possible topics of future studies include incorporating several detection techniques, such as asymmetric modulation and antenna shuffling, to the D-STTD communication system with AMC scheme to secure better throughput and transmission quality throughout the whole SNR ranges.

## 6. Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOE-HRD, Basic Research Promotion Fund) (KRF-2008-331-D00374).

## 7. References

[1] B. Vucetic, "An adaptive coding scheme for time-varying channels," IEEE Transactions on Communications, Vol. 39, pp. 653−663, May 1991.

[2] A. J. Goldsmith and S. G. Chua, "Variable-rate variable-power MQAM for fading channels," IEEE Transactions on Communications, Vol. 45, No. 10, pp. 1218−1230, October 1997.

[3] A. J. Goldsmith and S. G. Chua, "Adaptive coded modulation for fading channels," IEEE Transactions on Communications, Vol. 46, No. 5, pp. 595−602, May 1998.

[4] S. M. Alamouti, "A simple diversity technique for wireless communications," IEEE Journal on Selected Areas in Communications, Vol. 16, No. 8, pp. 1451−1458, October 1998.

[5] Texas Instruments, "Double-STTD scheme for HSDPA systems with four transmit antennas: Link level simulation results," TSG-R WG1 document, TSGR1#20(01)04 58, Busan, Korea, May 21−24, 2001.

[6] H. J. Lee, E. J. Powers, and J. Kang, "Low-complexity ZF detector for D-STTD systems in time-selective fading channels," Vehicular Technology Conference, IEEE 62nd, pp. 2043−2047, September 25−28, 2005.

[7] 3GPP TS 25.212 TSG RAN Multiplexing and channel coding (FDD).

[8] 3GPP TS 25.222 TSG RAN Multiplexing and channel coding (TDD).

[9] 3GPP TS 25.944 TSG RAN Channel coding and multiplexing examples.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 330-343
doi:10.4236/ijcns.2009.25036 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# TDTL Based Frequency Synthesizers with Auto Sensing Technique

**Mahmoud AL-QUTAYRI, Saleh AL-ARAJI, Abdulrahman AL-HUMAIDAN**
*College of Engineering, Khalifa University of Science, Technology and Research, Sharjah Campus, Sharjah, UAE*
*Email*: *mqutayri@kustar.ac.ae*
*Received March* 7, 2009; *revised April* 12, 2009; *accepted May* 20, 2009

## ABSTRACT

This paper presents a frequency synthesizer architecture based on the time delay digital tanlock loop (TDTL). The loop is of the first order type. The synthesizer architecture includes an adaptation mechanism to keep the complete system in lock. The mechanism uses a frequency sensing structure to control critical TDTL parameters responsible for locking. Both integer and fractional multiples of the loop reference frequency are synthesized by the new architecture. The ability of the TDTL based frequency synthesizer to respond to sudden variations in the system input frequency is studied. The results obtained indicate the proposed synthesizer has a robust performance and is capable of responding to those changes provided that they are within the bounds of its locking region.

**Keywords:** Time-Delay Tanlock Loop, Frequency Synthesizer, Phase Lock Loop, Indirect Synthesis

## 1. Introduction

The explosion in the growth of communication systems in general and wireless ones in particular is placing highly demanding requirements on the designers of such systems. Modern wireless communication systems have to support high data rates with low power consumption, high signal to noise ratios and compact designs. In addition to this, wireless systems must be able to deal with different communication standards operating at various frequencies [1−3]. The frequency synthesizer is one of the fundamental components of any wireless transceivers. As shown in Figure 1, the frequency synthesizer is basically a device that generates any number of operating frequencies ($F_{out}$) from a stable input reference frequency ($F_{ref}$) under the control of a command word or signal ($f_c$).

In wireless communication system transceivers frequency synthesizers are used as the local oscillators to generate the periodic signals required for the up-conversion as well as down-conversion processes [4−7]. Figure 2 shows the generic architecture of a wireless RF (radio frequency) transceiver with the frequency synthesizer block.

Depending on the process of frequency generation, frequency synthesizers are classified as direct or indirect [7,8]. Direct frequency synthesis can be divided into coherent and incoherent types. In the latter, synthesis is achieved by using fixed oscillators in addition to a group of frequency dividers and multipliers in order to generate

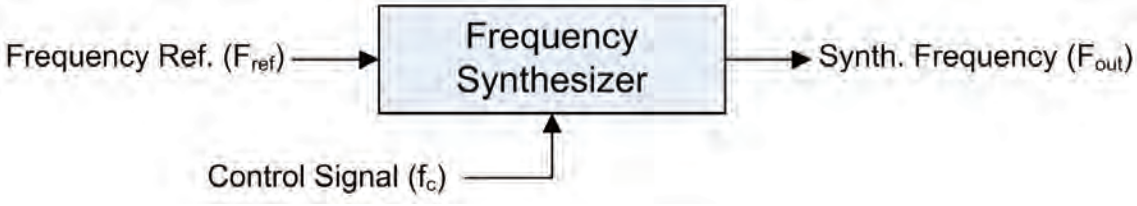



**Figure 1. Frequency synthesis process.**

Copyright © 2009 SciRes. 

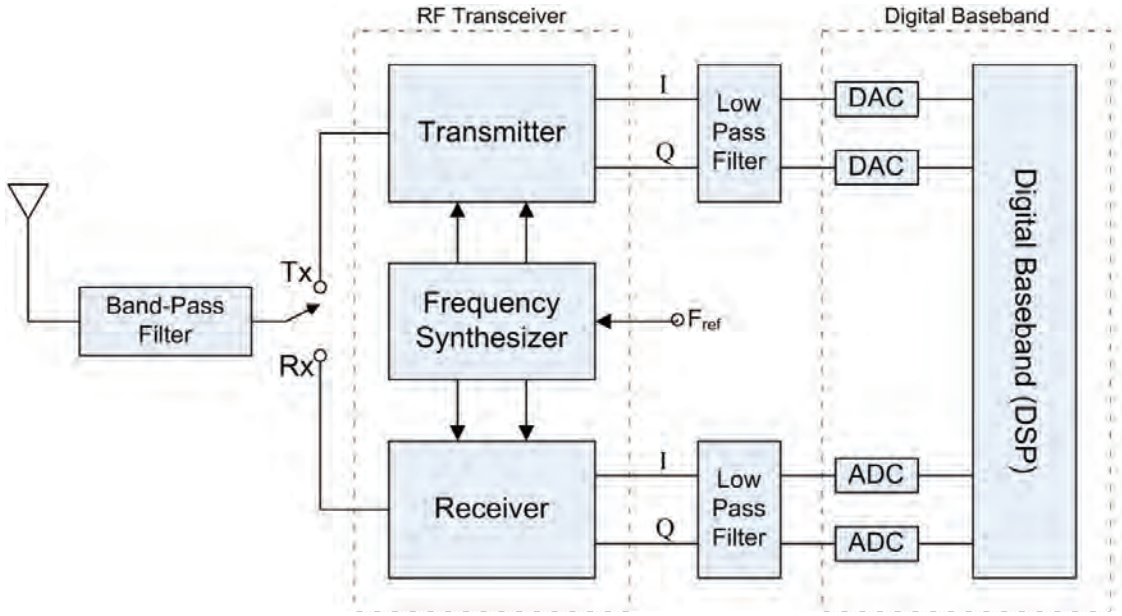



**Figure 2. Basic transceiver architecture.**

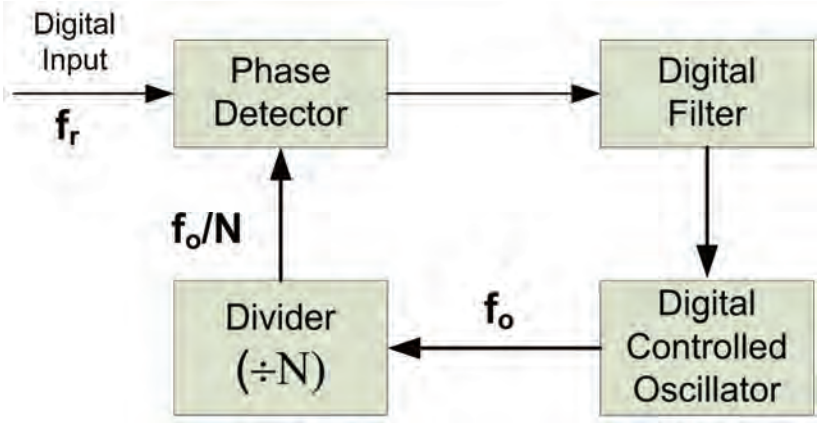



**Figure 3. Basic indirect frequency synthesizer.**

a variety of frequencies. In the coherent technique, however, only one reference source is used to generate the required output frequencies. In this approach, the stability and accuracy of the output frequencies are the same as that of the reference source. One example of the coherent technique is the phase accumulation synthesizer. The main advantages of the direct synthesis are the fast settling, fine step sizes, and simple implementation. The disadvantages of this technique are bandwidth limitations and undesired spurious harmonic generation. These problems pose severe limitations on future generation wireless communications such as 4G where OFDM is employed [7−10].

The second approach in frequency synthesis is the indirect frequency synthesis. The theory behind this type of synthesis is that it uses an unstable oscillator; the oscillator is then forced to be stabilized by feeding back a portion of the received (original) signal. The feedback is used to stabilize the loop as well as provide enough information in order to reconstruct the signal by generating a replica of the carrier [5,7,11−13]. The advantages of using such technique are the generation of spurious free output signals, the wideband system operation, and the less stringent requirement on the local oscillator frequency stability. The main limitations are the settling time and phase noise which can be minimized by using additional circuitry embedded in the loop. Two approaches of the indirect synthesis are the analog PLL synthesizer and the digital PLL synthesizer. The basic block diagram of an indirect frequency synthesizer is depicted in Figure 3.

Copyright © 2009 SciRes. 

This paper uses a novel type of digital phase lock loops called the time-delay digital tanlock loop (TDTL) as the locking element of the indirect frequency synthesizer system. The TDTL is a high performance phase locking system that can be implemented efficiently as all its components are of the digital type [14].

The paper is organized as follows. Section 2 presents the architecture and system equations of the time-delay digital tanlock loop. This also includes the locking range of the TDTL which is a major aspect of a phase lock loop. Section 3 discusses the process and the challenges of using the TDTL as a frequency synthesizer. The adaptation process introduced to enable the TDTL based frequency synthesizer to achieve locking for both integer and fractional frequency division is also detailed in Section 3. The results of the new TDTL based frequency synthesizer for various division factors and under different input conditions are presented in Section 4. The conclusions of this work are presented in Section 5.

## 2. Time-Delay Digital Tanlock Loop

The structure of the TDTL is shown in Figure 4. It consists mainly of two sample and hold blocks, a phase detector, a low pass filter, a digitally controlled oscillator, and a time-delay block. Being comprised of these components, the TDTL lends itself for implementation in various digital systems technologies. The TDTL offers an inexpensive implementation and improved performance compared with other synchronization techniques. Compared with the conventional digital tanlock loop in [15], the TDTL in Figure 4 does not preserve the linearity of the phase characteristics. This is due to the fact that the delay component of the loop is frequency dependent and hence causes the phase shift of the input signal to deviate somewhat from the original designed value over the locking range. Consequently an element of nonlinearity is introduced into the locking range. However, this supposed disadvantaged can basically be ignored because the TDTL preserves the wide locking range and fast acquisition behavior advantages of the conventional digital tanlock loop and uses a system architecture that is easier to implement in hardware compared to the conventional type. An in depth comparison of the conventional digital tanlock loop and TDTL with extensive results and discussion is given in [16]. The mathematical analysis of the TDTL under noise free conditions is detailed below. All of the signal notations are chosen in reference to the block diagram shown in Figure 4. The analysis follows a similar line to that given in [14,17,18].

The TDTL receives a continuous time sinusoid *y(t)* which is given by (1).

$$y(t) = A\sin\left[\omega_o t + \theta(t)\right] + n(t) \quad \textbf{(1)}$$

where A is the amplitude of the signal, $\omega_0$ is the free running frequency of the DCO, $\theta(t) = (\omega - \omega_0)t + \theta_0$ is the information-bearing phase and $n(t)$ is the additive white Gaussian noise (AWGN). The signal is assumed not to have a DC component. Usually the phase process $\theta(t)$ is a translation of frequency or phase steps. $\omega$ is the radian frequency of the input signal and $\theta_0$ is a constant. A phase lag $\psi = \omega\tau$ is induced to the input signal after it passes through the time delay block. Therefore, *x(t)* is generated, which is a phase shifted version of the input signal *y(t)*, this signal is given by (2).

$$x(t) = A\sin\left[\omega_o t + \theta(t) - \Psi\right] + n'(t) \quad \textbf{(2)}$$

where $n'(t)$ is the time-delayed AWGN due to $\tau$. The aforementioned continuous time signals pass to the sample and hold blocks, and thereby get transformed to the discrete time signals in (3) and (4).

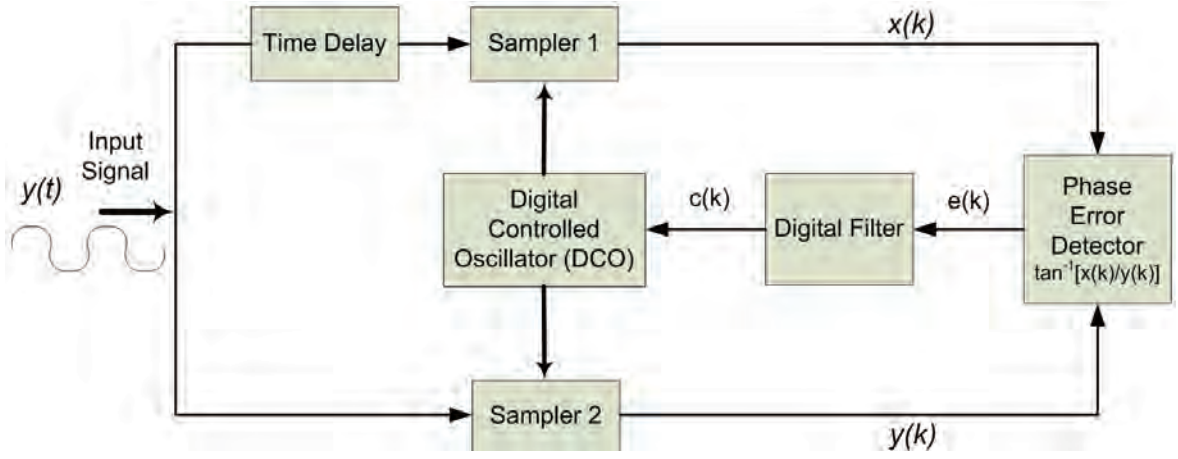



**Figure 4. Architecture of time-delay digital tanlock loop.**

Copyright © 2009 SciRes. 

$$y(k) = A\sin\left[\omega_o t(k) + \theta(k)\right] + n(k) \quad \textbf{(3)}$$

$$x(k) = A\sin\left[\omega_o t(k) + \theta(k) - \Psi\right] + n'(k) \quad \textbf{(4)}$$

where $\theta(k) = \theta[t(k)]$.

The sampling interval between the sampling instants $t(k)$ and $t(k-1)$ is given by (5).

$$\mathrm{T}(k) = T_0 - c(k-1) \quad \textbf{(5)}$$

where $T_0 = 2\pi/\omega_0$ is the nominal period of the DCO and $c(i)$ is the output of the digital filter at the i[th] sampling instant. Assuming $t(0) = 0$, the total time $t(k)$ elapsed up to the $k^{\text{th}}$ sampling instant is given by (6).

$$t(k) = \sum_{i=1}^{k} T(i) = kT_o - \sum_{i=0}^{k-1} c(i) \quad \textbf{(6)}$$

$$y(k) = A\sin\left[\theta(k) - \omega_o \sum_{i=0}^{k-1} c(i)\right] + n(k) \quad \textbf{(7)}$$

$$x(k) = A\sin\left[\theta(k) - \omega_o \sum_{i=0}^{k-1} c(i) - \Psi\right] + n'(k) \quad \textbf{(8)}$$

And therefore, the phase error between the input signal and the DCO can be also defined as:

$$\phi(k) = \theta(k) - \omega_o \sum_{i=0}^{k-1} c(i) - \Psi \quad \textbf{(9)}$$

Having defined the phase error, Equations (7) and (8) can be rewritten as

$$y(k) = A\sin\left[\phi(k) + \Psi\right] + n'(k) \quad \textbf{(10)}$$

$$x(k) = A\sin\left[\phi(k)\right] + n'(k) \quad \textbf{(11)}$$

These signals are applied to the phase detector producing the error signal $e(k)$ given in (12).

$$e(k) = f\left[\tan^{-1}\left(\frac{\sin[\phi(k)]}{\sin[\phi(k)+\Psi]}\right)\right] + \zeta(k) \quad \textbf{(12)}$$

where $f(\gamma) = -\pi + \left[(\gamma + \pi) \bmod 2\pi\right]$, $\zeta(k)$ is a random phase disturbance due to AWGN. The error signal $e(k)$ represents a nonlinearly mapped version of the phase error. However, the effect of the nonlinearity is minimum and *e(k)* can be approximately linear if $\Psi$ is equal to, or in the vicinity of $\pi/2$. This relatively small effect of nonlinearity is illustrated in Figure 5 by the plot for $\Psi_0=\pi/2$, which can be reconstructed as a pricewise linear one. The digital filter, which has a transfer function given by $D(z)$ receives the error signal *e(k)* and produces the signal $c(k)$ that drives the DCO. Therefore, the system difference equation can be derived from (6) and (9) as

$$\phi(k+1) = \phi(k) - \omega c(k) + \Lambda_o \quad \textbf{(13)}$$

where $\Lambda_0 = 2\pi(\omega - \omega_0)/\omega_0$, and the AWGN terms are neglected since noise-free analysis is assumed. In the case of the conventional digital tanlock loop, the linear characteristic function of the phase detector enables the description of the loop as a linear difference equation, and hence finding the lock range using the stability criterions of its Z-transformed transfer function [15]. However, the nonlinear characteristic function of the TDTL phase detector results in a nonlinear difference equation, which can only be solved by numerical analysis. The lock range of the TDTL was analyzed in [14], using fixed-point theorems, which have been utilized previously in the analysis of the lock range of sinusoidal ZC-DPLLs [19,20]. The digital filter of the first order loop is simply a gain block $G_1$, and the system equation is given by

$$\phi(k+1) = \phi(k) - K_1' h\left[\phi(k)\right] + \Lambda_o \quad \textbf{(14)}$$

where $K_1' = \omega G_1$. Defining $K_1$ as $\omega_o G_1$ will result in $K_1' = K_1/W$, where $W = \omega_o/\omega$. The nominal phase lag $\Psi_o$ induced by the time delay units on the input can be initially arranged by manipulating the parameters $\omega_o$ and $\tau$ in the manner given by $\Psi_o = \omega\tau_o$. Therefore, the locking range can be acquired by numerically solving the inequality

$$2|1-W| < K_1 < 2W\frac{\sin^2(\alpha) + \sin^2(\alpha + \Psi_o)}{\sin(\Psi_o)} \quad \textbf{(15)}$$

where $\alpha = \tan^{-1}(\beta)$,

$$\beta = \frac{\sin(\Psi)\tan(\eta)}{1-\cos(\Psi)\tan(\eta)} = \frac{\sin(\Psi)}{\cot(\eta)-\cos(\Psi)} \quad \text{and} \quad \eta = \frac{\Lambda_o}{K_1'}$$

One of the properties of the first order TDTL is that it converges to a nonzero steady state phase error, which is translated with a phase offset between the pulses of the DCO and the zero crossings of the input signal. The steady-state value of the phase error is given by $\phi_{ss} = \sigma + j\pi$ where $j \in \{1,0,-1\}$. Figure 5 shows the locking range of the first-order TDTL for different values of $\Psi_o$ as well as the conventional digital tanlock loop locking region [14].

The range of independent locking of the TDTL and the effect of initial phase error are studied in depth in [14,21].

Copyright © 2009 SciRes. 

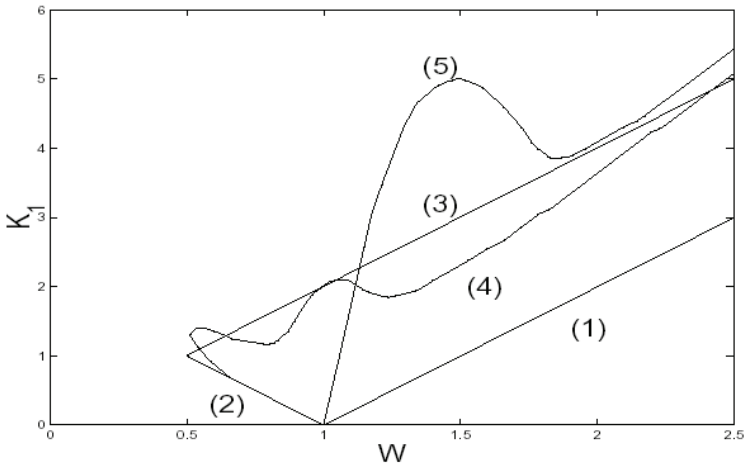



**Figure 5. Major locking range of the 1st TDTL for different values of $\Psi_0=\omega_0\tau$. Note: the region enclosed by (1), (2) and (3) is for the conventional digital tanlock loop; the region enclosed by (1), (2) and (4) is for the TDTL when $\Psi_0=\pi/2$; and the region enclosed by (1) and (5) is for the TDTL when $\Psi_0=\pi$.**

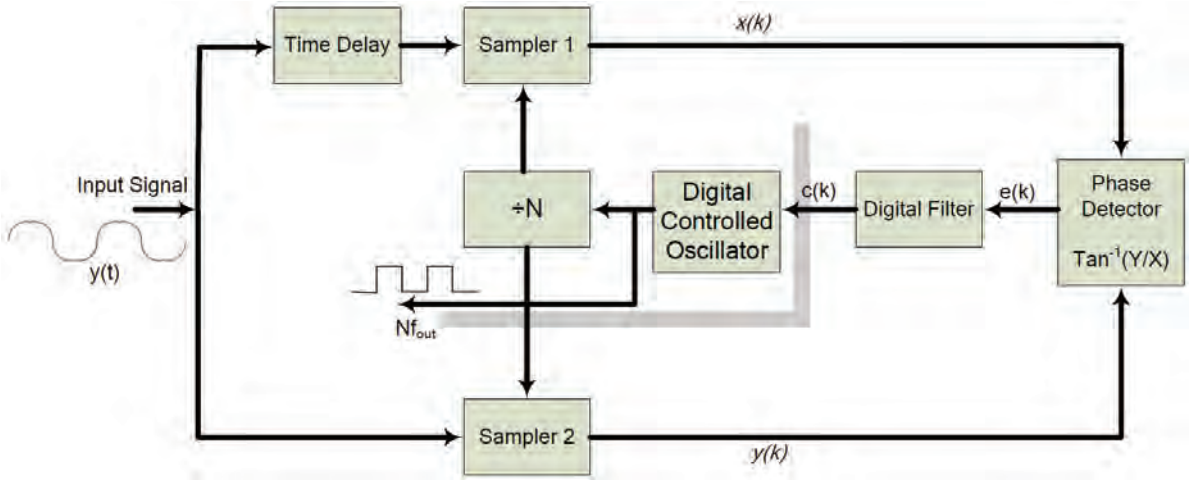



**Figure 6. Structure of the TDTL frequency synthesizer.**

Since the TDTL has non-linear characteristic function numerical analysis were used to determine the range of independent locking. The analysis shows that the TDTL offers an advantage on the conventional digital tanlock loop in this regard.

## 3. TDTL Frequency Synthesizer

As shown in Figure 6, the TDTL based Synthesizer is mainly comprised of two subsystems; namely the first order TDTL and the frequency divider. The two subsystems are configured to form an indirect frequency synthesizer. The divider generates various frequencies which are multiple of the DCO frequency, while the TDTL locks and stabilizes the synthesized frequency. This section gives a general overview of the synthesizer system that includes the operation of the TDTL and the divider as a frequency synthesizer block. Depending on the desired division factor, the synthesizer may be classified as an integer one or a fractional one. Both classes are discussed below.

### 3.1. Integer TDTL-FS

In this paper the TDTL is utilized to act as an indirect frequency synthesizer as depicted in Figure 6. This is so because the different frequencies that are produced in loop or in other words the frequency division operation are occurring within the feedback path of the loop. To achieve frequency division, a counter is introduced into the loop. This counter is placed between the DCO and the Sample and Hold blocks. Therefore, now the sam-

Copyright © 2009 SciRes. 

pling frequency of the loop is not the frequency of the DCO but the output of the divider or the counter.

The operation of frequency division is simple in which a counter is used that is clocked with the input signal. The counter has a limited number of states or counts that are equal to the division factor that is required. For example, in a divide by 2 cases the counter has 2 states (i.e. 2 counts) and produces an output when it reaches the final state before it resets i.e. it will count from 0 to 1 and will produce an output while it is residing at 1. In other words, the counter will produce 1 pulse for every 2 pulses in the case of a divide by 2 counters. Generally speaking, a divide by N counter will produce 1 pulse as an output for every N counts or input pulses, hence divide by N.

Following the introduction of the divider into the system, it was observed that the system response has changed, in which, for the same loop conditions, the system would go out of lock when the divider is introduced and would stabilize and lock if the divider is removed. Analysis of the system and associated problems indicated that the introduction of the divider into the system has the effect of moving the operating point of the system outside the locking range of the system. The reason for this is that the sampling frequency of the S/H blocks has decreased. Therefore, to compensate for this, the DCO frequency was multiplied by the same factor (N) in order to achieve the original sampling frequency. Another interesting point that was found is that in order to get the system back into its exact original operating point, not only the DCO frequency has to be adjusted but also the error signal that is fed to the DCO from the phase error detector had to be adjusted too.

The divider block has the effect of dividing the frequency of the DCO output which is considered to be a result of two factors, the DCO running frequency and the error signal. Therefore, when the output of the DCO is divided not only the frequency is divided but also the effect of the error signal that feeds the DCO and its weight (original value) in the system. As a result, both the DCO running frequency and the error signal have to be conditioned and adjusted when the divider is introduced into the system. The adjustments are done by multiplying the error signal and the DCO running frequency by the same division factor in order to get the system back into its original operating point. For example, as illustrated in Figure 7, the TDTL was configured to operate at point A within the locking range, where $W=\omega_0/\omega=1$ and $K=G\omega_0=1$, prior to the introduction of the divider.

After introducing the divider (÷N) and depending on the division factor, the operating point may shift outside the locking range towards point B, Figure 7. This is because the error signal effect will be divided by N and hence the output of the gain block will be divided by N as well as the DCO frequency and the new operating point will be at $W'=\omega_0/N\omega=1/N$ and $K'=G\omega_0/N^2=1/N^2$. So, in order to get the system back in lock, the DCO frequency $\omega_0$ and the Gain G are multiplied by the same division factor and this will bring the system back into its original state.

As stated above, the TDTL synthesizer requires some adjustments after introducing the divider in order to keep the loop locked in its basic original operating point. These adjustments are basically adjusting the DCO frequency and the error signal. In order to make this process as flexible as possible, an adaptation mechanism is introduced in the system. The mechanism basically cancels the effect of the division on the system stability through adjustment of some system parameters. This mechanism

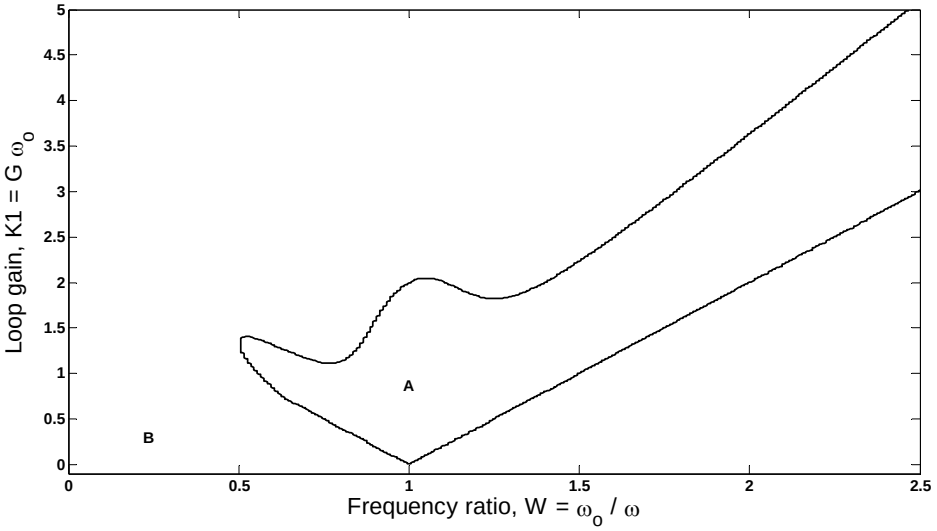
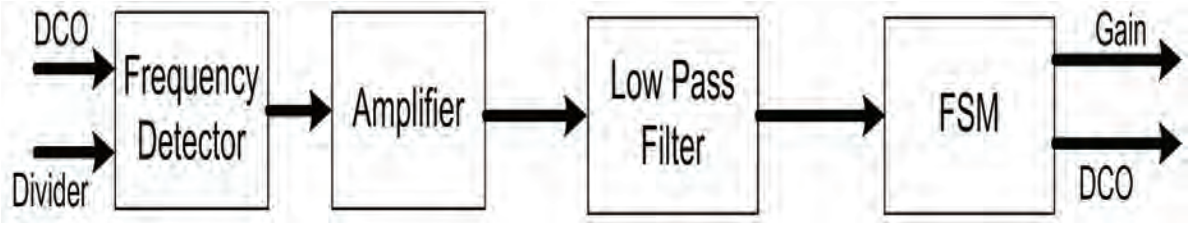
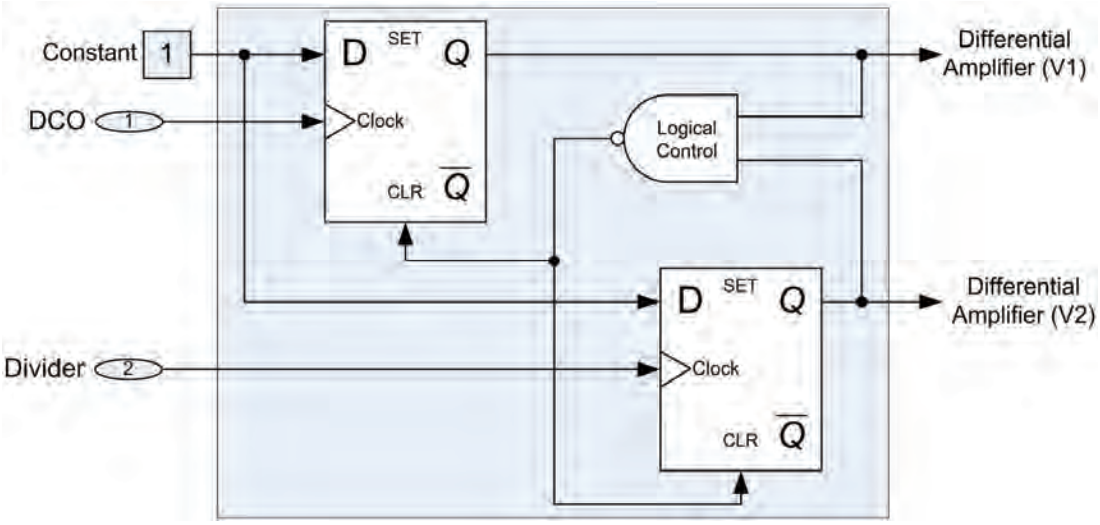



**Figure 7. Locking range of first-order TDTL.**

Copyright © 2009 SciRes. 




**(a)**




**(b)**

**Figure 8. (a) Block diagram of frequency sensing block (b) Structure of the frequency detector.**

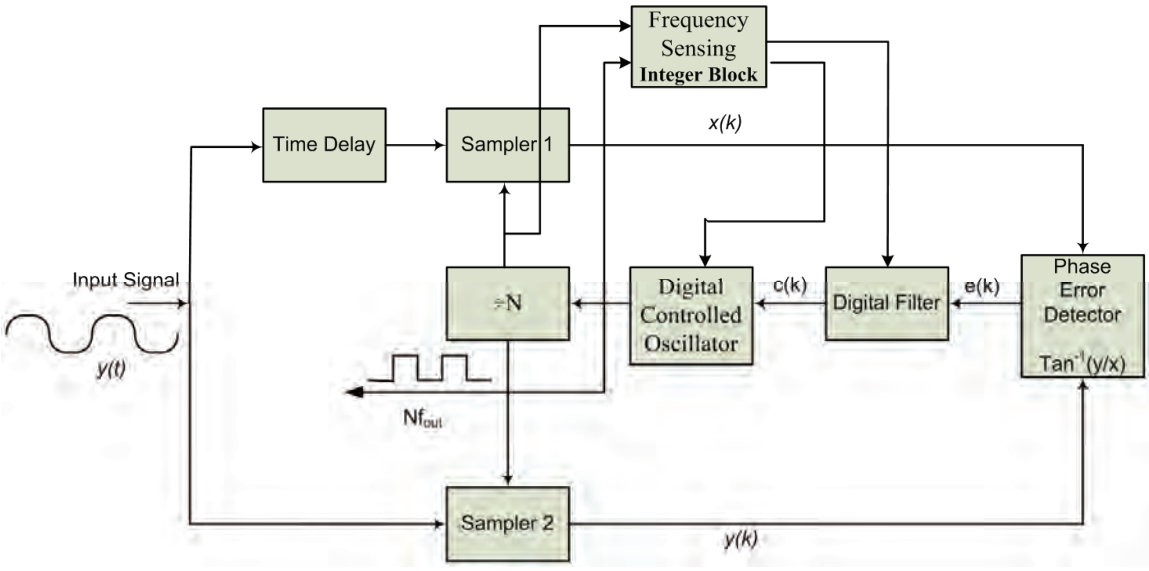



**Figure 9. Structure of integer TDTL-FS with FSB.**

is mainly composed of a frequency sensing block (FSB). The frequency sensing block consists of a frequency detector, a filter block, a differential amplifier and a finite state machine (FSM). The structure of the FSB and the frequency detector are shown in Figure 8a and Figure 8b respectively. The integer TDTL-FS incorporating the

Copyright © 2009 SciRes. 

adaptive FSB to keep it stable is depicted in Figure 9.

The basic functionality of the FSM block of the FSB is to compare between the two frequencies at the input of the divider and its output and produce an output (train of pulses with certain widths) that corresponds to the difference in frequency. This output is fed into an amplifier and then to a low pass filter in order to extract the required information. The extracted information will be in the form of a certain DC value, and this value is then fed into the FSM in order to be conditioned and to get a suitable output for adapting the system. The output of the FSM is used to adjust the gain of the variable gain block G in the loop and the frequency of the DCO. The synthesizer structure uses a variable gain block G that makes the adaptation process possible.

### 3.2. Fractional TDTL-FS

Fractional synthesizers are the type of systems that can produce not only integer multiple (÷N) of the source frequency but also a fractional multiple (÷N.p) of it. The fractional synthesizer basically uses a type of fractional dividers and employs it to produce fractional divisions. The fractional divider can be constructed using different methods. In this paper a modulus-2 prescaler divider was designed and subsequently introduced into the system. This type of fractional divider uses two integer dividers that are controlled by a control unit. The inputs of the control unit are driven by the outputs of the two dividers in order to generate the signal that controls the MUX and DeMUX blocks. This control unit is the one responsible for creating the fraction part of the division. The two integer dividers have to be increments of each other i.e. 5 and 6 or 7 and 8 etc, in general P/P+1. The block diagram of the modulus-2 prescaler is shown in Figure 10.

As discussed in the previous integer divider section, introducing a divider into the TDTL will cause some changes in the performance of the system that requires certain adjustments. Similarly, introducing a fractional divider requires employing the same adaptation principle; however the case now is more difficult because of the use of two independent dividers as well as a control mechanism that switches between these two dividers.

Before applying an adaptation mechanism, the loop was tested and analyzed after introducing the fractional divider. After different assessments, it was found that in the case of a fractional divider, the operating point of the loop does not shift outside the locking range into one single location but oscillates between two different locations.

Each division factor is responsible for shifting the operating point to one of the points. Therefore, in order to get the loop back into its nominal point, point A in Figure 7, the adaptation process has to compensate for the two divisions individually and one at a time. This observation implied that a more complex adaptation process is required that has the ability to adapt the system for two division factors and switch quickly between them as the control unit switches between the two dividers.

As a solution to the previous problems, duplicate blocks were added to the fractional divider that were used to condition the divider in such a way that it is now possible to extract the required information about the division factor. This approach includes duplicating the two dividers and using the duplicates as dummy counters that have access to the input signal all the time, not switched inputs, and two FSBs connected to those two counters. The outputs of the two FSBs are switched using a demultiplexer that is controlled by the same output of the control unit of the fractional divider. The block diagram of the adaptation process is shown in Figure 11 and the complete fractional TDTL-FS is depicted in Figure 12. The advantage of this approach is that the switching between the two adaptation signals from FSB1 and FSB2 is switched simultaneously with the switching of the division because they are both taking their switching signal from the same control unit.

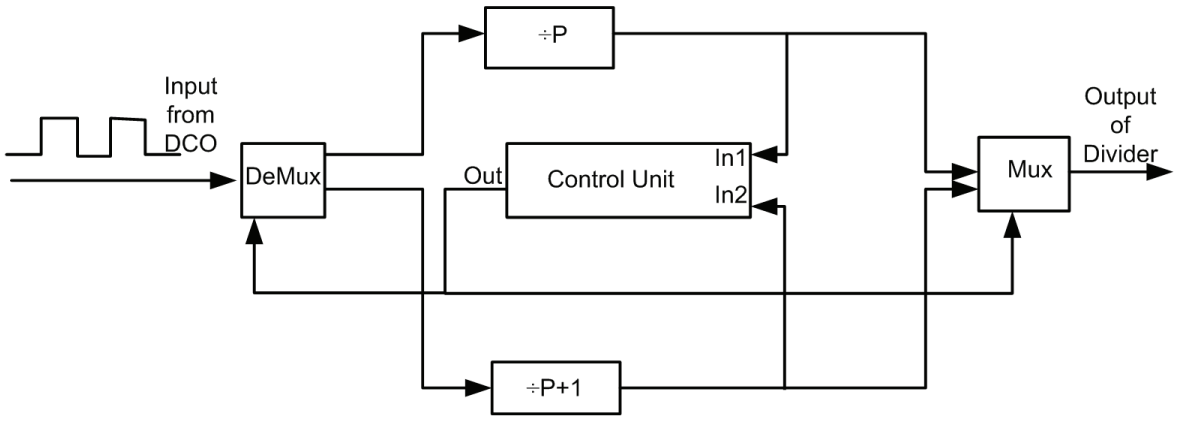



**Figure 10. Modulo-2 prescaler divider.**

Copyright © 2009 SciRes. 

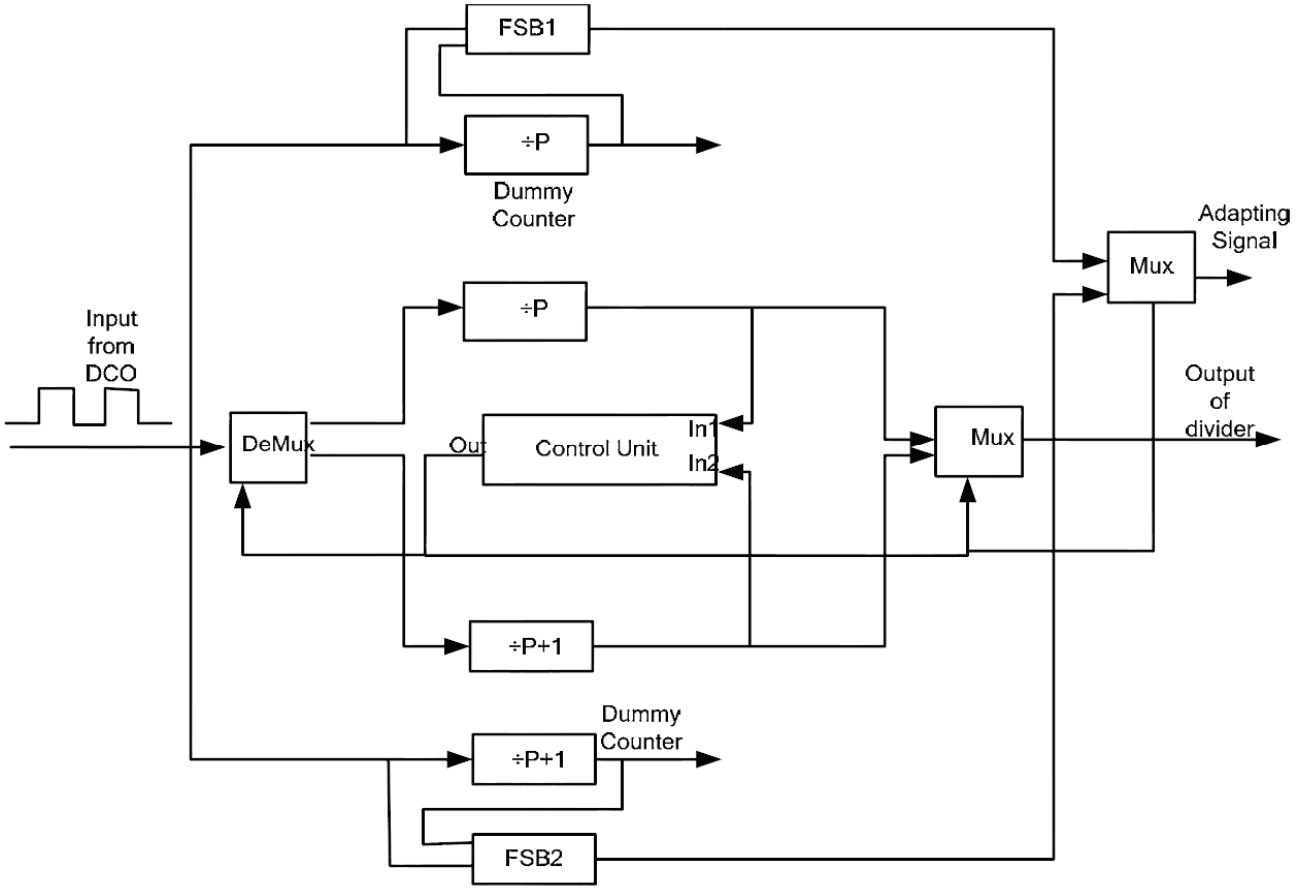



**Figure 11. Adaptation mechanism for fractional divider in TDTL-FS using FSB.**

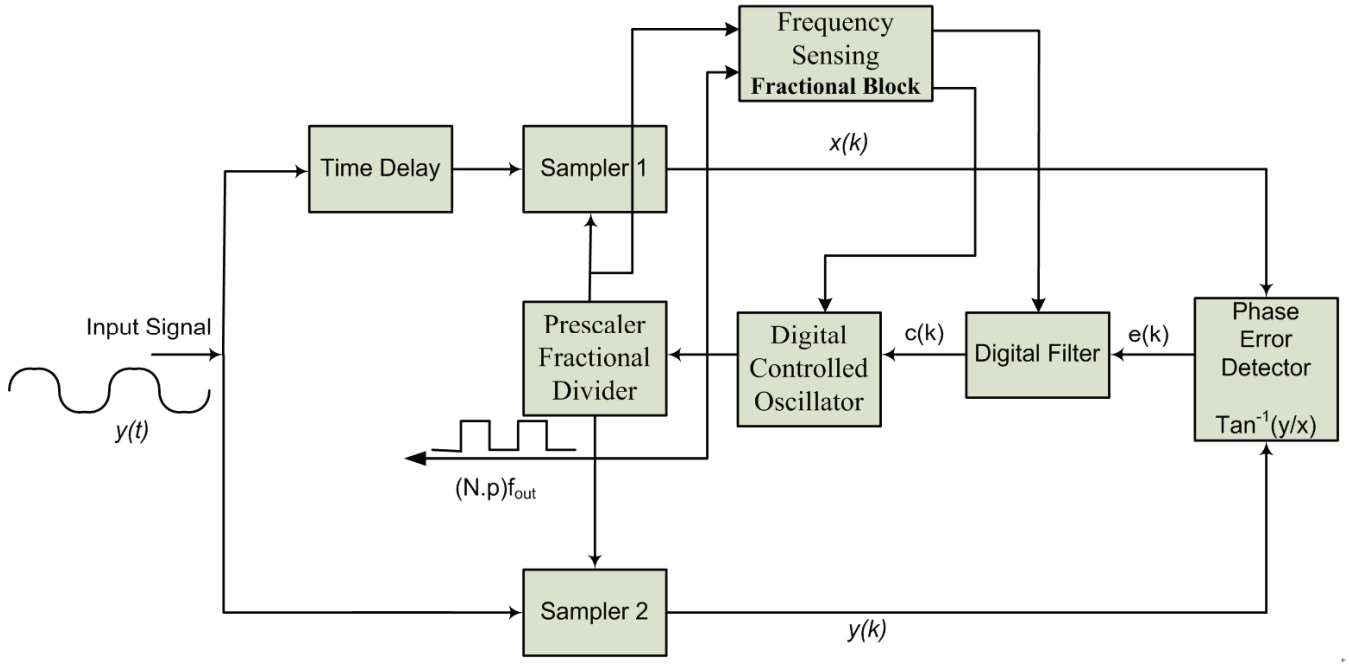



**Figure 12. Structure of fractional TDTL-FS with FSB.**

Copyright © 2009 SciRes. 

## 4. TDTL-FS Results

This section presents the simulation results of the TDTL frequency synthesizers for both integer and fractional divisions. The TDTL used in both cases is of the first-order type. The modeling, simulation, and testing of all system architectures were done using MATLAB/SIMULINK. The results in this section illustrate the ability of the TDTL frequency synthesizers to regain its locking state following the introduction of the division process into the loop. They also show that the system is capable of synthesizing frequencies that are integer as well as fractional multiples of the input frequency.

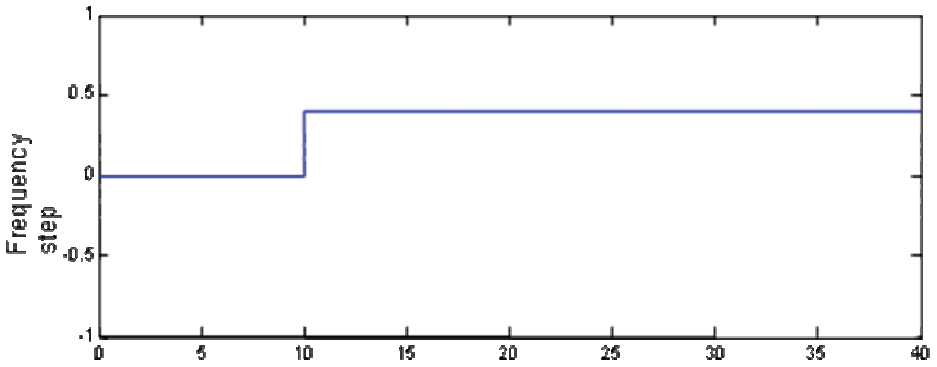
### 4.1. Integer Division

After applying the divider into the system, several input steps were subjected to the system. A positive step indicates an increase in the input frequency while a negative step indicates a decrease in the input frequency. In all cases and prior to the introduction of the division process the loop was chosen to operate at point A in Figure 7, which is considered the optimum point in the locking range as it allows symmetrical swing in the input signal. At point A the frequency ratio is $W=\omega_o/\omega=1$ and $K=G\omega_o=1$. The time delay was chosen to be $\psi_o=\omega_o\tau=\pi/2$. These conditions imply that the TDTL basic system is in lock.

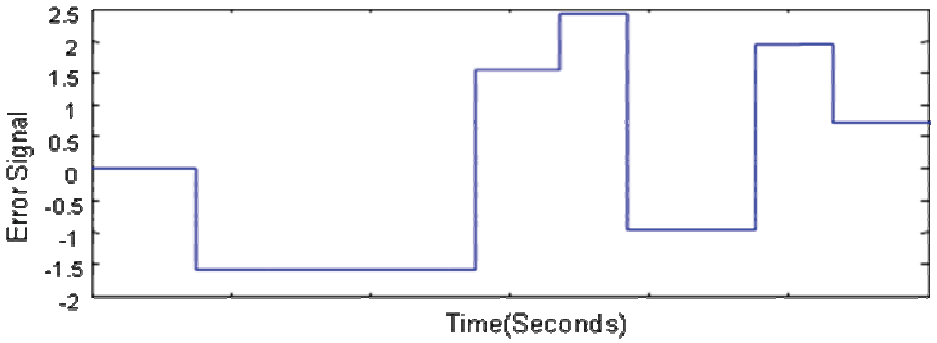
An example of applying a positive step to a TDTL-FS with and integer divider is shown in Figure 13. Figure 13(a) shows the input step with amplitude of 0.4 and system response in Figure 13(b) and Figure 13(c) prior to the incorporation of the FSB adaptation block. It can be clearly seen that the TDTL-FS is out of lock. Figure 13(c) depicts the phase plane response of the system which clearly shows the lack of convergence. The arrow in the figure indicates $\phi(0)$. The effect of the FSB on the stability of the TDTL-FS is shown in Figure 13(d). It can be seen that the system regained lock following the activation of the FSB block. Once the system stabilized it was able to generate correct integer division. This is illustrated in Figure 14 for a division factor of 16.

### 4.2. Fractional Division

The concept of TDTL-FS fractional synthesis using the FSB approach was achieved and simulated. An input with a positive frequency step was applied to the system. The system response to that step is shown in Figure 15. It can be clearly seen that the system locks onto the input but with a steady state error. The error does not converge to zero due to the limitations of the order of the loop filter, in which a simple gain block is used as the loop filter.

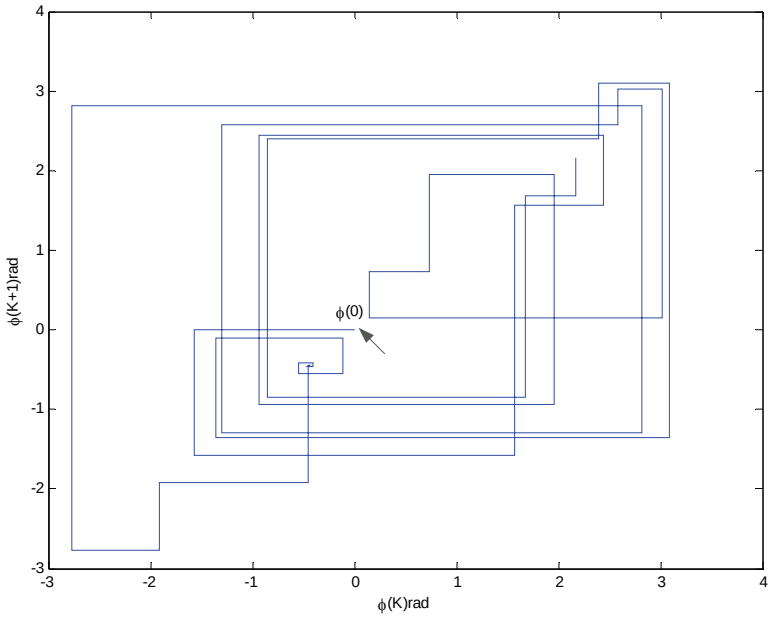



**(a)**

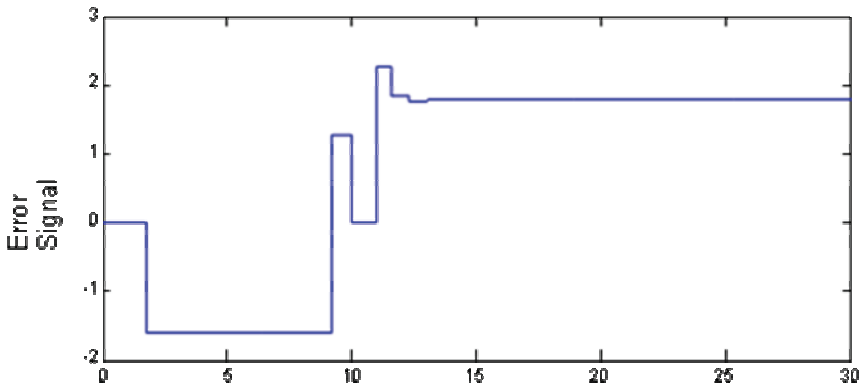



**(b)**

Copyright © 2009 SciRes. 




**(c)**




**(d)**

**Figure 13. TDT-FS system response for integer division (a) Positive step input, (b) Phase error without adaptation, (c) Phase plane without adaptation (d) Phase error with adaptation.**

It is important to mention that the response of the system to the frequency step is very similar to the response of the loop itself, this implies and proves that, in both cases, the system is operating at the same operating point. This observation concludes that the adaptation mechanism was successful in adjusting the loop and shifting it back into its original location. However, it is also important to point that the system adaptation using the FSB approach requires about 15 samples to initialize it and be able to adapt the system, and this is one of the disadvantages of this approach.

The division factor that was chosen for this simulation example was (÷3.25). The frequency division that was achieved for this example is shown in Figure 16. It can be clearly seen that the division cycles are broken into four parts, for the first three parts the input is divided by 3 and for the last part the input is divided by 4 and hence a division by 3.25 is achieved.

## 5. Conclusions

The TDTL is an efficient mixed-signal digital phase lock loop. This paper presented the architecture and governing system equations of the TDTL. It then presented a novel indirect frequency synthesis system based on the TDTL. The system shows a robust performance as it deals with sudden changes in the input signal frequencies.

Copyright © 2009 SciRes. 

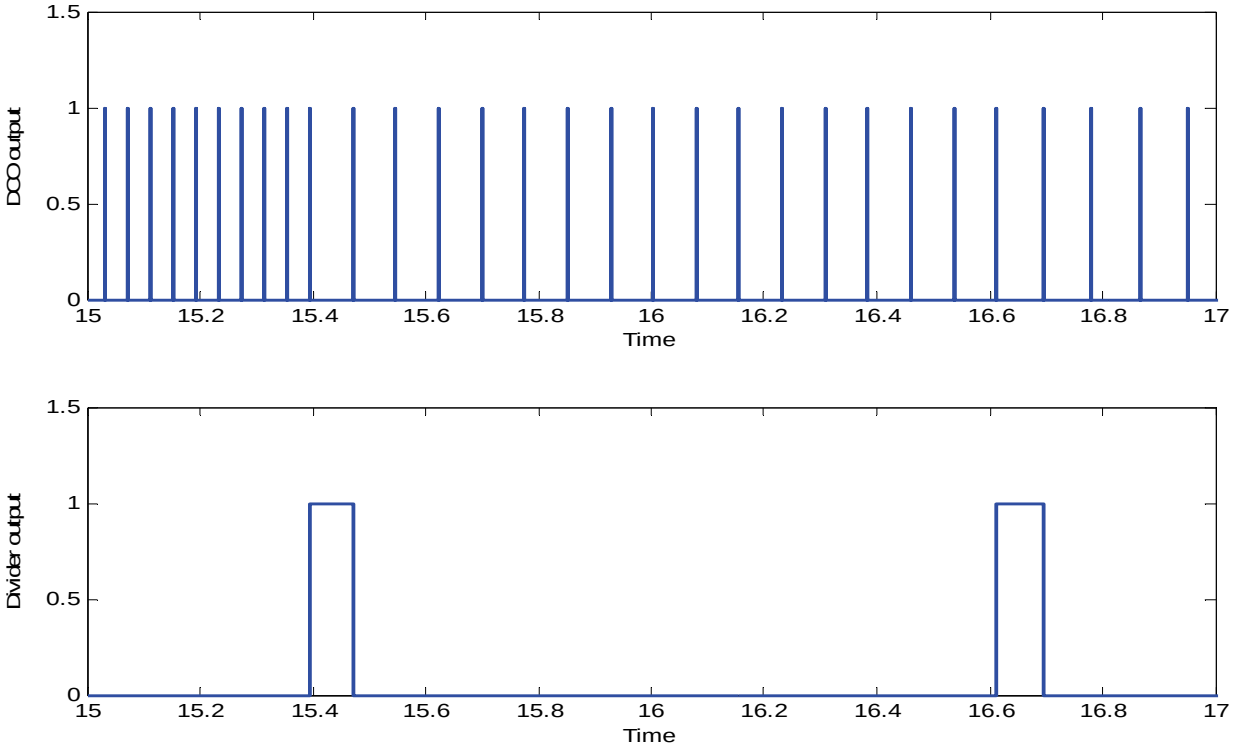
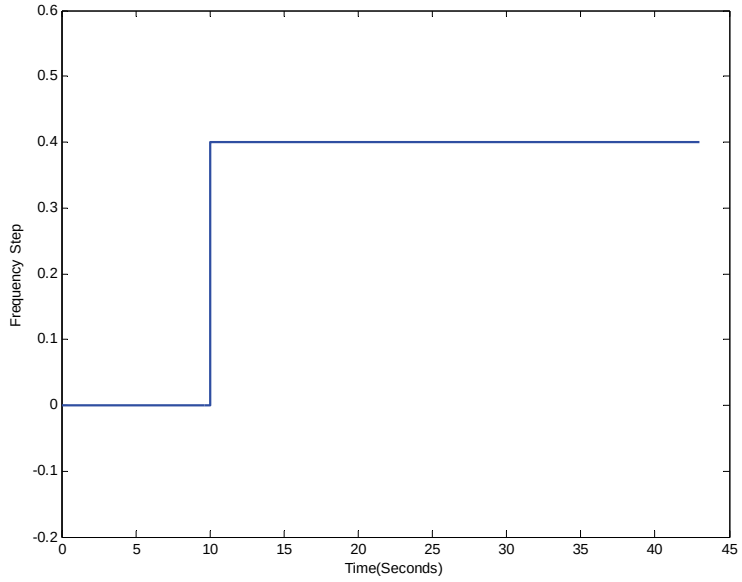



**Figure 14. First-order (Top) DCO output, (bottom) Divider output for ÷16.**

The achievement of locking following a system disturbance within an acceptable number of samples re-enforces the robustness aspect.

In order for the TDTL-FS system to be stable and hence usable an adaptive stabilization mechanism was incorporated with the synthesizer. The adaptive frequency sensing mechanism stabilizes the loop by controlling the gain block and the DCO of the loop in a way that counter acts the effect of the division process and brings the loop to its optimum operating point within the locking region.

The TDTL-based frequency synthesizer offers a number of advantages compared to other PLL synthesizers. These include an all digital architecture that is independent of the input signal level and a wider range of frequency synthesis. This is also coupled with relatively




Copyright © 2009 SciRes. 

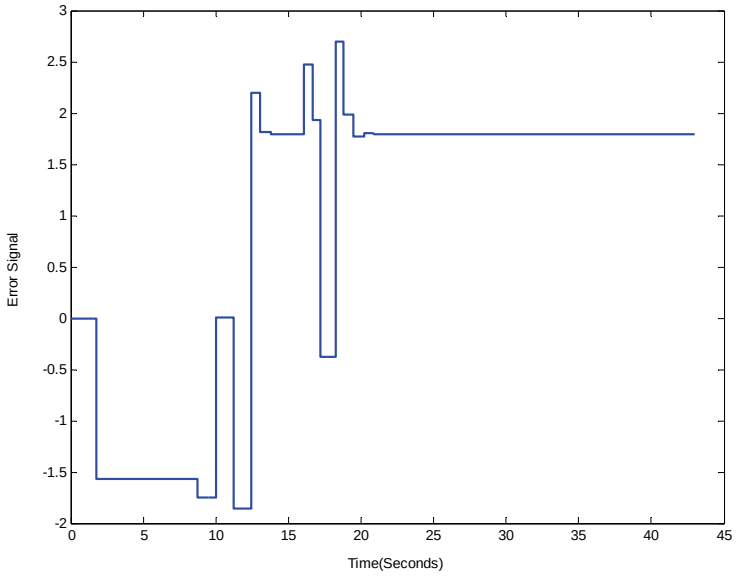



**Figure 15. TDTL-FS with FSB system response to a positive frequency step (Top): Frequency step input (Bottom): Output of the phase error detector.**

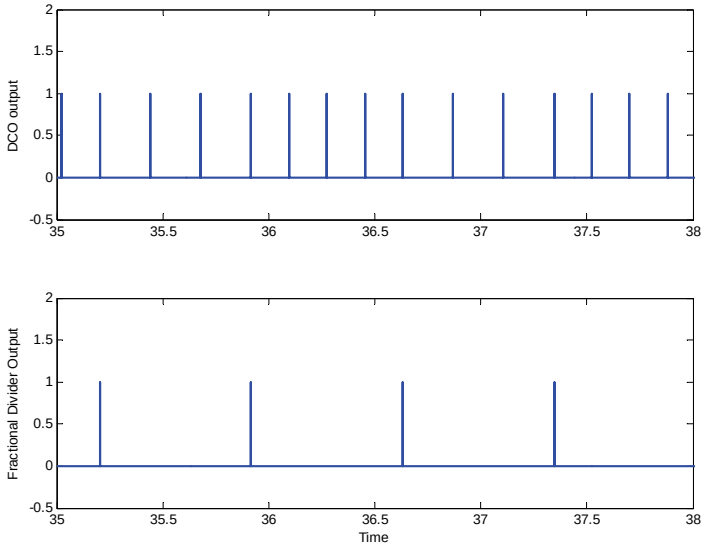



**Figure 16. Output of the divider and the DCO for a division of 3.25 (Top): DCO output (Bottom): Divider output.**

low system implementation complexity. The ability of the TDTL-FS to achieve locking as well as integer and fractional divisions were illustrated. The results indicate that fine divisions can be achieved. However, the speed of the adaptation process for fine divisions will need to be improved. This will be the subject of further work in the future.

## 6. References

[1] Y. Kim, B. Jang Jeong, J. Chung, C. Hwang, J. S. Ryu, K. Kim, and Y. Kim, "Beyond 3G: Vision, requirements, and enabling technologies," IEEE Communications Magazine, Vol. 41, No. 3, pp. 120−124, March 2003.

[2] A. Goldsmith, "Wireless communications," Cambridge University Press, 2005.

[3] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative wireless communications: A cross-layer approach," IEEE Wireless Communications, pp. 84−92, August 2006.

[4] C. Lam and B. Razavi, "A 2.6-GHz/5.2-GHz frequency synthesizer in 0.4-μm CMOS technology," IEEE Journal of Solid-State Circuits, Vol. 35, No. 5, pp. 788−794, May 2000.

Copyright © 2009 SciRes. 

[5] F. M. Gardner, "Phaselock techniques," 3rd Edition, Wiley, 2005.

[6] T. H. Lin, W. J. Kaiser, and G. J. Pottie, "Integrated low-power communication system design for wireless networks," IEEE Communications Magazine, pp. 142−150, December 2004.

[7] R. B. Staszewski and P. T. Balsara, "All-digital frequency synthesizer in deep-submicron CMOS," Wiley, 2006.

[8] S. T. Moon, A. Valero-Lopez, and E. Sanchez-Sinencio, "Fully integrated frequency synthesizers: A tutorial," International Journal of High Speed Electronics and Systems, Vol. 15, No. 2, pp. 353−375, June 2005.

[9] J. Vankka and K. A. I. Halonen, "Direct digital synthesizers: Theory, design and applications," Springer, 2001.

[10] D. D. Caro and A. G. M. Strollo, "High-performance direct digital frequency synthesizer using piecewise-polynomial approximation," IEEE Transaction Circuits and Systems I, Vol. 52, No. 2, pp. 324−337, February 2005.

[11] M. Kozak and E. G. Friedman, "Design and simulation of fractional-N PLL frequency synthesizer," International Symposium on Circuits and Systems, pp. 780−783, 2004.

[12] K. Shu and E. S. Sinencio, "CMOS PLL synthesizers: Analysis and design," Springer, 2005.

[13] K. Woo, Y. Liu, E. Nam, and D. Ham, "Fast-lock hybrid PLL combining fractional-N and integer-N modes of differing bandwidths," IEEE Journal of Solid-State Circuits, Vol. 43, No. 2, pp. 379−389, February 2008.

[14] Z. Hussain, B. Boashash, M. H. Ali, and S. A. Araji "A time-delay digital tanlock loop," IEEE Transaction Signal Processing, Vol. 49, No. 8, pp. 1808−1815, 2001.

[15] C. Lee and C. K. Un, "Performance analysis of digital tanlock loop," IEEE Transaction Communications, Vol. 30, pp. 2398−2411, October 1982.

[16] Z. M. Hussain, "Performance analysis of time-delay phase-shifter in additive Gaussian noise: A statistical comparison with Hilbert transformer," 6th International Symposium on Signal Processing and Its Applications (ISSPA), August 2001.

[17] M. A. Qutayri, S. A. Araji, and N. A. Moosa, "Improved first-order time-delay tanlock loop architectures," IEEE Transaction Circuits and Systems Part-I, Vol. 53, No. 9, pp. 1896−1908, 2006.

[18] S. R. A. Araji, Z. M. Hussain, and M. A. A. Qutayri, "Digital phase lock loops: Architectures and applications," Springer, 2006.

[19] J. C. Osborne, "Stability analysis of an nth power digital phase-locked loop-part I: First-order DPLL," IEEE Transaction Communications, Vol. 28, pp. 1343−1354, August 1980.

[20] Q. Nasir, "Extended lock range zero-crossing digital phase-locked loop with time delay," EURASIP Journal on Wireless Communications and Networks, Vol. 5, No. 3, pp. 413−418, 2005.

[21] Z. M. Hussain, "Convergence behavior of first-order time-delay digital tanlock loop," IEEE Communication Letters, Vol. 6, No. 7, pp. 291−293, July 2002.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 344-350
doi:10.4236/ijcns.2009.25037 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# A Novel Blind Channel Estimation for a 2x2 MIMO System

**Xia LIU[1], Marek E. BIALKOWSKI[2], Feng WANG[1]**
[1]*Student Member IEEE, School of ITEE, The University of Queensland, Brisbane, Australia*
[2]*Fellow IEEE, School of ITEE, The University of Queensland, Brisbane, Australia*
*Email*: {*xialiu, meb, fwang*}*@itee.uq.edu.au*
*Received May* 18, 2009; *revised June* 20, 2009; *accepted July* 17, 2009

## ABSTRACT

A novel blind channel estimation method based on a simple coding scheme for a 2 by 2 multiple input multiple output (MIMO) system is described. The proposed algorithm is easy to implement in comparison with conventional blind estimation algorithms, as it is able to recover the channel matrix without performing singular value decomposition (SVD) or eigenvalue decomposition (EVD). The block coding scheme accompanying the proposed estimation approach requires only a block encoder at the transmitter without the need of using the decoder at the receiver. The proposed block coding scheme offers the full coding rate and reduces the noise power to half of its original value. It eliminates the phase ambiguity using only one additional pilot sequence.

**Keywords:** MIMO, Channel Estimation, Semi-Blind Channel Estimation, Phase Ambiguity

## 1. Introduction

Multiple Input Multiple Output (MIMO) signal transmission schemes are attractive for high-speed data transmission in wireless communication systems because they offer an increased data throughput (capacity) without increasing operational bandwidth [1,2]. Also they are capable to enhance the quality of signal transmission through the use of transmitter or receiver diversity. These advantages are possible under the condition that the MIMO channel state information (CSI) is available at the receiver. Traditionally, CSI can be acquired by sending training sequences (also known as pilot signals) evenly spaced along a block of transmit symbols. The disadvantage of this approach is that the training sequences take up the precious bandwidth. In order to save the bandwidth and increase spectral efficiency, blind and semi-blind channel estimation methods can be applied to obtain the CSI.

Several blind channel estimation methods have been described in [3,4]. These methods are based on the subspace algorithm [5], which utilizes the orthogonality between the channel matrix and the Sylvester matrix-formed noise subspace. There are several drawbacks of subspace-based MIMO channel estimation methods. One is that they suffer from so-called multi-dimensional ambiguity. As a result, several pilot sequences are needed to eliminate this ambiguity. Two, in order to compensate for extra degrees of freedom in the noise subspace when the number of transmit antennas is smaller than the number of antennas at the receiver, the pre-coding is required [3,4]. Also, EVD is an inherent part of the algorithm, which leads to high implementation complexities.

In [6,7], a semi-blind channel estimation method employing orthogonal pilot maximum likelihood (OPML) estimator has been proposed. The method performs singular value decomposition (SVD) to the received signal correlation matrix to estimate the 'whitening' matrix of channel. By using the 'whitening' matrix, the OPML estimator shows a 1dB improvement of bit error rate (BER) compared to the conventional least squares (LS) training scheme if the same length of training sequence is used. However, it still requires a large number of training symbols to achieve the same performance as LS. Furthermore, SVD has to be applied twice to obtain the 'whitening' matrix and the rotation matrix. These operations lead to the increased computational complexity.

The work in [8] presents a new SVD-based blind channel estimation scheme which uses a simple block pre-coding structure. The advantage of this approach is that CSI can be recovered without ambiguity if the

Copyright © 2009 SciRes. 

proper modulation is applied. Another advantage of this scheme is that no block decoder is needed at the receiver. These advantages are gained at the expense of the coding rate. The coding rate decreases as more transmitting antennas are used. In particular, for a 2x2 MIMO system the code rate is 1/2, which results in wasting of the precious spectrum.

In this paper, we propose a novel blind channel estimation algorithm, which is of much lesser complexity than those based on SVD or EVD. Its important feature is that it preserves the advantages of the coding scheme described in [8] without sacrificing the coding rate. The new scheme offers a full coding rate (coding rate is equal to 1) when the number of transmitting antennas is equal to the number of receiving antennas. In the case of a 2x2 MIMO system, this coding scheme reduces the noise power to the half of the original noise power. This scheme exhibits the phase ambiguity. However, it can be eliminated using only one extra pilot sequence.

The rest of the paper is organized as follows. In Section 2, a model of MIMO system employing a block coding scheme is introduced. A new blind channel estimation method is described in Section 3. The solution of eliminating the phase ambiguity is given in Section 4. Simulation results are presented in Section 5. Section 6 concludes the paper.

## 2. System Description & Coding Scheme

In this paper, a narrow band block fading channel is assumed. The number of transmitting and receiving antennas is denoted as $N_t$ and $N_r$, respectively. Thus the channel $H$ is the $N_r$ x $N_t$ dimension channel matrix with $h_{ij}$ representing complex response between the $i$-th receiving antenna and the $j$-th transmitting antenna. In further considerations $N_t$ is assumed to be equal to $N_r$.

The input symbols at transmitter can be represented by

$$X = \{x_1, x_2, x_3, .....\} \quad \textbf{(1)}$$

where $X$ stands for independent identically distributed (i.i.d) Gaussian random signals with zero mean and the variance matrix given by $E\{x_n x_m{}^H\} = \begin{cases} \sigma_s^2, & n = m \\ 0, & n \neq m \end{cases}$, where $E\{\}$ implies the expectation and $\sigma_s{}^2$ is the power of one symbol.

The symbols are encoded using a block encoder structure before being transmitted. As a result, the $i$-th symbol block is an element of matrix group $A_i \in C^{N_r \times N_r N_t} = C^{2\times 4}$.

The data received at the other end of the communication channel is affected by the channel properties and an additive noise. Therefore the relationship between the transmitted encoded symbols and received data is given as:

$$Y_i = HA_i + N_i \quad \textbf{(2)}$$

where $Y_i$ is the $N_r$ x $N_rN_t$ received signal matrix and $N_i$ is the $N_r$ x $N_rN_t$ (i.i.d) Gaussian random noise matrix with zero mean.

The coded output of the transmitter can be written as:

$$A_i = \begin{bmatrix} A_{4i+1} & A_{4i+2} & A_{4i+3} & A_{4i+4} \end{bmatrix} \quad \textbf{(3)}$$

where

$$\begin{aligned} A_{4i+1} &= diag(U_1) X_{2i+1}^T, \\ A_{4i+2} &= diag(U_2) X_{2i+1}^T, \\ A_{4i+4} &= diag(U_2) X_{2i+2}^T, \quad i = 0,1,2,.....,n \end{aligned} \quad \textbf{(4)}$$

and $X_{2i+1} = [x_{4i+1} \quad x_{4i+2}]$ and $X_{2i+2} = [x_{4i+3} \quad x_{4i+4}]$. $U = [U_1^T \quad U_2^T]$, $U_1 = [1 \quad 1]$, $U_2 = [1 \quad -1]$ represent the encoder structure.

Therefore, the transmitted coded signals are

$$A_{4i+1} = \begin{bmatrix} x_{4i+1} \\ x_{4i+2} \end{bmatrix}, A_{4i+2} = \begin{bmatrix} x_{4i+1} \\ -x_{4i+2} \end{bmatrix}$$

$$A_{4i+3} = \begin{bmatrix} x_{4i+3} \\ x_{4i+4} \end{bmatrix}, A_{4i+4} = \begin{bmatrix} x_{4i+3} \\ -x_{4i+4} \end{bmatrix} \quad \textbf{(5)}$$

or

$$A_i = \begin{bmatrix} x_{4i+1} & x_{4i+1} & x_{4i+3} & x_{4i+3} \\ x_{4i+2} & -x_{4i+2} & x_{4i+4} & -x_{4i+4} \end{bmatrix} \quad \textbf{(6)}$$

From expression (6), one can observe that 4 symbols are sent in 4 symbol periods during one block. Therefore, the code rate is 1.

The received signal blocks can be written as (7):

$$\begin{aligned} Y_i &= \begin{bmatrix} y_{11}^{2i+1} & y_{12}^{2i+1} & y_{11}^{2i+2} & y_{12}^{2i+2} \\ y_{21}^{2i+1} & y_{22}^{2i+1} & y_{21}^{2i+2} & y_{22}^{2i+2} \end{bmatrix} \\ &= \begin{bmatrix} h_{11}x_{4i+1} + h_{12}x_{4i+2} & h_{11}x_{4i+1} - h_{12}x_{4i+2} & h_{11}x_{4i+3} + h_{12}x_{4i+4} & h_{11}x_{4i+3} - h_{12}x_{4i+4} \\ h_{21}x_{4i+1} + h_{22}x_{4i+2} & h_{21}x_{4i+1} - h_{22}x_{4i+2} & h_{21}x_{4i+3} + h_{22}x_{4i+4} & h_{21}x_{4i+3} - h_{22}x_{4i+4} \end{bmatrix} + N_i \\ &= HA_i + N_i \end{aligned} \quad \textbf{(7)}$$

Copyright © 2009 SciRes. 

in which $N_i = \begin{bmatrix} n_{11}^{2i+1} & n_{12}^{2i+1} & n_{11}^{2i+2} & n_{12}^{2i+2} \\ n_{21}^{2i+1} & n_{22}^{2i+1} & n_{21}^{2i+2} & n_{22}^{2i+2} \end{bmatrix}$ is the random Gaussian noise matrix. An equivalent representation of (7) is given by (8):

$$\begin{aligned} y_{11}^{2i+1} &= h_{11}x_{4i+1} + h_{12}x_{4i+2} + n_{11}^{2i+1} \quad y_{21}^{2i+1} = h_{21}x_{4i+1} + h_{22}x_{4i+2} + n_{21}^{2i+1} \\ y_{12}^{2i+1} &= h_{11}x_{4i+1} - h_{12}x_{4i+2} + n_{12}^{2i+1} \text{ and } y_{22}^{2i+1} = h_{21}x_{4i+1} - h_{22}x_{4i+2} + n_{22}^{2i+1} \\ y_{11}^{2i+2} &= h_{11}x_{4i+3} + h_{12}x_{4i+4} + n_{11}^{2i+2} \quad y_{21}^{2i+2} = h_{21}x_{4i+3} + h_{22}x_{4i+4} + n_{21}^{2i+2} \\ y_{12}^{2i+2} &= h_{11}x_{4i+3} - h_{12}x_{4i+4} + n_{12}^{2i+2} \text{ and } y_{22}^{2i+2} = h_{21}x_{4i+3} - h_{22}x_{4i+4} + n_{22}^{2i+2} \end{aligned} \quad \textbf{(8)}$$

By linking (8) directly to individual channel matrix elements, one obtains:

$$\begin{aligned} h_{11}x_{4i+1} &= \frac{y_{11}^{2i+1} + y_{12}^{2i+1}}{2} - \frac{n_{12}^{2i+1} + n_{11}^{2i+1}}{2}, \\ h_{12}x_{4i+2} &= -\frac{y_{12}^{2i+1} - y_{11}^{2i+1}}{2} + \frac{n_{12}^{2i+1} - n_{11}^{2i+1}}{2} \end{aligned} \quad \textbf{(9)}$$

$$\begin{aligned} h_{21}x_{4i+1} &= \frac{y_{21}^{2i+1} + y_{22}^{2i+1}}{2} - \frac{n_{22}^{2i+1} + n_{21}^{2i+1}}{2}, \\ h_{22}x_{4i+2} &= -\frac{y_{22}^{2i+1} - y_{21}^{2i+1}}{2} + \frac{n_{22}^{2i+1} + n_{21}^{2i+1}}{2} \end{aligned} \quad \textbf{(10)}$$

$$\begin{aligned} h_{11}x_{4i+3} &= \frac{y_{11}^{2i+2} + y_{12}^{2i+2}}{2} - \frac{n_{12}^{2i+2} + n_{11}^{2i+2}}{2}, \\ h_{12}x_{4i+4} &= -\frac{y_{12}^{2i+2} - y_{11}^{2i+2}}{2} + \frac{n_{12}^{2i+2} + n_{11}^{2i+2}}{2} \end{aligned} \quad \textbf{(11)}$$

$$\begin{aligned} h_{21}x_{4i+3} &= \frac{y_{21}^{2i+2} + y_{22}^{2i+2}}{2} - \frac{n_{22}^{2i+2} + n_{21}^{2i+2}}{2}, \\ h_{22}x_{4i+4} &= -\frac{y_{22}^{2i+2} - y_{21}^{2i+2}}{2} + \frac{n_{22}^{2i+2} - n_{21}^{2i+2}}{2} \end{aligned} \quad \textbf{(12)}$$

As a result, the relationship between the raw (transmitted) data and the received data is given by (13):

$$\overline{Y_i} = \overline{H}\overline{X_i} + \overline{N_i} \quad \textbf{(13)}$$

in which the individual terms are identified by (14)

$$\begin{bmatrix} \frac{y_{11}^{2i+1} + y_{12}^{2i+1}}{2} \\ \frac{y_{11}^{2i+1} - y_{12}^{2i+1}}{2} \\ \frac{y_{11}^{2i+2} + y_{22}^{2i+2}}{2} \\ \frac{y_{21}^{2i+2} - y_{22}^{2i+2}}{2} \end{bmatrix} = \begin{bmatrix} h_{11} & 0 & 0 & 0 \\ 0 & h_{12} & 0 & 0 \\ 0 & 0 & h_{21} & 0 \\ 0 & 0 & 0 & h_{22} \end{bmatrix} \begin{bmatrix} x_{4i+1} \\ x_{4i+2} \\ x_{4i+3} \\ x_{4i+4} \end{bmatrix} + \begin{bmatrix} n_{4i+1} \\ n_{4i+2} \\ n_{4i+3} \\ n_{4i+4} \end{bmatrix} \quad \textbf{(14)}$$

where

$$\begin{aligned} n_{4i+1} &= \frac{n_{12}^{2i+1} + n_{11}^{2i+1}}{2}, n_{4i+2} = \frac{n_{11}^{2i+1} - n_{12}^{2i+1}}{2}, \\ n_{4i+3} &= \frac{n_{22}^{2i+2} + n_{21}^{2i+2}}{2}, n_{4i+4} = \frac{n_{21}^{2i+2} - n_{22}^{2i+2}}{2} \end{aligned} \quad \textbf{(15)}$$

Due to the fact that the elements in *Ni* represent a Gaussian random noise also the elements in $\overline{N_i}$ obey the Gaussian distribution. However, the average power of each element in $\overline{N_i}$ is half of that in $N_i$. In this case, the noise power is suppressed by the coding scheme.

## 3. Blind Channel Estimation

The blind channel estimation requires the knowledge of the correlation matrix of $\overline{Y}$, which is given as:

$$R_i = E\{\overline{Y_i}\,\overline{Y_i}^H\} = \overline{H}\,E\{\overline{X_i}\,\overline{X_i}^H\}\overline{H}^H + E\{\overline{N_i}\,\overline{N_i}^H\} \quad \textbf{(16)}$$

Because the power of each element in $\overline{N_i}$ is half of that in the noise matrix $N_i$, then Equation (16) can be rewritten as:

$$R_i = E\{\overline{Y_i}\,\overline{Y_i}^H\} = \overline{H}\,E\{\overline{X_i}\,\overline{X_i}^H\}\overline{H}^H + \frac{1}{2}E\{N_i N_i^H\} \quad \textbf{(17)}$$

If the information symbol sequence is of unit power then (17) becomes:

$$R_i = \overline{H}\,\overline{\overline{X}_i}\,\overline{H}^H + \frac{1}{2}E\{N_i N_i^H\} \quad \textbf{(18)}$$

where

$$\overline{\overline{X}} = E\{\overline{X_i}\,\overline{X_i}^H\} = I_{N_r N_t \times N_r N_t} \quad \textbf{(19)}$$

Thus (18) can be converted to (19)

$$R_i = \overline{H}\,\overline{H}^H + \frac{1}{2}E\{N_i N_i^H\} \quad \textbf{(20)}$$

By introducing $\bar{h} = vec(H)$, the following holds:

Copyright © 2009 SciRes. 

$$\overline{HH}^{H} = diag(\left|\overline{h}\right|^2) \quad \textbf{(21)}$$

where vec (.) means vector operation in which columns of $H$ are stacked on top of each other and |.| denotes the absolute value.

$$\left|\overline{h}\right|^2 = \left[\begin{array}{cccc} \left|h_{11}\right|^2 & \left|h_{12}\right|^2 & \cdots & \left|h_{N_r N_t}\right|^2 \end{array}\right] \quad \textbf{(22)}$$

As a result

$$R_i = diag(\left|\overline{h}\right|^2) + \frac{1}{2}E\{N_i N_i{}^H\} \quad \textbf{(23)}$$

The estimation of $\overline{h}$ is equivalent to finding the roots of the diagonal elements in $R$.

To obtain the solution, the square-root algorithm can be applied. One problem that is faced using this approach is that it introduces the phase ambiguity in the estimated $\overline{h}$. This is because the square roots are obtained for the norms of the elements of the channel matrix $H$. In the next section, a method for the phase ambiguity elimination is described.

## 4. Phase Ambiguity Elimination

It is apparent that the proposed blind channel estimation algorithm provides the information about the estimated norm of each element in channel matrix $H$ of the 2x2 MIMO system as shown by the following.

$$\hat{H} = \left[\begin{array}{cc} \left|\hat{h}_{11}\right| & \left|\hat{h}_{12}\right| \\ \left|\hat{h}_{21}\right| & \left|\hat{h}_{22}\right| \end{array}\right] \quad \textbf{(24)}$$

Now the task is to obtain the phases of these elements. By sending one pilot sequence $P$ where

$$P = \left[\begin{array}{c} P_1 \\ P_2 \end{array}\right] = \left[\begin{array}{c} 1 \\ 1 \end{array}\right] \quad \textbf{(25)}$$

one obtains

$$Y_p = HP + N \quad \textbf{(26)}$$

in which $Y_p$ and $P$ are known. More specifically, we have

$$\begin{aligned} Y_p &= \left[\begin{array}{c} X_p^1 + j \cdot Y_p^1 \\ X_p^2 + j \cdot Y_p^2 \end{array}\right] \\ &= \left[\begin{array}{cc} \left|h_{11}\right| e^{j\theta_{11}} & \left|h_{12}\right| e^{j\theta_{12}} \\ \left|h_{21}\right| e^{j\theta_{21}} & \left|h_{22}\right| e^{j\theta_{22}} \end{array}\right] \left[\begin{array}{c} 1 \\ 1 \end{array}\right] + \left[\begin{array}{c} N_1 \\ N_2 \end{array}\right] \end{aligned} \quad \textbf{(27)}$$

By replacing $\left|h_{N_r N_t}\right|$ in (25) by the estimated values, the following is obtained

$$X_p^1 = \left|\hat{h}_{11}\right| \cos\theta_{11} + \left|\hat{h}_{12}\right| \cos\theta_{12} \quad \textbf{(28)}$$

$$Y_p^1 = \left|\hat{h}_{11}\right| \sin\theta_{11} + \left|\hat{h}_{12}\right| \sin\theta_{12} \quad \textbf{(29)}$$

$$X_p^2 = \left|\hat{h}_{21}\right| \cos\theta_{21} + \left|\hat{h}_{22}\right| \cos\theta_{22} \quad \textbf{(30)}$$

$$Y_p^2 = \left|\hat{h}_{21}\right| \sin\theta_{21} + \left|\hat{h}_{22}\right| \sin\theta_{22} \quad \textbf{(31)}$$

To determine $\theta_{11}$, $\theta_{12}$, $\theta_{21}$ and $\theta_{22}$, Equations (9)(10) and (11)(12) are used in which (9)/(10) and (11)/(12) are formed. This operation results in the following

$$\begin{aligned} \frac{h_{11}}{h_{21}} &= \frac{\left|h_{11}\right|}{\left|h_{21}\right|} e^{j(\theta_{11}-\theta_{21})} = \frac{\left|h_{11}\right|}{\left|h_{21}\right|} e^{j\sigma_1} = \frac{\frac{y_{11}^{2i+1} + y_{12}^{2i+1}}{2} - \frac{n_{12}^{2i+1} + n_{11}^{2i+1}}{2}}{\frac{y_{21}^{2i+1} + y_{22}^{2i+1}}{2} - \frac{n_{22}^{2i+1} + n_{21}^{2i+1}}{2}}, \\ \frac{h_{12}}{h_{22}} &= \frac{\left|h_{12}\right|}{\left|h_{22}\right|} e^{j(\theta_{12}-\theta_{22})} = \frac{\left|h_{12}\right|}{\left|h_{22}\right|} e^{j\sigma_2} = \frac{\frac{y_{12}^{2i+1} - y_{11}^{2i+1}}{2} - \frac{n_{12}^{2i+1} + n_{11}^{2i+1}}{2}}{\frac{y_{22}^{2i+1} - y_{21}^{2i+1}}{2} - \frac{n_{22}^{2i+1} + n_{21}^{2i+1}}{2}} \end{aligned} \quad \textbf{(32)}$$

$$\begin{aligned} \frac{h_{11}}{h_{21}} &= \frac{\left|h_{11}\right|}{\left|h_{21}\right|} e^{j(\theta_{11}-\theta_{21})} = \frac{\left|h_{11}\right|}{\left|h_{21}\right|} e^{j\sigma_1} = \frac{\frac{y_{11}^{2i+2} + y_{12}^{2i+2}}{2} - \frac{n_{12}^{2i+2} + n_{11}^{2i+2}}{2}}{\frac{y_{21}^{2i+2} + y_{22}^{2i+2}}{2} - \frac{n_{22}^{2i+2} + n_{21}^{2i+2}}{2}}, \\ \frac{h_{12}}{h_{22}} &= \frac{\left|h_{12}\right|}{\left|h_{22}\right|} e^{j(\theta_{12}-\theta_{22})} = \frac{\left|h_{12}\right|}{\left|h_{22}\right|} e^{j\sigma_2} = \frac{\frac{y_{12}^{2i+2} - y_{11}^{2i+2}}{2} - \frac{n_{12}^{2i+2} + n_{11}^{2i+2}}{2}}{\frac{y_{22}^{2i+2} - y_{21}^{2i+2}}{2} - \frac{n_{22}^{2i+2} + n_{21}^{2i+2}}{2}} \end{aligned} \quad \textbf{(33)}$$

Copyright © 2009 SciRes. 

where $\sigma_1=\theta_{11}-\theta_{21}$ and $\sigma_2=\theta_{12}-\theta_{22}$.

Using Equations (32) and (33), $\sigma_1$ and $\sigma_2$ can be estimated by including noise impact. Then $\theta_{11}$, $\theta_{21}$ and $\theta_{12}$, $\theta_{22}$ can be expressed using the estimated $\sigma_1$ and $\sigma_2$ as,

$$\theta_{21} = \hat{\sigma}_1 - \theta_{11} \quad (34)$$

$$\theta_{22} = \hat{\sigma}_2 - \theta_{12} \quad (35)$$

By substituting (34) into Equations (32) and (35) into Equations (33), the phases $\theta_{11}$, $\theta_{21}$, $\theta_{12}$ and $\theta_{22}$ can be determined. Therefore, the phase ambiguity can be resolved.

## 5. Simulation Results

The validity of the proposed blind channel estimation algorithm for a 2x2 MIMO is investigated via computer simulations. For reference purposes, comparisons are made with a training-based channel estimation using the least square (LS) method and a semi-blind channel estimation using an orthogonal pilot maximum likelihood (OPML) algorithm. The following is the necessary information that is used to perform comparisons with the alternative channel estimation algorithms.

The performance of the proposed channel estimation is assessed in terms of mean square error (MSE) as given by

$$MSE = E\{\left\| H - \hat{H} \right\|_F^2\} \quad (36)$$

in which $\|.\|_F^2$ stands for the Frobenius norm.

In the LS method, the estimated channel matrix is given as [10],

$$\hat{H}_{LS} = YP^{\dagger} \quad (37)$$

where $\{.\}^{\dagger}$ stands for the pseudo-inverse operation.

The MSE of LS method is given as

$$MSE_{LS} = E\{\left\| H - \hat{H}_{LS} \right\|_F^2\} \quad (38)$$

According to [11] and [12], the minimum value of MSE for the LS method is given as

$$MSE_{\min}^{LS} = \frac{M_t^2 M_r}{\rho} \quad (39)$$

in which $\rho$ stands for transmitted power to noise ratio in the training mode. Here we assume that the SNR in the proposed estimation scenario is equal to $\rho$.

In [6] and [7], a WR-based semi-blind channel estimation method was introduced. Following that method, the MIMO channel matrix $H$ can be decomposed by applying the singular value (SV) decomposition

$$H \xrightarrow{SVD} P\Sigma Q^H \quad (40)$$

where $P$ and $Q$ are two singular vectors corresponding to eigenvalues. $\Sigma$ represents eigenvalues of $H$ by the diagonal matrix. For both $P$ and $Q$ the following properties hold: $PP^H = P^HP = I$ and $QQ^H = Q^HQ = I$. The whitening matrix $W$ is given by $W = P\Sigma$ and $Q$ is the rotation matrix. The whitening matrix $W$ can be obtained blindly by computing the second-order statistics of a received signal. Details are given in Section 2.3 of [13] and thus are not repeated here. The matrix $W$ is assumed to be perfectly known at the receiver. Training sequences are used for estimating the unitary matrix $Q$. It has already been proved in [6] and [7] that such a unitary matrix $Q$ helps increasing estimation gains because it uses a smaller number of parameters.

To estimate the rotation matrix $Q$, several algorithms can be applied. The orthogonal pilot maximum likelihood (OPML) algorithm offers the best performance, as demonstrated in [6]. In this algorithm, the training matrix $X_p$ is set to have orthogonal properties with unit power and length equal to $N_t$, $X_p^H X_p = X_p X_p^H = I$. The OPML estimator is expressed as (36),

$$\begin{aligned} &\arg\min \left\| Y_P - WQ^H X_P \right\|^2 \\ &subj.\ \ to\ \ QQ^H = I \end{aligned} \quad (41)$$

where $Q$ is obtained by minimizing the likelihood.

Let $\hat{M} = W^H Y_p X_p^H$, then by applying SVD to $\hat{M}$, we have

$$\hat{U}_M \hat{\Sigma}_M \hat{V}_M^H = SVD(\hat{M}) \quad (42)$$

It can be shown that the estimated $\hat{Q}$ of $Q$ that minimizes the likelihood is given as [6]:

$$\hat{Q} = \hat{V}_M \hat{U}_M^H \quad (43)$$

The channel matrix $H$ is then estimated as

$$\hat{H} = W\hat{Q}^H \quad (44)$$

In Figure 1, the performance of LS method, OPML algorithm and the proposed blind estimation method for a 2x2 MIMO system are presented. The length of training sequences used by LS method and OPML algorithm is set to 2.

From Figure 1, one can see that when the number of received symbols increases, the blind estimation accuracy is improved. The performance of blind channel estimation is always better than offered by LS. This is mainly because the proposed algorithm reduces the noise power to half of the original one. When making comparison with OPML algorithm, one finds that the per-

Copyright © 2009 SciRes. 

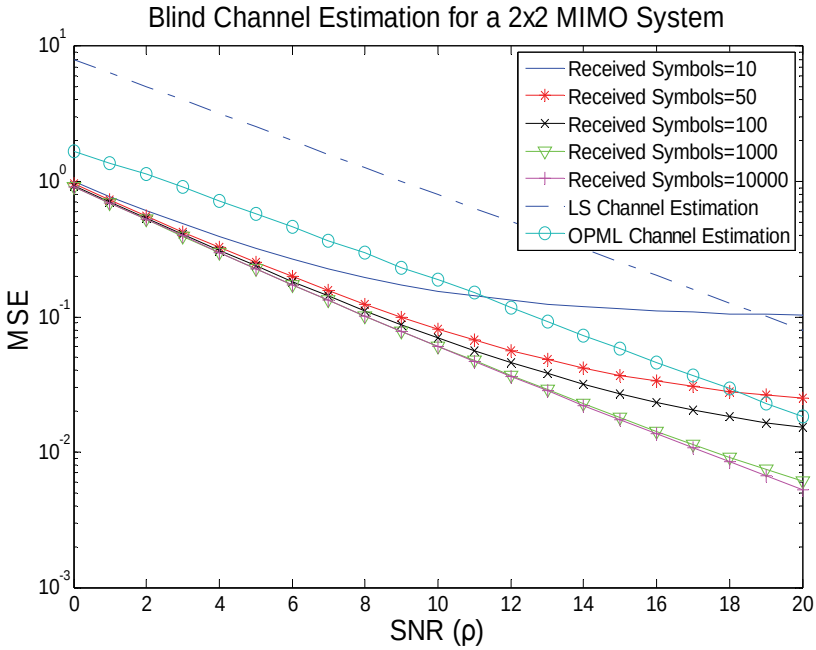



**Figure 1. Performance of LS, OPML and newly proposed blind estimation algorithm for a 2x2 MIMO system for different values of SNR.**

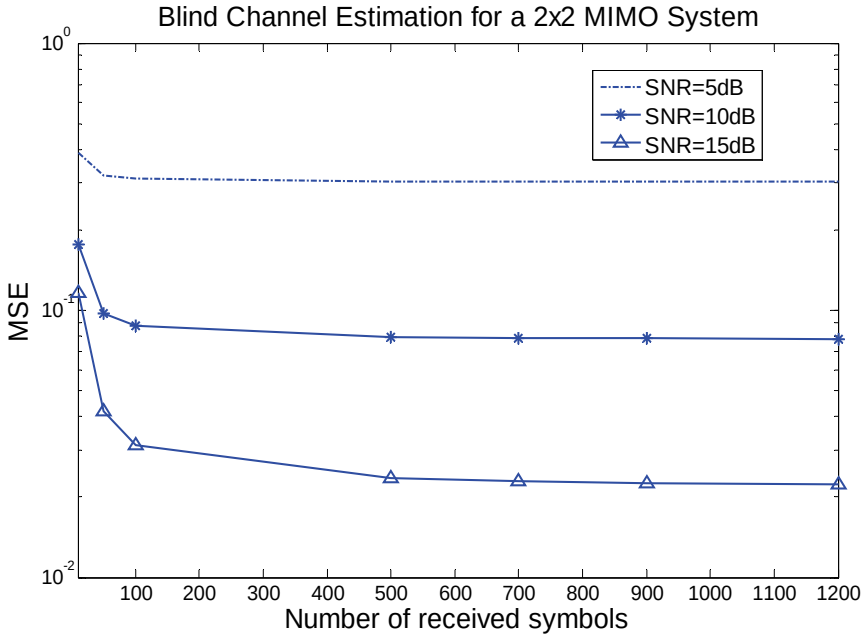



**Figure 2. MSE as a function of number of received symbols for a newly proposed blind channel estimation method for a 2x2 MIMO system for different values of SNR.**

formance of the proposed estimation algorithm is always better when the number of received symbols is equal or more than 100.

Figure 2 illustrates the convergence of the blind channel estimation for a 2x2 MIMO system. It can be seen that for each of the three assumed values of SNR, the convergence occurs within the first 100 received symbols. The use of a larger number of symbols (100 to 500) provides only a slight improvement. The full convergence occurs at about 700 symbols. One can see that with 100

Copyright © 2009 SciRes. 

received symbols the estimation is already very accurate irrespectively of the assumed value of SNR.

## 6. Conclusions

In this paper we have presented a novel blind algorithm for estimating a channel of a 2x2 MIMO. The proposed algorithm operates in conjunction with a suitable coding scheme and eliminates phase ambiguity for the estimated channel matrix coefficients. The coding scheme exhibits high spectral efficiency and reduces the noise power to the half of the original one that is present in a 2x2 MIMO system. The proposed algorithm involves the square-root operation which shows a low level of processing complexity. The simulation results prove a fast convergence rate for estimating the channel. The performance of the proposed algorithm is better than of the training-based Least Squares (LS) algorithm. Also it shows superiority over the Orthogonal Pilot Maximum Likelihood (OPML) semi-blind estimation algorithm when the number of received signal symbols exceeds 100.

## 7. References

[1] Telatar, "Capacity of multiple antenna Gaussian channels," European Transactions on Telecommunications, Vol. 10, No. 6, pp. 585−595, November/December 1999.

[2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," Wireless Personal Communications, Vol. 6, pp. 311−335, 1998.

[3] S. Zhou, B. Muquet, and G. B. Giannakis, "Subspace-based (semi-) blind channel estimation for block precoded space-time OFDM," IEEE Transactions on Signal Processing, Vol. 50, No. 5, pp. 1215−1228, May 2002.

[4] R. Zhang, "Blind OFDM channel estimation through linear Precoding: A subspace approach," in Proceedings Asilomar'02, Pacific Grove, CA, November 2002.

[5] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," IEEE Transactions on Signal Processing, Vol. 43, pp. 516−525, February 1995.

[6] A. K. Jagannatham and B. D. Rao, "Whitening-rotation-based semi-blind MIMO channel estimation," IEEE Transactions on Signal Processing, Vol. 54, No. 3, March 2006.

[7] A. Jagannatham and B. D. Rao, "Constrained ML algorithms for semi-blind MIMO channel estimation," Proceedings of IEEE Communication Society Globecom, 2004.

[8] X. Liu and M. E. Bialkowski, "SVD-Based blind channel estimation for a MIMO OFDM system employing a simple block pre-coding scheme," Proceedings of IEEE Eurocon, Poland, 2007.

[9] S. Shahbazpanahi, A. B. Gershman, and J. H. Manton, "Closed-form blind MIMO channel estimation for orthogonal space-time block codes," IEEE Transactions on Signal Processing, Vol. 53, No. 12, December 2005.

[10] S. M. Kay, "Fundamentals of statistic signal processing: Estimation theory," Prentice-Hall, Incorporation, 1993.

[11] M. Biguesh and A. B. Gershman, "MIMO channel estimation: Optimal training and tradeoffs between estimation techniques," Proceedings ICC'04, Paris, France, June 2004.

[12] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," IEEE Transactions on Signal Processing, Vol. 54, No. 3, March 2000.

[13] A. K. Jagannatham and B. D. Rao, "A semi-blind technique for MIMO channel matrix estimation," Proceedings IEEE Workshop on SPAWC, Roma, Italy, 2003.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 351-356
doi:10.4236/ijcns.2009.25038 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Iterative Detection and Decoding with PIC Algorithm for MIMO-OFDM Systems

**Zhongpeng WANG**
*School of Information Technology and Electronic Engineering*
*Zhejiang University of Science and Technology, Hangzhou, China*
*Email*: *wzp*1966@sohu.com
*Received February* 4, 2009; *revised March* 25, 2009; *accepted April* 30, 2009

## ABSTRACT

In this paper, we propose a new iterative detection and decoding scheme based on parallel interference cancel (PIC) for coded MIMO-OFDM systems. The performance of proposed receiver is improved through the joint PIC MIMO detection and iterative detection and decoding. Its performance is evaluated based on computer simulation. The simulation results indicate that the performance of the proposed receiver is greatly improved compared to coded MIMO-OFDM systems based on VBLAST detection scheme.

**Keywords:** Iterative Detection and Decoding, MIMO-OFDM, PIC Signal Detection, VBLAST

## 1. Introduction

In multipath environments, multiple input multiple output wireless communication system can increase spectral efficiency. Furthermore, to achieve high data rate wireless communications, space-time communication systems need to be in wideband frequency selective channels. Orthogonal frequency division multiple (OFDM) has become a popular technique for transmission of signals over wireless channels. OFDM are robust to frequency selective fading channels for OFDM systems use the discrete Fourier transform (DFT) to modulate the data on orthogonal frequency carriers and effectively divide the wideband channel into a number of narrowband flat channels. One important advantage of the OFDM transmission technique is that the intersymbol interference (ISI) can be removed if the channel delay spread is less than he inserted guard interval.

Clearly, conventional MIMO detection algorithms can be applied for MIMO-OFDM system [1−4]. The MIMO maximum likelihood detection detector is the optimal receiver, but its complexity is best high. A number of sub-optimum receivers of low to moderate complexity have been devised, yet all suffer from rather limited performance. The conventional VBLAST algorithm exhibits the best tradeoff between performance and complexity. However, it involves an intensive computation and hence it may be difficult to implement it for high data rate communications. Linear ZF and MMSE have the best low complexity but the performances are the worse. The QR [3] detection receiver avoids the matrix inversion, but the performance is not good as ZF-VBLAST. The QR detection based on ordering MMSE criterion is proposed [5]. Compared to zero-forcing criterion the detection method based on MMSE criterion needs estimating the signal-noise ratio or variance of noise.

Now considerable research interests have been focused on techniques and algorithms which realize various benefits of turbo principle for MIMO systems [6−8]. The method based on turbo principle, is regarded as an essential technique to furthermore improve system performance with soft iterative detection and decoding through an exchange of information. One of major drawbacks of such turbo-MIMO concepts is that its complexity increases exponentially with the number of transmit/receive antennas, the number of bits per symbol and /or the code constraint length.

However, the research work about iterative detection and decoding based on hard information is very litter. The hard iterative detection method has markedly advantage to soft iterative method in complexity. In this paper, we consider MIMO-OFDM systems with hard iterative detection process. In [9], joint processing of zero-forcing

Copyright © 2009 SciRes. 

detection and MAP decoding for MIMO-OFDM system. Inspired by [9,10], we introduce a new iterative detection receiver. This approach first utilizes parallel interference cancellation to detect signals of all layers, while the detected signals are regarded as input of channel decoder. For improving the performance of allover system, the output of decoder is regarded as input of PIC detection to do PIC again. By exchanging information between the MIMO detection and decoder, the performance of receiver may greatly be enhanced. Computer simulation result states the performance of proposed detection scheme is better than conventional coded MIMO-OFDM system.

The rest of this paper is organized as follows. In Section 2, we describe MIMO-OFDM systems model. In Section 3, a joint iterative detection and decoding scheme is proposed for MIMO-OFDM systems based on parallel interference cancel (PIC). The simulation results and performance analysis are presented in Section 4 and 5. Concludes follow in Section 6.

## 2. MIMO-OFDM Systems Model

Before introducing the signal detection, we briefly describe a MIMO-OFDM system. The combination of OFDM and VBLAST can overcome intersymbol interference in frequency selective fading channels. A multicarrier system can be efficiently implemented in discrete time using an inverse FFT (IFFT) to act as a modulator and an FFT to act as a demodulator. The VBLAST architecture is based on a single carrier signal processing algorithm. Therefore, to combine it with OFDM, the VBLAST detection process has to be performed on every subcarrier at the receiver. The detailed system configuration of the VBLAST-OFDM is shown in Figure 1−2.

### 2.1. MIMO-OFDM Systems

In this section, we consider a coded MIMO-OFDM communication system with $n_T$ transmit antennas and $n_R$ receive antennas, denoted by $(n_T, n_R)$. Figure 1 is diagram of MIMO-OFDM transmitter. At the transmitter the input bit stream is de-multiplexed and coded to generate $n_T$ symbol streams. The encoded data stream is then interleaved and launched into the IFFT modulators and added cyclic prefix (CP). Finally, the OFDM signals are transmitted over every transmit antenna.

Figure 2 shows the block diagram of a VBLAST-OFDM receiver. Each receiver antenna receivers signals sent from all transmit antennas. After the cyclic prefix is removed, each received signal passes through a FFT block for demodulation.

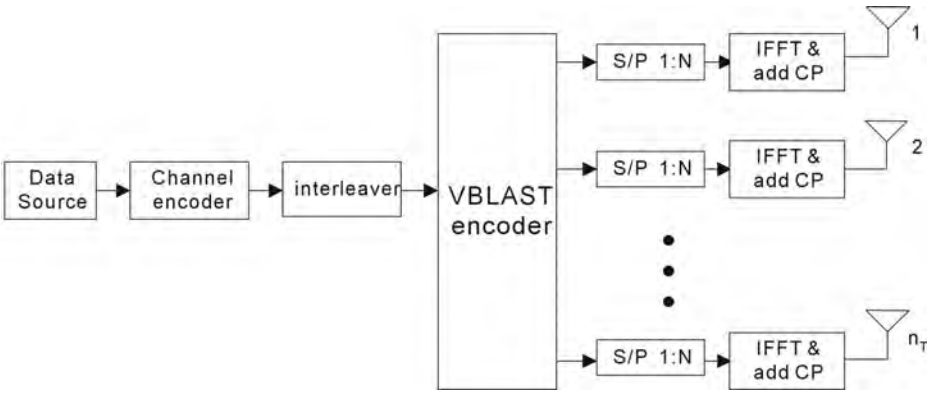
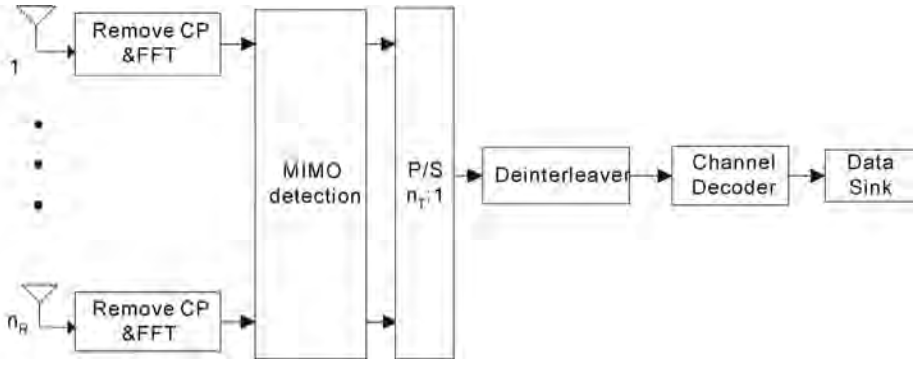



**Figure 1. VBLAST-OFDM transmitter.**




**Figrue 2. VBLAST-OFDM receiver.**

Copyright © 2009 SciRes. 

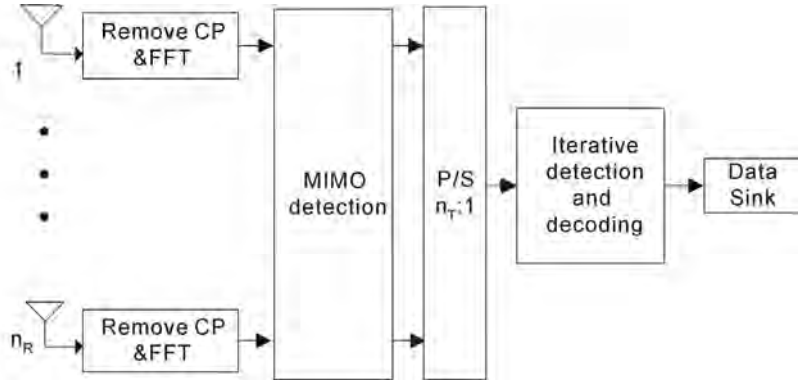



**Figure 3. Proposed receiver structure for MIMO-OFDM system.**

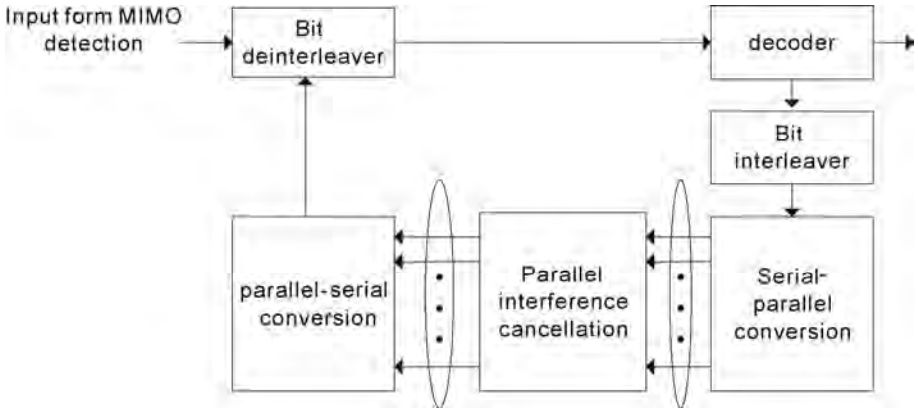



**Figure 4. Proposed detection block.**

At the receiver, we assume perfect OFDM synchronization. The receiver signal after demodulation, at receive antenna $j$ for subchannel $k$, is given by

$$r_j[n,k] = \sum_i H_{ij}[n,k]s_i[n,k] + w_j[n,k] \quad \textbf{(1)}$$

where $i = 1,\cdots,n_T$, $j = 1,\cdots,n_R$, $s_i[n,k]$ is the transmitted symbol from the i-th transmit antenna at k subchannel, $w_j[n,k]$ in (1) denotes the additive complex Gaussian noise at the $j$th receiver antenna, and is assumed to be zero-mean with variance $\sigma_n^2$. $H_{ij}[n,k]$ in (1) denotes the channel frequency response for the $k$th tone at time $n$, corresponding to the $i$th transmit and the $j$th receiver antenna.

During the reception, each receiver antenna receives the signal transmitted from all the $n_T$ transmit antennas. First, the cyclic prefix of each received signal is removed. After passing through a serial-to-parallel converter and the fast Fourier transformation blocks (FFTs), the subcarriers are separated. Then, the N information symbols belonging to each subcarrier are routed to their corresponding VBLAST multi-antenna processing unit where the de-mapping and decoding are performed. The detected bits are converted back into a serial form accordingly to recover the transmitted data bits.

## 2.2. Channel Models

We assume that the OFDM signal is transmitted over a wireless communications environment by a mutipah fading channel and a given coherence bandwidth. The complex baseband equivalent of a fading channel impulse response from transmit antenna $i$ to receiver antenna $j$ can be expressed as,

$$h_{ij}(t,\tau) = \sum_{l=1}^{L} \alpha_{ij}^l(t)\delta(\tau - \tau_l) \quad \textbf{(2)}$$

where $\alpha_{ij}^l(t)'s$ are wide-sense stationary narrowband complex Gaussian processes and are assumed to be independent among different paths form transmit antenna $i$ to receiver antenna $j$. L is the number of multipaths. Thus for each receive antenna, the channel frequency response for the $k^{\text{th}}$ subcarrier at time n is

$$H_{ij}[n,k] = \sum_{l=0}^{L-1} \alpha_{ij}^l(t)e^{-j2\pi k \Delta f \tau_l} \quad \textbf{(3)}$$

Copyright © 2009 SciRes. 

where $\triangle f = \frac{B}{K}$ is the subcarrier spacing, $B$ is the total bandwitdth, and $K$ is the total number of subcarriers.

## 3. Proposed Scheme

To the above the VBLAST-OFDM system model, we proposed the new detection scheme for MIMO channel matrix at every sub-carrier. To reduce the complexity and prevent error propagation, the MIMO PIC detection is applied to obtain the initial estimation of transmitting signals for MIMO-OFDM architecture. Furthermore, iterative detection and decoding is used to enhance the performance of system in a progressive fashion.

Figure 3 presents a simple schematic of the proposed receiver scheme with PIC MIMO detection for MIMO-OFDM systems. For a wideband system, the OFDM demodulator is applied for each transmitter stream over $N$ parallel subchannels. To perform the iterative detection and decoding processing, the estimated signal by decoder is used to reconstruct the transmitted coded signal. Consequently, the PIC detection uses the reconstructed signal to improve the detection performance and start the iterative process. Figure .4 illustrates the proposed iterative detection and decoding block in Figure 3 in detail.

The PIC in [11] is another conventional detection algorithm which detects all layers simultaneously by subtracting interference from all the other layers regenerated by the estimate from hard decision based on ZF or MMSE criteria. The PIC detection scheme based on MIMO system algorithm is described as below:

Input $\boldsymbol{H}$ , $\boldsymbol{r}$

$$\tilde{\boldsymbol{s}} = \boldsymbol{H}^{+}\boldsymbol{r}$$

$$\tilde{\boldsymbol{s}} = \mathrm{Q}(\boldsymbol{G}\,\boldsymbol{r})$$

For i=1: $n_T$

$$r = r - \sum_{\substack{j=1 \\ j\neq i}}^{n_T-1} H(:, j)\cdot \tilde{s}_j$$

$$\boldsymbol{G} = (\boldsymbol{H}(:, j))^{+}$$

$$\tilde{s}_j = Q(\boldsymbol{G}\cdot r)$$

End

Output $\tilde{\boldsymbol{s}}$

where $(\bullet)^{+}$ is operation of calculating inverse of matrix (or vector). Comparing to conventional VBLAST, the PIC algorithm doesn't need calculate the pseudo-inverse matrix of the channel, so the complexity of proposed receiver is reduced. However, the detection scheme doesn't obtain the gain with applying the odering of the layers. Below sections, the performance of proposed scheme is study.

## 4. Simulations Results

In this section, we investigate the characteristics of our proposed receiver through computer simulations. We assume a perfect channel estimation and synchronization at the receiver.

In simulations, BPSK is used as the subcarrier modulation. We evaluate the performance of MIMO-OFDM systems. The FFT size is 64, and the cyclic prefix length is L=16. For the 20 MHz channel, channel assumes an exponentially decaying power delay profile with 4 multipaths which are sample-spaced, and assume the channel is constant during one packet. In the simulation, the antenna configuration consists of 4 transmit antennas, and 4 receive antennas and BPSK is used. For comparison, we also include the performance of V-BLAST and the performance of conventional PIC receiver in signal detection.

We first evaluate the performance of PIC algorithm in MIMO-OFDM without coding case. Figure 5 shows the comparison performance of the three detection methods. Simulation states that PIC detection scheme obtains about 6dB gain than linear ZF method at error bit rate $10^{-3}$. At low SNR case, the PIC detection scheme outperforms the conventional V-BLAST detection. At high SNR, the advantage of the proposed scheme died down because the affect of error diffuse is taken off. The performance of conventional ZF-VBLAST detection is the best when the SNR is higher than 8 dB. The linear ZF has the worst performance.

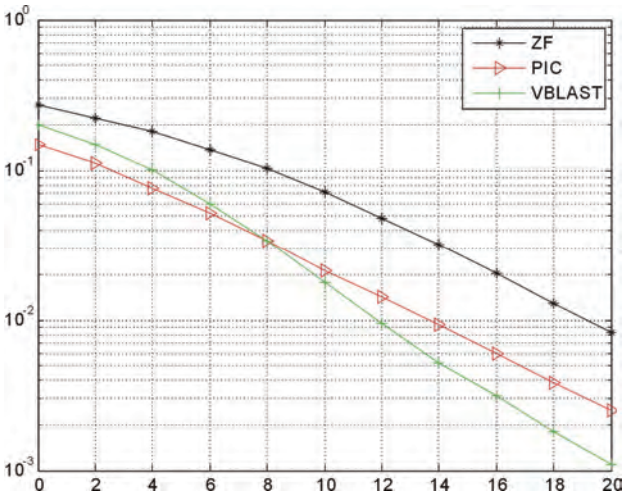



**Figure 5. BER comparison for the (4,4) MIMO-OFDM system with difference detection methods.**

Copyright © 2009 SciRes. 

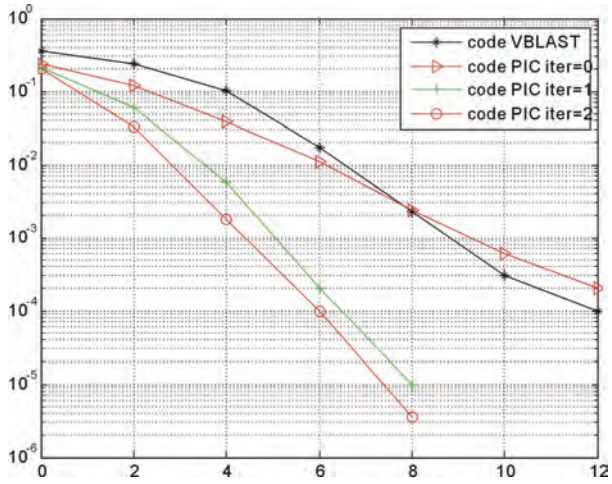



**Figure 6. BER comparison for the (4,4) code MIMO-OFDM system with difference detection methods.**

Figure 6 shows the performance of proposed iterative detection and decoding scheme for code MIMO-OFDM systems. The convolutional code used in the simulations is a rate $1/2$, and the generator vectors are $g_0 = (1,0,11)$ and $g_1 = (1,1,0,1)$. The channel decoder uses the Viterbi algorithm. The other conditions is the same as to above case. As can be shown in this plot, the proposed MIMO detection scheme greatly outperforms the conventional linear Zero-Forcing detection.

The coded BER of proposed scheme is produced after 0, 1, 2 iterations. When iteration is 0, the proposed scheme is the conventional PIC algorithm for code MIMO-OFDM systems. We also observe that the proposed scheme outperforms the conventional coded PIC detection algorithm without iteration. After first iteration, the BER is significantly improved. There is a litter difference between the first iteration and the second iteration. Simulation states that proposed scheme with 2 iterations obtains about 4dB gain than conventional MIMO-VBLAST algorithm without iteration at error bit rate 10-3.

From the simulation results presented in this section, the proposed scheme is quite effective in all simulation configurations. Below we furthermore analyze performance of proposed.

In actual, the difference algorithms have difference diversity order. For linear ZF receiver, the MIMO system may obtain $n_R - n_T - 1$ order diversity gain and the conventional VBLAST algorithm may obtain $\geq n_R - n_T - 1 \leq n_R$ order diversity gain [12]. However, PIC detection scheme may obtain $n_R$ order diversity gain when the interference comes from the all other layers is completely cancelled. Below we mainly analyze effect error propagation for our proposed detection scheme.

## 5. Conclusions

In this paper, we proposed a iterative detection and decoding scheme with PIC algorithm for MIMO-OFDM systems. To demonstrate the potential of this proposed technique, we have investigated the PIC and conventional VBLAST algorithms in MIMO-OFDM systems. We have also studied the effect of error forward coding on the system. By computer simulation, the performance of the proposed scheme is evaluated. The simulation results states that the performance of proposed scheme is greatly improved compared to conventional VBLAST detection receiver for MIMO-OFDM systems.

## 6. References

[1] P. W. Wolniansky, G. J. Roschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," Proceedings International Symposium on Signals, Systems, and Electronics (ISSSE'98), Pisa, Italy, pp. 230−235, September 1998.

[2] D. Wubben, J. Rinas, R. Bohnke, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for detecting layered space-time codes," Proceedings ITG Conference on Source and Channel coding, Berlin, Germany, pp. 399−405, January 2002.

[3] B. Hassibi, "An efficient square-root algorithm for blast," in Proceedings IEEE International Conference Acoustic, Speech, Signal Processing, Istanbul, Turkey, pp. 5−9, June 2000.

[4] W. Zhu, J. Y. Jin, and Y. W. Park, "Detection algorithm improving parallel interference cancellation for V-BLAST system over error propagation," Proceedings ICACT2006, Gangwon-Do, Korea, pp. 11−15, February 2006.

[5] X. Li and X. Cao, "Low complexity signal detection algorithm for MIMO-OFDM systems," Electronics letters, Vol. 41, No. 2, pp. 20−21, January 2005.

[6] S. Haykin, M. Sellathurai, Y. D. Jong, and T. Willink, "Turbo-MIMO for wireless communications," IEEE Communications Magazine, Vol. 42, No. 10, pp. 48−53, October 2004.

[7] M. Sellathurai and S. Haykin, "Turbo-BLAST for wireless communications: Theory and experiments," IEEE Transaction Signal Processing, Vol. 50, No. 10, pp. 2538−2546, October 2002.

[8] C. Berrou, A. Glavieum, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-Codes (1)," Proceedings of ICC'93, pp. 1064−1070, May 1993.

Copyright © 2009 SciRes. 

[9] I. Sohn and J. Y. Ahn, "Joint processing of zero-forcing detection and MAP decoding for a MIMO-OFDM system," ETRI Journal, Vol. 26, No. 5, October 2004.

[10] Z. P. Wang, "A group iterative QR receiver based on flat MIMO channels," in Proceeding IEEE WICOM2007, Shanghai, China, pp. 408–412, September 2007.

[11] W. H. Chin, A. G. Constantinides, and D. B. Ward, "Parallel multistage detection for multiple antenna wireless systems," IEEE Electronic letters, Vol. 38, No. 12, pp. 597–599, June 2002.

[12] A. Paulraj, R. Nabar, and D. Gore, "Introduction to space-time wireless communications," Cambridge University Press, pp. 153, 2003.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 357-362
doi:10.4236/ijcns.2009.25039 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Research on Error's Distribution in Triangle Location Algorithm

**Jian ZHU, Hai ZHAO, Jiuqiang XU, Yuanyuan ZHANG**
*The Northeast University, Shenyang, China*
*Email*: *zhujian*1981710@163.*com*
*Received July* 18, 2008; *revised February* 18, 2009; *accepted March* 27, 2009

## ABSTRACT

The error will inevitably appear in triangle location in WSNs (wireless sensor networks), and the analysis of its character is very important for a reliable location algorithm. To research the error in triangle location algorithm, firstly, a triangle location model is set up; secondly, the condition of the minimal error is got based on the analysis of the location model; at last, by analyzing the condition, the law of error's changing is abstracted, and its correctness is validated by some testing. The conclusions in this paper are all proved, and they provide a powerful foundation and support for reliable location algorithm's research.

**Keywords:** Localization, Error Distribution, Triangle, MicaZ, WSNs

## 1. Introduction

The information usually binds with the node's position in pervasive computing [1,2] as approximately 80% information is related with position in this computation [3]. Therefore, the location study becomes an essential topic in pervasive computing [4]. Nowadays, most existing location algorithms [5−7] contain two basic steps: 1) distance (or angle) measurement; 2) location computation.

Currently, the triangle location is a hot research topic [8,9]. Because of the unavoidable location error, the key problem of reliable location is how to decrease the error as much as possible. But many literatures about the triangle location are lack of researches on the change law of location error, and they did not consider the error's distribution in a triangle. This article draws a conclusion: the error's change in triangle is stochastic, and the errors in different position are different. If a location algorithm has not considered the error's changing law, its application scope will be confined or even can not be applied at all. So, the analysis of error is very important for a reliable triangle location algorithm.

## 2. Mathematical Model of Location Question

In the three-dimensional space, it can determine the coordinate of a location point according to the distance from the location point to four reference points, and this is the same with the basic principle of global positioning system (GPS) [10]. In pervasive computing, however, the most coordinate systems are two-dimensional space and they also do not need the clock synchronization. Therefore, it can fix on the position of a location point as long as the distance from this location point to three reference points is known. As shown in Figure 1, the basic principle of trilateral location is the solution of three circles' intersection whose radii and the coordinate of circle center are known.

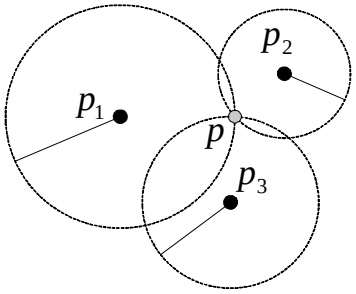



**Figure 1. Three-edge measurement location.**

Copyright © 2009 SciRes. 

Due to the limit of pervasive terminal's hardware and energy consumption, the ranging error between nodes is usually biggish. Here, it supposes the ranging error's scope is in (0, ±ε) and takes (ε>0). So three circles are no longer intersect at a point, but constitute a small region, denoted by $C_p$. Supposes the coordinate of the point to be located p is (x, y), then it can locate this point and get three reference points denoted by $p_i$ which constitutes the triangle. The coordinate of $p_i$ is respectively $(x_i, y_i)$, i=1, 2, 3, the distance to the location point is $r_i$, and $\varepsilon_i$ is the ranging error, makes $Cp_i$, $C_p$ and $S_p$:

$$C_{p_i} = \{(x,y) \mid (x-x_i)^2 + (y-y_i)^2 \le (r_i+\varepsilon_i)^2, (x-x_i)^2 + (y-y_i)^2 \ge (r_i-\varepsilon_i)^2\} \quad \textbf{(1)}$$

$$C_p = \{(x,y) \mid x \in \bigcap_{i=1}^{3} C_{p_i}, y \in \bigcap_{i=1}^{3} C_{p_i}\} \quad \textbf{(2)}$$

$$S_{p_i} = \{(x,y) \mid x^2 + y^2 = {\varepsilon_i}^2, \varepsilon_i > 0\} \quad \textbf{(3)}$$

In order to simplify the analysis, supposes ranging error is equal, namely $\varepsilon_i=\varepsilon$, i=1,2,3, then, when $\varepsilon\equiv 0$, the set of points $C_p$ in Formula (2) will assemble into one point (e.g. Figure 1); However, when $\varepsilon>0$, $C_p$ will be a convex small region, and its area represents location error size, as shown in Figure 2.

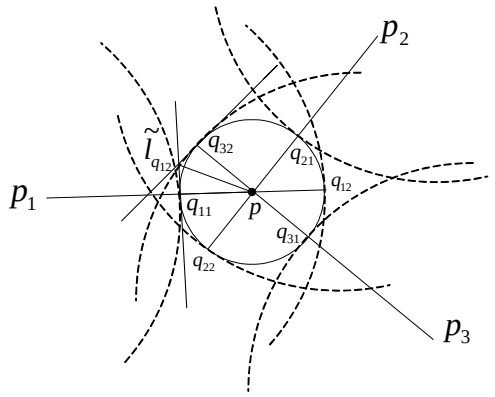



**Figure 2. The analysis of location error.**

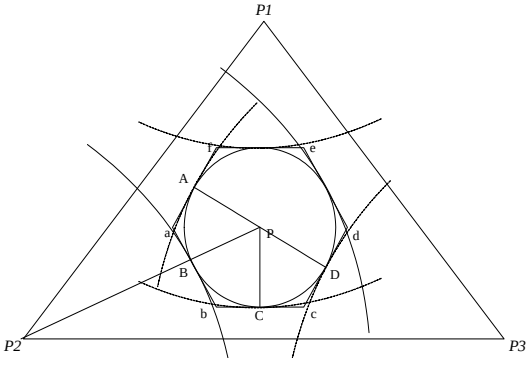



**Figure 3. The area of location error.**

In the figure above, supposes $lpp_i$ is a straight line which pasts point p, $p_i$ and intersects the circle $S_p$ in Formula (3) at two points $q_{ij}$, j=1, 2. Make tangent $\tilde{l}q_{ij}$ for Sp pasting $q_{ij}$, then the tangent will intersect as arbitrary hexagon like Figure 3, and region $\tilde{C}p_i$ is between the straight line $\tilde{l}q_{i1}$ and $\tilde{l}q_{i2}$. Take $\tilde{C}p=\tilde{C}p_1\cap\tilde{C}p_2\cap\tilde{C}p_3$, then the final $\tilde{C}_p$ will be the hexagon abcdef in Figure 3. The area of $C_p$ edge can be linearized as measuring error ε is less, and it will be estimated as $\tilde{C}_p$, namely $C_p$ will approximately be regard as $C_p$, it may also find in Figure 3 that the $C_p$ region cut by arc is almost the same with hexagon region $\tilde{C}_p$. Thus, the node location question in the two-dimensional space is transformed into discussing how to arrange three reference points for minimizing the area S ($\tilde{C}_p$) of region $\tilde{C}_p$.

## 3. Condition of Minimum Location Error

Lemma 1: Supposes a, b, c is three numbers whose value is bigger than 0, then $a+b+c \ge \sqrt[3]{abc}$ and the equal sign holds only when a=b=c.

Supposes $\alpha_1$, $\alpha_2$, $\alpha_3$ is respectively the angle (central angle) between vector PA and PB, PB and PC, PC and PD. The region constituted by $\tilde{C}_p$ is a circumscribed hexagon of circle $S_p$ as shown in Figure 3, then

$$S(\tilde{C}_p) = 2\varepsilon^2(\tan\frac{\alpha_1}{2} + \tan\frac{\alpha_2}{2} + \tan\frac{\alpha_3}{2}) \quad \textbf{(4)}$$

As to $\alpha_1$, $\alpha_2$, $\alpha_3$, it has the following relationship:

$$\alpha_1 + \alpha_2 + \alpha_3 = \pi$$

when $0\le x\le\pi/2$, then $(\tan x)''=2\tan x(1+\tan x)\ge 0$, and we may derive the following expression using Lemma 1:

$$S(\tilde{C}_p) = 6\varepsilon^2 * \frac{1}{3}(\tan\frac{\alpha_1}{2} + \tan\frac{\alpha_2}{2} + \tan\frac{\alpha_3}{2}) \ge 6\varepsilon^2 * \sqrt[3]{\tan\frac{a_1}{2}\tan\frac{a_2}{2}\tan\frac{a_3}{2}} \quad \textbf{(5)}$$

The equality above holds only when $\alpha_1 = \alpha_2 = \alpha_3=\pi/3$, namely conclusion 1: S ($\tilde{C}_p$) can obtain the minimum value only when $\alpha_1 = \alpha_2 = \alpha_3$.

As $\alpha_1 = \alpha_2 = \alpha_3=\pi/3$, it is not difficult to deduce according to the geometry principle that the angle of three linked lines formed by three reference points and a point to be located P is mutually 120°. Therefore, only when the angle of three linked lines formed by three reference points and the point to be located is mutually 120°, error zone $\tilde{C}_p$ can become the circumscribe hexagon of the circle. In this case, the location error of the location point can achieve minimum, and this conclusion is the so-called formation condition of the minimum location error.

As shown in Figure 4,in the triangle which formed by three arbitrary arranged reference points, there is only

Copyright © 2009 SciRes. 

one error zone of the point to be located presents hexagon at most, and the location error increases along with the distance from this point to point P. For instance, as the location point is S and its error zone is an anomalous hexagon; its error zone area will be obviously bigger than the error zone area of point P.

## 4. The Change Law of Location Error

Take Figure 4 as an example, after the offset of arbitrary point (S) and the minimum error point (P), all three angles formed mutually by point S and three reference points of the triangle will change. Supposes $\alpha_4$, $\alpha_5$, $\alpha_6$ is respectively the angle formed by intersection of vector Sa and Sb, Sb and Sc, Sc and Sd. According to computation, the location error zone area of point S is:

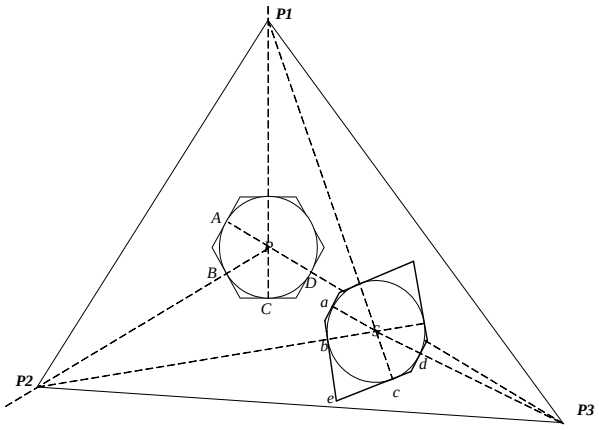



**Figure 4. The location error of random position.**

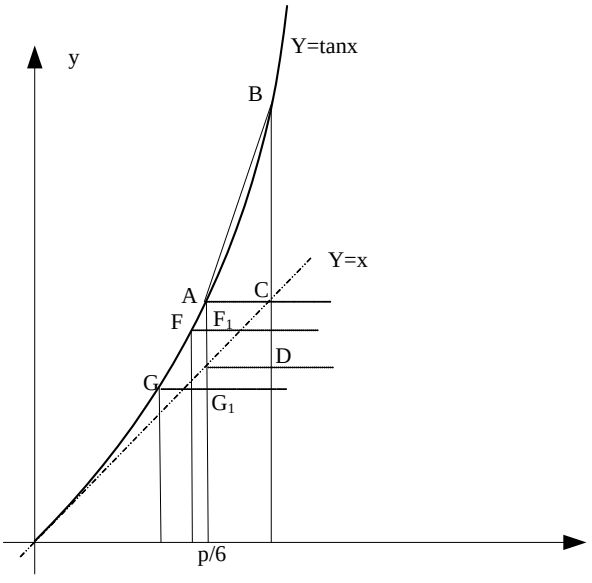



**Figure 5. The rule of location error's increase.**

$$S(\tilde{C}_S) = 2\varepsilon^2(\tan\frac{\alpha_4}{2} + \tan\frac{\alpha_5}{2} + \tan\frac{\alpha_6}{2}) \quad \textbf{(6)}$$

After the change of point S, $\alpha_4$, $\alpha_5$ and $\alpha_6$ are all changed compared with $\alpha_1$, $\alpha_2$, $\alpha_3$ in the Formula (4), supposed these three angles' variation is $\Delta\alpha_4$, $\Delta\alpha_5$, $\Delta\alpha_6$, then there must be increment and decrement among them, and the sum of these three variation should be 0 .meanwhile, the size of $\Delta\alpha_4$, $\Delta\alpha_5$, $\Delta\alpha_6$ is decided by the angle which is formed mutually by point S and three reference points. For instance, angle $P_1SP_2$ in Figure 4 is equal to the sum of $\alpha_4$ and $\alpha_6$, it means the sum of $\alpha_4$ and $\alpha_6$ will be smaller as angle $P_1SP_2$ gets smaller, and as the sum of $\alpha_4$, $\alpha_5$ and $\alpha_6$ is 180° based on the preceding text, the value of $\alpha_5$ will be bigger.

The angle of point S and three reference points is not equal to 120°, and $\alpha_4/2$, $\alpha_5/2$ and $\alpha_6/2$ is not equal to 30°. $\Delta\alpha_5$ in Figure 4 is increment, and the other two are decrement. The error zone area can reach its minimum value when $\alpha_1/2$, $\alpha_2/2$ and $\alpha_3/2$ are all equal to 30° in terms of the Formula (5). For the quantitative investigation on relationship between the central angle variation of hexagon and the error zone area, this article has made the following deduction:

$$\tan' x = \frac{1}{\cos^2 x} \quad \textbf{(7)}$$

The Formula (7) is the derivative formula of tangent function, when x is 0, the tangent derivative value is 1, namely when tangent function in (0, 0) the tangent slope is the same with the slope of y=x. Tangent slope of tangent function will be bigger than 1 if $x \in (0, \pi/2)$, and it increases along with the increase of x, as shown in Figure 5.

In Figure 5, A is a point whose abscissa is $\pi/6$ on the tangent function curve, and the abscissa of point B is equal to a radian corresponds to $\alpha_5$ in Formula (6); the ordinate is tan ($\pi/6+\Delta\alpha_5$). BC is the variation of tangent function corresponds to $\Delta\alpha_5$, makes the slope of straight line AB as $K_{AB}$, then $K_{AB}>K_{FA}>K_{GA}$. Since $BC=\Delta\alpha_5 K_{AB}$, $AF_1=\Delta\alpha_4 K_{AF}$ and $AG_1=\Delta\alpha_6 K_{AG}$, supposes $BC>AF_1+AG_1$, then after the angle in Formula (4) change into the angle in Formula (6), the variation of the error zone area is:

$$\Delta S(\tilde{C}_S) = 2\varepsilon^2(BC - AF_1 - AG_1) \quad \textbf{(8)}$$

Because:

$$AF_1 + AG_1 = \Delta a_4 K_{AF} + \Delta a_6 K_{AG} < \Delta a_4 K_{AB} + \Delta a_6 K_{AB}$$
$$= K_{AB}(\Delta a_4 + \Delta a_6) = K_{AB}\Delta a_5 = BC$$

So the supposition is correct, namely $BC>AF_1+AG_1$. After central angle's relative minimum error point (P) was changed, the error zone area will increase, and this has confirmed the conclusion 1 in Formula (5). It can be

Copyright © 2009 SciRes. 

also seen from the figure above, BC is to be bigger if $\Delta\alpha_5$ enlarged, meanwhile, it deduces that along with the increase of $\alpha_5$ the error zone area will also increase, the deduce process is as follows:

Supposes $\Delta\alpha_4 \leq \Delta\alpha_6$, then $K_{AG} \leq K_{AF}$, therefore:

$$\begin{aligned} &AF_1 + AG_1 = \Delta a_4 K_{AF} + \Delta a_6 K_{AG} \\ &\leq \Delta a_6 (K_{AG} + K_{AF}) \leq \Delta a_6 K_{AF} \end{aligned} \quad \textbf{(9)}$$

In the above expression, $AF_1+AG_1 = \Delta\alpha_6 K_{AF}$ only when $\Delta\alpha_4 = \Delta\alpha_6$ and $K_{AG}=K_{AF}$, and $\Delta\alpha_6 = \Delta\alpha_5/2$ holds at this time according to the geometry principle, so the variation of error zone area here can express as:

$$\begin{aligned} &\Delta S(\tilde{C}_S) = 2\varepsilon^2 (BC - AF_1 - AG_1) \\ &\geq 2\varepsilon^2 (K_{AB}\Delta a_5 - K_{AF}\frac{\Delta a_5}{2}) = 2\Delta a_5 \varepsilon^2 (K_{AB} - K_{AF}/2) \end{aligned} \quad \textbf{(10)}$$

In the above expression, as $\Delta\alpha_5$ gets bigger, $K_{AB}$ increases, $K_{AF}$ reduces, then we may get from the tangent function characteristic that the difference of $K_{AB}$ and $K_{AF}/2$ will increase along with $\Delta\alpha_5$, and the variation of error zone area will increase if $\Delta\alpha_5$ enlarged.

Therefore, it can draw the conclusion 2: after an arbitrary location point offset with the minimum location error point P, once one or two central angles' degree increase which among 6 central angles of the new formed hexagon, the error zone will be bigger along with the increase amplitude of the central angle, and $\lim_{a\to\pi} \Delta S(\tilde{C}_S) = +\infty$.

Synthesizes the above conclusion and Figure 4, it can be seen that when $\alpha_5$ approaches to $\pi$, straight line be will be parallel with straight line ce, and the area of quadrangle Sbec will be close to infinity. Therefore, the final error zone area will be also close to infinitely.

## 5. Application of the Location Error Change Law

According to the conclusion above, since distribution of the location error in a triangle is stochastic, it will definitely appear the maximum location error and the minimum location error in a triangle. The minimum location error in a triangle can only appear on one point, whereas the maximum location error influences positioning system's accuracy directly, as a result, this article has conducted a research to the layout shape of the reference point in view of the maximum location error.

As shown in Figure 6, there are two different kinds of reference point layout schemes under the premise of the same coverage area.

P and $P_1$ is respectively minimum error point in two triangles, as the point to be located S in $\Delta$ABC approaches to point A, angle BSC can only reduce to 60° at most, and according to the relation between central angle and included angle in Figure 4, since point S arrives at

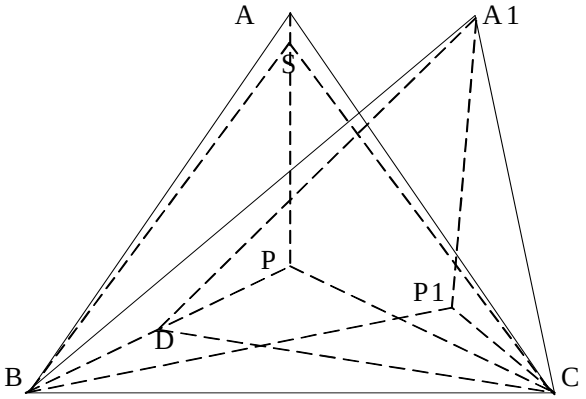



**Figure 6. The two way of reference nodes' layout.**

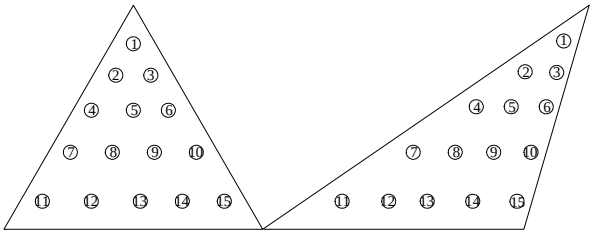



**Figure 7. Two way of reference node's layout.**

point A, three central angles will be respectively 30°,30°,120°; as the point D in $\Delta A_1BC$ approaches to point B, angle $A_1DC$ can reduce to the value which is smaller than 60°, it is not difficult by the geometry principle to deduce that if point D arrives at point B, three central angles will be respectively smaller than 30°, smaller than 30°, bigger than 120°, and the location error of point A in $\Delta$ABC is bigger than the location error of point B in $\Delta A_1BC$ by conclusion 2. Point A is the maximum error point in $\Delta$ABC, and the error of point B in $\Delta A_1BC$ is bigger than the maximum error in $\Delta$ABC, so this article obtains the inference 1: The equilateral triangle positioning system's accuracy is higher than non-equilateral triangle positioning system under the same condition.

In order to verify the above deduction, this article designs an experiment as following figure. Under the platform of MicaZ node, laying the point to be located uniformly in the region encircled by three reference points, the layout graph is as follows:

The settings of nodes' parameters are as table 1:

The location error of each point to be located is shown in Figure 8, and it can be seen that as the reference point is an equilateral triangle, the maximum location error will be about $0.14m^2$; when the reference point is arranged as a non-equilateral triangle, its maximum location error will be $0.153m^2$ as seen in Figure 8, so the maximum location error is higher than the equilateral

Copyright © 2009 SciRes. 

**Table 1. The detailed parameters of nodes in NS-2.**

| Parameter | Value |
|---|---|
| Nodes distribution | As Figure 7 |
| Channel bandwidth | 250kbps |
| Frequency | 2.4GHz |
| Wireless model | Shadowing |
| MAC protocol | IEEE 802.15.4 |
| Power for transmission (mwatt) | 2.0 |
| Power for idle (mwatt) | 1.0 |
| Power for reception(mwatt) | 1.0 |
| Power for sleep (mwatt) | 0.001 |
| Power for sleep/idle transition (mwatt) | 0.2 |
| Time for sleep/idle transition (s) | 0.005 |

triangle obviously. Therefore, inference 1 is correct, and it also indicates the analysis on the error change law in this article conforms to reality.

An experiment is designed to research the distribution of error in a triangle, the platform is the same with the above, and the result is shown in Figure 9:

In Figure 9, the axis X and Y construct a 100m×100m area, $\Delta ABO$ is in the area; Axis Z is the precision of location in a triangle. It is shown in Figure 9 that, the precision in the center of a triangle is higher than the fringe areas. So, while in applications, it is better that, keep the unknown nodes in the center area of the three reference nodes.

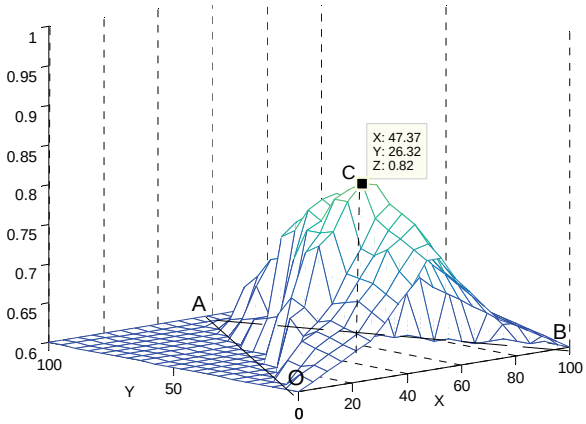



**Figure 9. The precision in the center of a triangle.**

## 6. Conclusions

This article mainly studied the formation condition of minimum error and the changing law of error on location point in arbitrary triangle according to the theory deduce. The change law of error was verified by practical test, and the research results indicate that the change law of error studied in this article may provide basis and theory instruction for the high performance and reliable location algorithm.

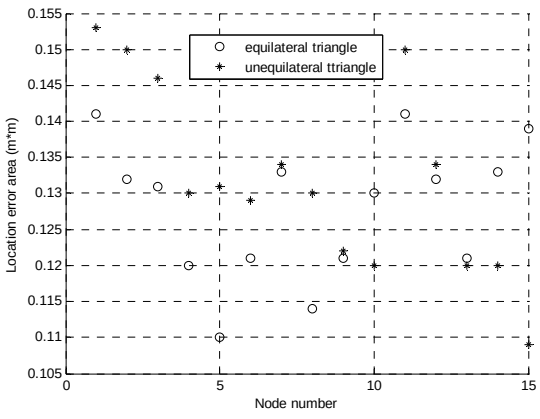



**Figure 8. The infection of the largest location error.**

## 7. References

[1] W. K. Edwards, "Discovery systems in ubiquitous computing," IEEE Pervasive Computing [J], Vol. 5, No. 2, pp. 70–77, 2006.

[2] R. Milner, "Ubiquitous computing: Shall we understand it? [J]," Computer Journal, Vol. 49, No. 4, pp. 383–389, 2006.

[3] J. Y. Junq, "Human-centred event description for ubiquitous service computing [A]," 2007 International

Copyright © 2009 SciRes. 

Conference on Multimedia and Ubiquitous Engineering [C], pp. 1147−1151, 2007.

[4] D. Noh and H. Shin, "An effective resource discovery in mobile ad-hoc network for ubiquitous computing [J]," Journal of KiSS: Computer Systems and Theory, Vol. 33, No. 9, pp. 666−676, 2006.

[5] G. M. Huang, "A novel TDOA location algorithm for passive radar [C]," CIE international Conference of Radar Proceedings, pp. 41484−41888, 2007.

[6] G. V. Zaruba, "Indoor location tracking using RSSI readings from a single Wi-Fi access point [J]," Wireless Networks, Vol. 13, No. 2, pp. 221−235, 2007.

[7] J. Su, "Location technology research under the environment of WLAN [J]," WSEAS Transactions on Computers, Vol. 6, No. 8, pp. 1050−1055, 2007.

[8] J. Ding, L. R. Hitt, and X. M. Zhang, "Markov chains and dynamic geometry of polygons [J]," Linear Algebra and Its Applications, pp. 255−270, 2003.

[9] J. Ding, L. R. Hitt, and X. M. Zhang , "Sierpinski pedal triangles [J]," Linear Algebra and Its Application, pp. 1−15, 2005.

[10] J. Hightower, G. Borriello, and R. Want, "An indoor 3D location sensing technology based on RF signal strength [R]," Technical Report, University of Washington, Seattle WA, pp. 1−16, 2000.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 363-370
doi:10.4236/ijcns.2009.25040 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Authentication and Secret Message Transmission Technique Using Discrete Fourier Transformation

**Debnath BHATTACHARYYA[1], Jhuma DUTTA[1], Poulami DAS[1], Samir Kumar BANDYOPADHYAY[2], Tai-hoon KIM[3]**
[1]*Computer Science and Engineering Department, Heritage Institute of Technology, Kolkata, India*
[2]*Department of Computer Science and Engineering, University of Calcutta, Kolkata, India*
[3]*Hannam University, Daejeon, Korea*
*Email*: {*debnathb, jhumadutta*81, *dasp*88}*@gmail.com, skb*1*@vsnl.com, taihoonn@empal.com*
*Received April* 2, 2009; *revised May* 9, 2009; *accepted June* 28, 2009

## ABSTRACT

In this paper a novel technique, Authentication and Secret Message Transmission using Discrete Fourier Transformation (ASMTDFT) has been proposed to authenticate an image and also some secret message or image can be transmitted over the network. Instead of direct embedding a message or image within the source image, choosing a window of size 2 x 2 of the source image in sliding window manner and then convert it from spatial domain to frequency domain using Discrete Fourier Transform (DFT). The bits of the authenticating message or image are then embedded at LSB within the real part of the transformed image. Inverse DFT is performed for the transformation from frequency domain to spatial domain as final step of encoding. Decoding is done through the reverse procedure. The experimental results have been discussed and compared with the existing steganography algorithm S-Tools. Histogram analysis and Chi-Square test of source image with embedded image shows the better results in comparison with the S-Tools.

**Keywords:** Data Hiding, Authentication, Frequency Domain, Discrete Fourier Transformation (DFT), Inverse Discrete Fourier Transform (IDFT), S-Tools

## 1. Introduction

The most popular technique for image authentication or steganographic technique is embedding message or image within the source image, generally termed data hiding. It provides secret message transmission over the communication channel. Moreover several techniques are available for secret message transmission by hiding a message inside an image without changing its visible properties. Although it changes source, instead of direct embedding message or image within the source image, the embedding is done in the frequency domain.

The presented work deals on information and image protection against unauthorized access in frequency domain. A picture in the spatial domain can be described as a collection of pixel values describing the intensity values. The DFT changes an *N* point input signal into two point output signals. The input signal contains the *N*/2 –1 signal being decomposed, while the two output signals contain the *amplitudes* of the component sine and cosine waves. The input signal is said to be in the time domain. This is because the most common type of signal in the Discrete Fourier Transformation (DFT) is composed of samples taken at regular intervals of *time*. Any kind of sampled data can be fed into the DFT, regardless of how it was acquired. The frequency domain signal is represented by a vector F [u,v], and consists of two parts, for each of the samples. These are called the Real part of F [u,v] written as: *ReF* [u,v], and the Imaginary part of F [u,v], written as: *ImF* [u,v]. In the sample "real part" means the *cosine wave amplitudes* while "imaginary part" means the *sine wave amplitudes*. The formula of DFT for a function f (x, y) of size M x N is given in Equation 1 for frequency domain transformation.

Copyright © 2009 SciRes. 

$$F(u,v)=\frac{1}{\sqrt{MN}}\sum_{x-0}^{M-1}\sum_{y-0}^{N-1}f(x,y)\cos\left(\frac{2\pi ux}{M}\right)-f(x,y)\,j\sin\left(\frac{2\pi vy}{N}\right)$$
$$\text{for } u=0,1,\ldots M-1 \quad v=0,1,\ldots N-1 \tag{1}$$

For the cause of the proposed algorithm the simpler from of Equation (1) is as given in Equation (2)

$$F(u,v)=\mathrm{Re}\,F(u,v)-\mathrm{Im}\,F(u,v)$$
$$\text{for } u=0,1,\ldots M-1 \quad v=0,1,\ldots N-1 \tag{2}$$

where the ReF( u ,v) and ImF( u ,v) is given in Equation (3) and (4).

$$\mathrm{Re}\,F(u,v)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\left(\frac{2\pi ux}{M}\right) \tag{3}$$
$$\text{for } u=0,1,\ldots M-1 \quad v=0,1,\ldots N-1$$

$$\mathrm{Im}\,F(u,v)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\sin\left(\frac{2\pi vy}{N}\right) \tag{4}$$
$$\text{for } u=0,1,\ldots M-1 \quad v=0,1,\ldots N-1$$

Similarly inverse discrete Fourier transformation, where the frequency domain gets converted to the spatial domain, digital image may be written as in Equation (5).

$$f(x,y)=\frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cos\left(\frac{2\pi ux}{M}\right)+F(u,v)\,j\sin\left(\frac{2\pi vy}{N}\right)$$
$$\text{for } x=0,1,\ldots M-1 \quad y=0,1,\ldots N-1 \tag{5}$$

Now consider the real part of the transformed image and embed authenticating message or image bits at the LSB of each component (pixel) of the transformed image. After embedding, the embedded image is converted into spatial domain by using IDFT for transmitting over the network. The technique provides more security as embedding the message or image has been done by considering a window of the source image in sliding window manner and then transforming into frequency domain.

## 2. Earlier Works

N. Nameer and E. Eman in April 2007, implemented an algorithm based on hiding a large amount of data (image, audio, text) file into color BMP image. They used adaptive image filtering and adaptive image segmentation with bits replacement on the appropriate pixels. These pixels are selected randomly rather than sequentially by using new concept defined by main cases with their sub cases for each byte in one pixel [1].

P. K. Amin, Ni. Liu, and K. P. Subbalakshmi in 2005, described a discrete cosine transform (DCT) based spread spectrum data-hiding algorithm that provides statistical security [2].

R. Chandramouli and N. Memon in 2001, considered some specific image based steganography techniques and shown that an observer can indeed distinguish between images carrying a hidden message and images which do not carry a message [3].

S. Dumitrescu, X. L. Wu and Z. Wang in 2003 introduced an approach to detecting LSB steganography in digital signals. They shown that the length of hidden messages embedded in the LSB of signal samples can be estimated with relatively high precision. That approach was based on some statistical measures of sample pairs that are highly sensitive to LSB embedding operations [4].

B. Chen and G. W. Wornell in 2001 described the problem of embedding one signal (e.g., a digital watermark), within another "host" signal to form a third, "composite" signal [5].

P. Moulin and J. A. O'Sullivan in 2000 analyzed Information hiding as a communication game between an information hider and an attacker, in which side information is available to the information hider and to the decoder. They derived several Capacity formulas [6].

P. Moulin and M. K. Mıhçak in 2002 described an information-theoretic model for image watermarking and data hiding. Some recent theoretical results been used to characterize the fundamental capacity limits of image watermarking and data-hiding systems. Capacity was determined by the statistical model used for the host image, by the distortion constraints on the data hider and the attacker, and by the information available to the data hider, to the attacker, and to the decoder. They considered autoregressive, block-DCT and wavelet statistical models for images and compute data hiding capacity for compressed and uncompressed host-image sources [7].

C. Y. Lin and S. F. Chang in 1998 described a different goal from that of image watermarking which embeds into the image a signature surviving most manipulations. They described an effective technique for image authentication which can prevent malicious manipulations but allow JPEG lossy compression. The authentication signature was based on the invariance of the relationship between DCT coefficients of the same position in separate blocks of an image [8].

S. Pavan, G. Sridhar, and V. Sridhar in 2005 proposed a hybrid image registration algorithm to identify the spatial or intensity variations between two color images. The proposed approach extracts salient descriptors from the two images using a multivariate entropy-based detector. The transformation parameters are obtained after establishing the correspondence between the salient descriptors of the two images [9].

Copyright © 2009 SciRes. 

H. H. Pang, K. L. Tan, and X. Zhou, in 2004 introduced StegFD, a steganographic file driver that securely hides user-selected files in a file system so that, without the corresponding access keys, an attacker would not be able to deduce their existence. They proposed two schemes for implementing steganographic B-trees within a Steg FD volume [10].

## 3. Our Work

The presented work is based on information and image protection against unauthorized access in frequency domain. The ASMTDFT uses gray scale image of size (M x N) to be authenticated .The technique inserts authenticating message or image $X_{m,n}$ of size (M/2*N/ 2*3)-16 bits (maximum) as the first 16 bit holds the dimension of the file. DFT given in equation-1 is used to transform the image from spatial domain to frequency domain. The encoding and decoding scheme is given in Figure 1 and Figure 2 respectively.

### 3.1. Insertion Technique

Using the proposed scheme embedding is done completely in the frequency domain. DFT is applied on window of size 2 x 2 in sliding window manner to convert from spatial domain to frequency domain. Each pixel (8 bits) in spatial domain is transformed into two parts one is real part and another one is imaginary part. The authenticating bits are inserted at the LSB of the real part (excluding 1st pixel). The process is repeated for the whole image matrix in the same manner. After embedding inverse DFT is performed to convert from frequency domain to spatial domain. The algorithm for insertion is given in Subsection 3.1.1

#### 3.1.1. Insertion Algorithm

1) Take a message file or image whose size is less than or equal to (M/2*N/2*3)-16 bits where M x N is the size of the cover image.

2) Take 2 x 2 window of the cover image in sliding window manner and repeat Step 3 and 4 until the ends of the cover image.

3) Apply the Discrete Fourier Transformation.

4) Consider the real part of the frequency component and do the following.

- Take three frequency component values but not the first one and do the following.
  - Consider the Least Significant Bit position of the DFT component.
    - Replace the bit by one authenticating bit.

5) Apply the Inverse Discrete Fourier Transformation.

6) Stop.

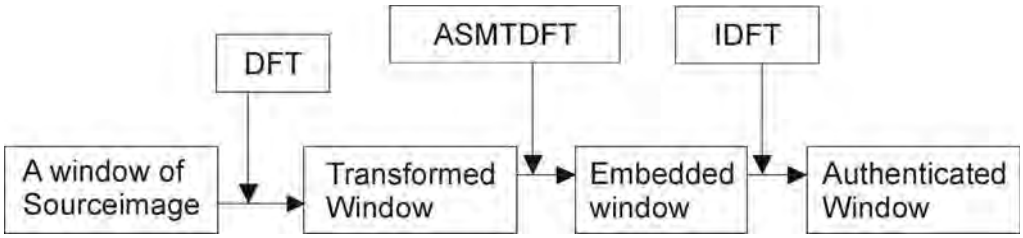



**Figure 1. Encoding scheme using ASMTDFT.**

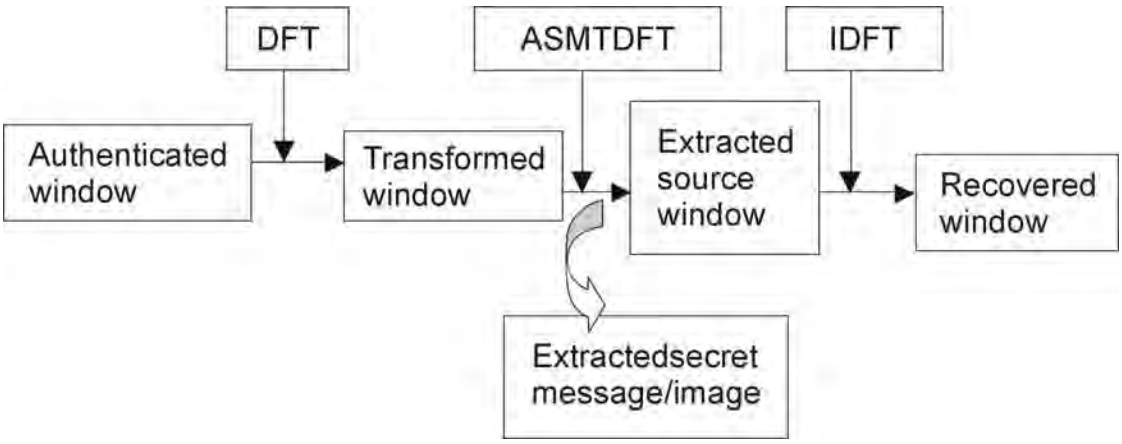



**Figure 2. Decoding scheme using ASMTDFT.**

Copyright © 2009 SciRes. 

### 3.2. Extraction Technique

During decoding the embedded image has been taken as input in spatial domain. To convert form spatial domain to frequency domain DFT is applied using the same window of size 2 x 2. Apply the extraction algorithm to extract the authenticating message or image from the transformed image. The process is repeated for the whole embedded image matrix in the same manner. Inverse DFT is performed to transform from frequency domain to spatial domain to generate original source image. The algorithm for extraction is given in Section 3.2.1

#### 3.2.1. Extraction Algorithm

1) Take the authenticated image as input.
2) Consider 2 x 2 mask of the input image at a time and repeat Step 3 and 4 until the ends of the embedded image.
3) Apply the Discrete Fourier Transformation.
4) Consider the real part of the frequency component and do the following.

- Take three frequency component values but not the first one and do the following.
  - Extract the Least Significant Bit.
  - Replace this bit position by '1' or by '0'.

5) Apply the Inverse Discrete Fourier Transformation.
6) Stop.

## 4. Results and Comparisons

In this section results are analyzed and comparative studies have been made between proposed technique and S-Tools in terms of test for homogeneity i.e. Chi-square test and histogram analysis. Subsection 4.1 illustrates Chi-Square test. Subsection 4.2 deals with histogram analysis.

Figure 3(a) shows source image 'Hill' and Figure 3(b) shows the authenticating image 'Lotus' and Figure 3(c) and Figure 3(d) are embedded image using proposed algorithm and S-Tools respectively. The authenticating image 'Lotus' has been embedded into the source image 'Hill'. Some differences may be observed between source image and embedded image by S-Tools [11] but no such differences are observed in source image and embedded image by proposed technique.

Figure 4(a) to Figure 4(d) indicate the comparison of visual changes for another source image 'Rasmancha' (Figure 4(a)) embedding with the same Lotus image (Figure 4(b)). The results are embedded image using proposed algorithm (Figure 4(c)) and S-Tools (Figure 4(d)). Here, may be observed some variations between source image and embedded image by S-Tools but no such differences are observed in source image and embedded image by ASMTDFT.

**(a). Hill.** **(b). Lotus.** **(c). ASMTDFT.** **(d). S-tools.**

**Figure 3. Comparison of visual fidelity in embedding 'Lotus' using ASMTDFT and S-Tools.**

**(a). Rashmancha.** **(b). Lotus.** **(c). ASMTDFT.** **(d). S-tools.**

**Figure 4. Comparison of visual fidelity in embedding 'Lotus' using ASMTDFT and S-Tools.**

Copyright © 2009 SciRes. 

**Table 1. Comparison of Chi-Square values in ASMTDFT.**

| Images | File Size | Uncertainty | Degree of freedom | Calculated Chi-Square | Tabulated Chi-Square |
|---|---|---|---|---|---|
| Source and authenticated Hill image by Lotus | 98 x 130 | 0.01 | 255 | 264.219 | 310.457 |
| Authenticated and Extracted Hill Image | 98 x 130 | 0.001 | 255 | 315.089 | 347.650 |
| Source and Authenticated Rashmancha Image | 98 x 130 | 0.01 | 255 | 241.284 | 310.457 |
| Authenticating Image & Extracted Image | 33 x 26 | 0.01 | 255 | 0.00 | 310.457 |

**(a). Authenticated hill.** **(b). Extracted lotus.** **(c). Extracted hill.**

**Figure 5. Comparison of visual fidelity in extracting 'Lotus' and source image 'Hill' using ASMTDFT.**

Using ASMTDFT we are also able to separate the Source image and the authenticating image from the authenticated image. Figure 5(a) to Figure 5(c) show this result. Figure 5(a) is the Authenticated Hill image. Now using extraction procedure of ASMTDFT Figure 5(b) is the extracted Lotus image and Figure 5(c) is the Extracted Hill (Source) image. This extraction is not possible by s-tools.

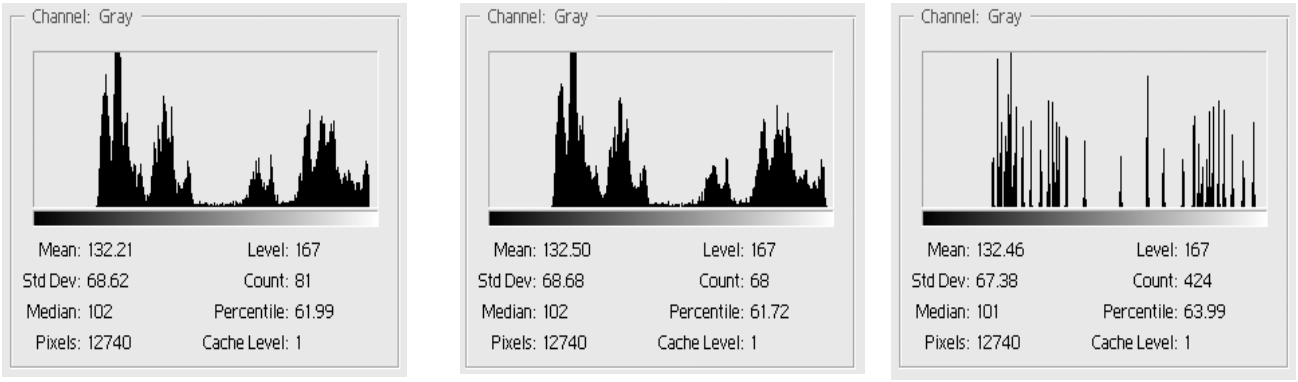
## 4.1. Chi-Square Test

The Chi-Square test has been performed for the source image and authenticated image, and also for the Authenticating Image & Extracted Image. The values of chi -squares are given in Table 1 for different images, which show that the calculated chi-square value is less than the tabulated chi-square value for some level of significance, which indicates the homogeneity of the images. They are more significant for 1% level of uncertainty. For the authenticating and extracted image the Chi-Square value is zero. That is we are able to extract the original image without any noise.

## 4.2. Histogram Analysis

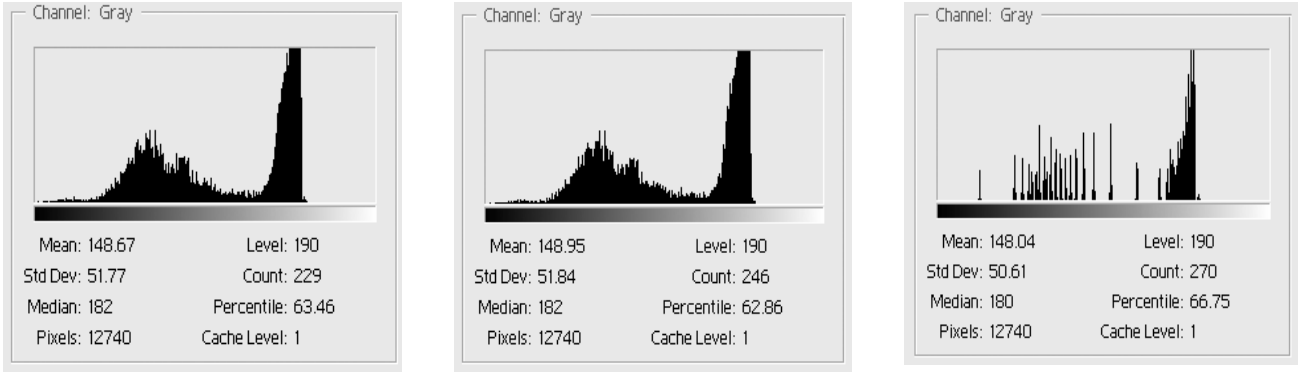
Histogram analyses have been performed between source image 'Hill' and for the embedded image using 'Lotus' by applying proposed technique and S-Tools and also for the 'Rashmancha' image. In both the cases noticeable differences are observed in frequency distribution table of pixel values in source image and embedded image using S-Tools algorithm. But very small variances are observed in frequency distribution table of pixel values in source image and embedded image using proposed technique. Figure 6 shows the visual effect of histograms in embedding source image 'Hill' with proposed technique and S-Tools. The histogram of the source image 'Hill', the histogram of the embedded image by 'Lotus' image using proposed technique and the histogram of the image embedded using 'Lotus' image by applying S-Tools are shown in Figure 6. It is seen clearly that in the proposed technique the histogram remains almost identical with the source image even after embedding the image with 'Lotus' image where as in case of embedding

Copyright © 2009 SciRes. 



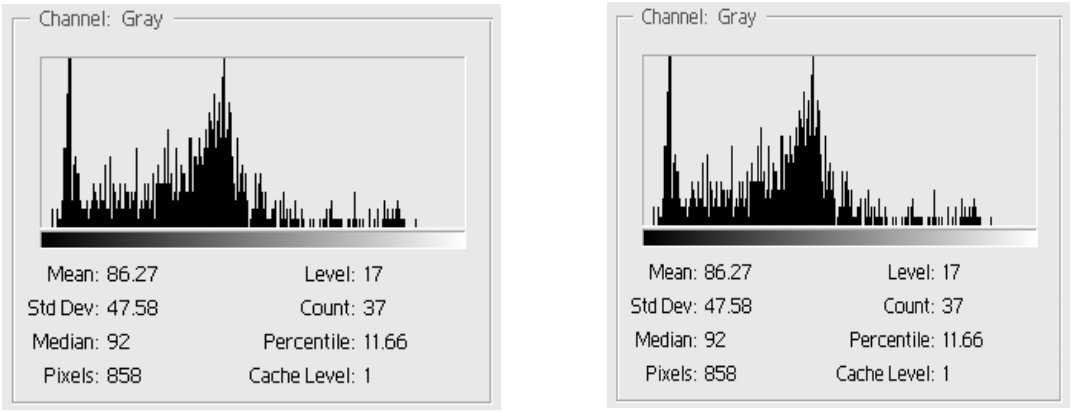

**(a). Hill.** **(b). ASMTDFT.** **(c). S-Tools.**

**Figure 6. Histogram for source image 'Hill', embedded image using.**




**(a). Rashmancha.** **(b). ASMTDFT.** **(c). S-Tools.**

**Figure 7. Histogram for source image 'Rashmancha', embedded image using ASMTDFT and S-Tools.**




**(a). Lotus.** **(b). Extracted Lotus.**

**Figure 8. Histogram for authenticating image 'Lotus', extracted image 'Lotus' using ASMTDFT.**

Copyright © 2009 SciRes. 

**Table 2. Histogram analysis.**

| Images | Level | Count | Mean | Std. Dev |
|---|---|---|---|---|
| Source Hill image | 167 | 81 | 132.21 | 68.62 |
| Authenticated Hill image by Lotus using ASMTDFT | 167 | 68 | 132.50 | 68.68 |
| Authenticated Hill image by Lotus using S-Tools | 167 | 424 | 132.46 | 67.38 |
| Source Rashmancha image | 190 | 229 | 148.67 | 51.77 |
| Authenticated Rashmancha image by Lotus using ASMTDFT | 190 | 246 | 148.95 | 51.84 |
| Authenticated Rashmancha image by Lotus using S-Tools | 190 | 270 | 148.04 | 50.61 |
| Lotus Image | 17 | 37 | 86.27 | 47.58 |
| Extracted Lotus Image | 17 | 37 | 86.27 | 47.58 |

with Stools there is a noticeable change in histogram in compare to the histogram of source image 'Hill'. From these observations it may be inferred that the proposed technique may obtain better performance in embedding. Histogram analyses have also been done for another source image 'Rashmancha', which is depicted in Figure 7(a)−7(c). Figures 8(a), 8(b) show the histogram of the embedding image 'Lotus' and the histogram of the extracted image 'Lotus'. From the histograms we see that there are no differences. So we can conclude that the proposed algorithm gives a very good result for extraction. Table 2 gives a clear idea of the histograms of different images in a tabular form. Here we have considered a particular gray level value for images and check the variances in terms of total no. pixels, Mean, Standard deviation. The comparisons have been done between the source image and authenticated image both for ASMTDFT & S-Tools, and also for the authenticating image and the extracted image which we have got using ASMTDFT.

## 5. Conclusions

In this paper the proposed technique implemented here for image authentication and secret message transmission. The algorithm used here is the bit level message or image insertion and extraction in the frequency domain. Using ASMTDFT we are also able to extract the source image. In this technique 2 x 2 window is selected for better result of authentication. Insertion and extraction is done in frequency domain instead of spatial domain for more security. From the results of Chi-Square test and histogram analysis and comparison with S-Tools the proposed technique may obtain better result.

## 6. Acknowledgements

This work was supported by the Security Engineering Research Center, granted by the Korea Ministry of Knowledge Economy. And this work has successfully completed by the active support of Prof. Tai-hoon Kim, Hannam University, Republic of Korea and Prof. Samir Kumar Bandyopadhyay, University of Calcutta, India.

## 7. References

[1] N. Nameer and E. Eman, "Hiding a large amount of data with high security using steganography algorithm," Journal of Computer Sciences, pp. 223−232, April 2007.

[2] P. K. Amin, N. Liu, and K. P. Subbalakshmi, "Statistically secure digital image data hiding," Multimedia Signal Processing, IEEE 7th Workshop, Shanghai, pp. 1−4, October 2005.

[3] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," International Conference on Image Processing, Thessaloniki, Greece, pp. 1019−1022, 2001.

[4] S. Dumitrescu, X. L. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," IEEE Transactions on Signal Processing, Vol. 51, No. 7, pp. 1995−2007, July 2003.

Copyright © 2009 SciRes. 

[5] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," IEEE Transaction on Information Theory, Vol. 47, pp. 1423–1443, 2001.

[6] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," IEEE International Symposium on Information Theory, Sorrento, Italy, pp. 19, June 2000.

[7] P. Moulin and M. K. Mıhcak, "A framework for evaluating the data-hiding capacity of image sources," IEEE Transaction on Image Processing, Vol. 11, pp. 1029–1042, September 2002.

[8] C. Y. Lin and S. F. Chang, "A robust image authentication method surviving JPEG lossy compression," SPIE, pp. 296–307, 1998.

[9] S. Pavan, G. Sridhar, and V. Sridhar, "Multivariate entropy detector based hybrid image registration," IEEE ICASSP, Vol. 2, pp. 873–876, March 18–23, 2005.

[10] H. H. Pang, K. L. Tan, and X. Zhou, "Steganographic schemes for file system and B-tree," IEEE Transaction on Knowledge and Data Engineering, Vol. 16, No. 6, pp. 701–713, June 2004.

[11] http://www.spychecker.com/download/download_stools.html, visited as on March 28, 2009

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 371-377
doi:10.4236/ijcns.2009.25041 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Regulation of Queue Length in Router Based on an Optimal Scheme

**Nannan ZHANG**
*Key Laboratory of Integrated Automation of Process Industry, Northeastern University, Shenyang, China*
*Email*: *nannanathome@163.com*
*Received July* 12, 2008; *revised January* 20, 2009; *accepted March* 5, 2009

## ABSTRACT

Based on the proportionally fair scheme that Kelly proposed to solve the optimization problems for utility function in networks, and in order to improve the congestion control performance for the queue in router, the linear and terminal sliding active queue management (AQM) algorithms are designed. Especially in the terminal sliding AQM algorithm, a special nonlinear terminal sliding surface is designed in order to force queue length to reach the desired value in finite time. The upper bound of the time is also obtained. Simulation results demonstrate that the proposed congestion algorithm enables the system be better transient and stable performance. At the same time, the robustness is guaranteed.

**Keywords:** Congestion Control, Sliding Mode Control (SMC), Active Queue Management (AQM), Kelly's Proportional Fair Scheme

## 1. Introduction

Active Queue Management (AQM), as a class of packet dropping/marking mechanism in the router queue, has been recently proposed in order to convey congestion notification early enough to the senders, so that the senders are able to reduce the transmission rates before the queue overflows and any sustained packet loss occurs [1]. There are three typical kinds of AQM algorithms: One is heuristic algorithms, such as RED (Random Early Detection) [2], BLUE [3]; One is the utility function optimal model based on economics, like REM (Random Exponential Marking) [4], AVQ (Adaptive Virtual Queue) [5]; The other one is based on the sourcing and queuing dynamic model, as PI [6] and VRC (Virtual Rate Control) [7]. The advantage of the latter two algorithms is that the design of controller is based on explicit model, so the stability analysis and parameters modulation can be given theoretically.

TCP/AQM dynamics have time varying roundtrip times (RTTs) and uncertainties, which requires more robustness for the designed schemes. This paper focus on the source algorithm which Kelly [8] proposed based on economic utility function through an optimization framework. It is well known that sliding mode control (SMC) is an effective method of robustness control, and sliding mode control systems possess strong robustness against parameter perturbations and external disturbances [9], which is very suitable for time varying network system. In the recent years, many studies have been focus on the domain [10−12]. In these papers, they proposed robust sliding mode control algorithms. Because the three controllers are insensitive to variance of the network parameters, they are suitable for time-varying network systems. However, one of the representative characteristics of the three controllers is that the convergence of the system states to the equilibrium point is asymptotical but not in finite time, which means the queue length in buffer, can not converge to the desired value in finite time. In addition, there still exists chattering when system motion is in steady state. Recently, the terminal sliding mode control (TSMC) has been developed [13, 14], which can guarantee the finite reaching time to the sliding surface from initial states and the finite reaching time to the equilibrium point.

Kelly *et al.* [8] have studied the convergence in the presence of communication delays [4,15] through an optimization framework, which is a new and effective

Copyright © 2009 SciRes. 

method to analyze the performance of congestion control for the Internet. So in this paper we propose an AQM algorithm based on Kelly's scheme by using sliding mode control. The structure of the paper are as follows: in Section 2 we analyze the Kelly's method, in Section 3 and 4, we design the linear and terminal sliding mode AQM controller respectively to study the convergence of the queue, the conclusion is given in the last section.

## 2. Kelly's Optimal Scheme

In this section we briefly describe the rate allocation problem in the Kelly's optimization framework.

Consider a network with a set $L$ of resources and a set $I$ of users. Let $C_l$ denote the finite capacity of resource $l \in L$. Each user $i \in I$ has a fixed route $r_i$, which is a set of resources traversed by user $i$'s packets. We define a zero-one matrix $A$, where $A_{i,l} = 1$ if $l \in r_i$ and $A_{i,l} = 0$ otherwise. When its rate is $x_i$, user $i$ receives utility $U_i(x_i)$. We take the view that the utility functions of the users are used to select the desired rate allocation among the users. The utility $U_i(x_i)$ is an increasing, strictly concave and continuously differentiable function of $x_i$ over the range $x_i \geq 0$. Under this setting, the rate allocation problem of interest can be formulated as the following optimization problem [15]:

$$SYSTEM(U, A, C) \quad \max_{x_i \geq 0} \sum_{i \in I} U_i(x_i) \quad \text{s.t. } A^T x \leq C, \tag{1}$$

where $C = (C_l, l \in L)$. The first constraint is the capacity constraint which states that the sum of the rates of all users utilizing resource should not exceed its capacity $C_l$.

Each user $i$ adjusts its rate according to the following differential equation.

$$\frac{d}{dt} x_i(t) = k_i \left( \omega_i - x_i(t) \sum_{l \in r_i} p_l \left( \sum_{i \in I} x_i(t) \right) \right), \tag{2}$$

where $k_i$ and $\omega_i$ are positive constants, $k_i$ is the gain parameter $\omega_i$ shows the users' willingness to pay per unit time. $p_l(\cdot)$ is an increasing function of the aggregate rate of the users going through it, and it can also be seen as the packet loss function that similar to ECN possibility function [16].

The simplified dynamic model is

$$\dot{r}(t) = k(\omega - r(t)p(t)), \tag{3}$$

where $r(t)$ is the user's sending rate at time $t$, $p(t)$ is the marker probability of ECN, $k$ and $\omega$ are the corresponding parameters.

Assume the network model is single user and single link. The dynamic buffer length at bottleneck is that

$$\dot{q}(t) = r(t) - C, \tag{4}$$

where $q(t)$ is the instantaneous queue length in buffer, $C$ is link capacity.

Let $x_1(t) = q(t) - q_d$, $x_2(t) = r(t) - C$, (3) and (4) can be described as

$$\dot{x}_1(t) = x_2(t), \tag{5}$$

$$\dot{x}_2(t) = k\left[\omega - (x_2(t) + C)p(t)\right], \tag{6}$$

where $q_d$ is the reference queue length.

The AQM algorithms based on control theory treat the queue length in router as a controllable state, and through a feedback mechanism adjust the probability of traffic drop or marker. In order to maintain the queue length at the desired value, then the system can get high link utilization and low time delay.

Our control objective is through design of the marker probability $p(t)$ with $0 \leq p(t) \leq 1$, obtain higher link utilization, low packet loss rate and small queue fluctuations.

## 3. Design of Sliding Mode Control Algorithm

There are mainly two steps in the design of sliding mode control systems. One is the design of sliding surface, on which the system should get desired quality, such as asymptotically stable; The other is designing the sliding mode controller, which should guarantee the arriving condition.

### 3.1. Sliding Surface Design

First choose a sliding surface as conventional

$$S(t) = cx_1(t) + x_2(t). \tag{7}$$

The objective of sliding mode control is to make the state slide to origin along the sliding surface in a finite time. That means the error of queue length is zero, and the sending rate and link capacity are totally matching.

When arrive at the sliding surface, $S(t) = 0$, so

$$cx_1(t) + x_2(t) = 0. \tag{8}$$

Substituting (8) into (5), we can obtain the sliding mode dynamics as follows

Copyright © 2009 SciRes. 

$$\dot{x}_1(t) = -cx_1(t), \quad \textbf{(9)}$$

$$x_1(t) = x_1(t_0)e^{-c(t-t_0)}, \quad \textbf{(10)}$$

where $t_0$ means the initial time. So the system motion on the sliding surface (7) can converge to the origin point in finite time if $c > 0$.

### 3.2. Sliding Mode Controller Design

Let $S(t) = 0$, we can get the equivalent control law

$$p_{eq}(t) = \frac{cx_2 + k\omega}{k(x_2 + C)} = \frac{c(r(t) - C)}{kr(t)} + \frac{\omega}{r(t)} \quad . \quad \textbf{(11)}$$

Apparently, this controller can make the system (5) (6) stable, but it can not satisfy the physical meaning of the marker probability $0 \le p_{eq} \le 1$. However (11) is helpful for us to design a more reasonable AQM controller

$$p(t) = \left( \frac{\alpha c}{k} \left| \frac{r(t) - C}{r(t)} \right| + \beta \right) sign(S(t)) + \frac{\omega}{r(t)}. \quad \textbf{(12)}$$

*Theorem 1:* If the control law (12) is used for system (5) and (6), the reaching condition is satisfied if $\alpha \ge 1, \beta > 0$.

*Proof:* When $S(t) > 0$,

$$\begin{aligned} \dot{S}(t) &= c(r(t) - C) + k\left[ \omega - r(t)\left( \frac{\omega}{r(t)} + \frac{\alpha c}{k} \left| \frac{r(t) - C}{r(t)} \right| + \beta \right) \right] \\ &= c(r(t) - C) - \alpha c \left| r(t) - C \right| - kr(t)\beta \\ &\le (1 - \alpha)c \left| r(t) - C \right| - kr(t)\beta \\ &< (1 - \alpha)cr(t) - kr(t)\beta \\ &= r(t)[(1 - \alpha)c - k\beta]. \end{aligned}$$

So if $S(t) > 0$, choose $\alpha \ge 1, \beta > 0$ and the reaching condition $S(t)\dot{S}(t) < 0$ is satisfied.

When $S(t) < 0$,

$$\begin{aligned} \dot{S}(t) &= c(r(t) - C) + k\left[ \omega - r(t)\left( \frac{\omega}{r(t)} - \frac{\alpha c}{k} \left| \frac{r(t) - C}{r(t)} \right| + \beta \right) \right] \\ &= c(r(t) - C) + \alpha c \left| r(t) - C \right| + kr(t)\beta \\ &\ge (\alpha - 1)c \left| r(t) - C \right| + kr(t)\beta \\ &> (\alpha - 1)cr(t) + kr(t)\beta \\ &= r(t)[(\alpha - 1)c + k\beta]. \end{aligned}$$

So if $S(t) < 0$, choose $\alpha \ge 1, \beta > 0$ and the reaching condition $S(t)\dot{S}(t) < 0$ is satisfied too.

*Theorem 2:* The sliding mode dynamics (9) can converge to origin point after the time

$$t_{\max} = \frac{\left| cx_1(0) + x_2(0) \right|}{k\beta r_{\min}}, \quad \textbf{(13)}$$

where $r_{\min}$ is the lowest sending rate.

*Proof:* Choose a Lyapunov function candidate for (9) as follows

$$V(x,t) = \frac{1}{2}S^2(t), \quad \textbf{(14)}$$

then the time derivative of $V(t)$ along (9) is

$$\begin{aligned} \dot{V} &= S\dot{S} \\ &= S\left( cx_2 + k(\omega - (x_2 + C)p(t)) \right) \\ &= Scx_2 - \left| S \right| (\alpha c \left| x_2 \right| + k\beta r(t)) \\ &\le -\left| S \right| (\alpha c \left| x_2 + k\beta r(t) \right| + c \left| x_2 \right| \left| S \right|) \\ &\le -\left| S \right| ((\alpha - 1)c \left| x_2 \right| + k\beta r(t)). \end{aligned} \quad \textbf{(15)}$$

From Theorem 1 we can get $\alpha \ge 1, \beta > 0$, so

$$\dot{V} < -\left| S \right| k\beta r(t) < -\left| S \right| k\beta r_{\min}. \quad \textbf{(16)}$$

By (14), we have

$$\left| S \right| = \sqrt{2V}. \quad \textbf{(17)}$$

Substituting (17) into (16), we can obtain the following inequality

$$\dot{V} < -\sqrt{2V}(k\beta r_{\min}), \quad \textbf{(18)}$$

then

$$V(x,t) < \left( -\frac{k\beta r_{\min}}{\sqrt{2}} + \sqrt{V(0)} \right)^2. \quad \textbf{(19)}$$

So we can get

$$t_{\max} < \frac{\sqrt{V(0)}}{k\beta r_{\min}} = \frac{\left| cx_1(0) + x_2(0) \right|}{k\beta r_{\min}}. \quad \textbf{(20)}$$

### 3.3. Simulation Results

In this section we validate the effectiveness and performance of the controller proposed in this paper by simulations. We consider the dumbbell network topology with a single bottleneck link in Figure 1.

Choose the parameters of network as follows: the maximum buffer of each router is 500 packets and the link capacity is $C = 1250$packets/s. The desired queue length $q_d$ is 200 packets. The initial queue length is 400 packets. The PSMC-AQM controller parameters are $\alpha = 1,\ \omega = 10,\ \beta = 0.05,\ k = 15,\ c = 12$. In order to

Copyright © 2009 SciRes. 

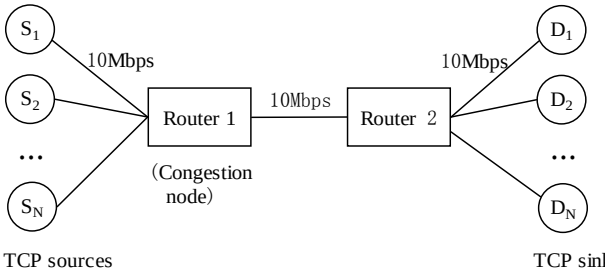



**Figure 1. Simulation network topology.**

reduce the chattering problem, a saturation function is used. The RED (Random Early Detection) algorithm is also simulated under the same network condition for the purpose of comparison. In addition, we use the parameters minimum 80packets and maximum 320packets.

Queue length, link utilization and data dropping between RED-AQM and PSMC-AQM are compared. Figure 2 and Figure 3 demonstrate that the PSMC-AQM controller considers not only the queue length but also the matching condition of the assemble rate and the link capacity, so it reflects the network condition better than RED. Both of the algorithms can control the queue length near the desired 200 packets, and PSMC-AQM get less data loss rate, so it makes higher link utilization rate. The results are showed in Table 1.

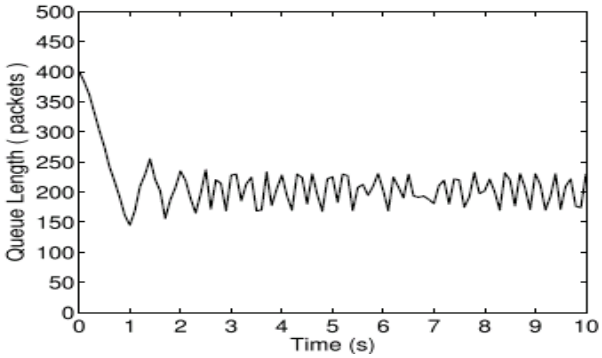



**Figure 2. Average queue length using RED.**

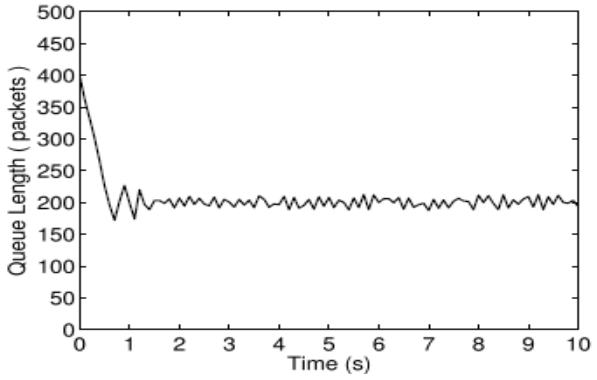



**Figure 3. Average queue length using PSMC.**

**Table 1. Comparison between RED and PSMC.**

| performance | RED | PSMC |
|---|---|---|
| rate of link utilization | 99.67 | 99.86 |
| rate of data loss | 2.35 | 0.09 |

## 4. Design of Terminal Sliding Mode Control Algorithm

Last section introduces sliding mode control into the optimization based internet congestion control model, and designs a linear sliding surface, and then validates the effectiveness of the algorithm in simulation. But the sliding mode along the sliding surface is asymptotically stable, that means the converging time could be quite long. For speediness is so important for a router algorithm, a special nonlinear sliding surface named terminal sliding surface is proposed in this section. The terminal sliding surface guarantees the finite reaching time to the sliding surface from initial states and the finite reaching time to the origin point. So the converging time is limited, and the speed of the sliding mode control system is enhanced, further the congestion control performance is improved.

### 4.1. Design of Terminal Sliding Surface

We design a nonlinear terminal sliding surface as follows:

$$S(t) = d_1 x_1(t) + d_2 x_2(t) + d_3 (x_1(t))^{q/p}, \quad \textbf{(21)}$$

where $d_1 > 0$, $d_2 > 0$, $d_3 > 0$, $p$ and $q$ are odd positive integers and they satisfy $q<p<2q$. The sliding surface $S(t)$ corresponds to a combination of the queue length error, the error between incoming traffic rate and link capacity.

When the system state trajectories are on the terminal sliding surface, $S(t)$ satisfies $S(t) = 0$. So we can obtain the following equality

$$x_2(t) = -d_2^{-1} [\, d_1 x_1(t) + d_3 (x_1(t))^{q/p} ]. \quad \textbf{(22)}$$

Substituting (22) into (5), we can obtain the sliding mode dynamics as follows:

$$\dot{x}_1(t) = -d_2^{-1} d_1 x_1(t) - d_2^{-1} d_3 (x_1(t))^{q/p}. \quad \textbf{(23)}$$

In order to prove that (23) can converge to the equilibrium point in finite time, we introduce a lemma as follows

*Lemma 1* [13]: Assume that a continuous, positive definite function $V(t)$ satisfies the following differential inequality

$$\dot{V}(t) \le -\alpha V^{\eta}(t), \quad \forall t \ge 0, \quad V(0) \ge 0, \quad \textbf{(24)}$$

Copyright © 2009 SciRes. 

where $\alpha > 0$, $0 < \eta < 1$ are constants. Then $V(t)$ satisfies the following inequality:

$$V^{1-\eta}(t) \le V^{1-\eta}(0) - \alpha(1-\eta)t, \quad 0 \le t \le t_r \tag{25}$$

and

$$V(t) = 0, \quad \forall t \ge t_r \tag{26}$$

with $t_r$ given by

$$t_r = \frac{V^{1-\eta}(0)}{\alpha(1-\eta)}. \tag{27}$$

*Theorem 3*: The sliding mode dynamics (23) can converge to the equilibrium point after the time $t_r$ and $t_r$ satisfies

$$t_r = \frac{\|x_1(0)\|^{2(1-\eta)}}{2^{1-\eta}\alpha(1-\eta)}, \tag{28}$$

where $x_1(0)$ is the initial value of $x_1(t)$ and $\alpha = 2^{\eta} d_2^{-1} d_3$, $\eta = \frac{q/p+1}{2}$.

*Proof*: Choose a Lyapunov function candidate for the system (23) as follows:

$$V(t) = \frac{1}{2} x_1^{\mathrm{T}}(t) x_1(t) \tag{29}$$

then the time derivative of $V(t)$ along (23) is

$$\begin{aligned} \dot{V}(t) &= x_1^{\mathrm{T}}(t)\dot{x}_1(t) \\ &= x_1^{\mathrm{T}}(t)[-d_2^{-1}d_1 x_1(t) - d_2^{-1}d_3 (x_1(t))^{q/p}] \\ &= -d_2^{-1}d_1 \|x_1(t)\|^2 - d_2^{-1}d_3 \|x_1(t)\|^{q/p+1} \\ &\le -d_2^{-1}d_3 \|x_1(t)\|^{q/p+1}. \end{aligned} \tag{30}$$

By (29), we have

$$\|x_1(t)\| = \sqrt{2V(t)} \tag{31}$$

Substituting (31) into (30), we can obtain the following inequality

$$\dot{V}(t) \le -\alpha V^{\eta}(t) \tag{32}$$

where $\alpha = 2^{\eta} d_2^{-1} d_3$, $\eta = \frac{q/p+1}{2}$.

According to Lemma 1, we know that the sliding mode dynamics (23) can converge to the equilibrium point after the time $t_r$ and $t_r$ satisfies the following equality

$$t_r = \frac{V^{1-\eta}(0)}{\alpha(1-\eta)} = \frac{\|x_1(0)\|^{2(1-\eta)}}{2^{1-\eta}\alpha(1-\eta)} \tag{33}$$

So the system motion on the terminal sliding surface (21) can converge to the equilibrium point in finite time.

### 4.2. Design of Terminal Sliding Mode Controller

In the subsection, we design a robust terminal sliding mode controller to satisfy the reaching condition. We consider the following control structure of the form

$$p(t) = p_{eq}(t) + p_N(t). \tag{34}$$

*Theorem 4*: If the control law is used for system (5) and (6) as follows:

$$\begin{aligned} p_{eq}(t) &= \frac{d_1 x_2 + (q/p) d_3 x_1^{q/p-1} x_2 + d_2 k\omega}{d_2 k(x_2 + C)} \\ &= \frac{d_1(r(t) - C)}{d_2 k r(t)} + \frac{q d_3 x_1^{q/p-1}}{p d_2 k r(t)} + \frac{\omega}{r(t)}, \end{aligned} \tag{35}$$

$$p_N(t) = \frac{k_v S(t) + \varepsilon S^2 \operatorname{sgn}(S(t))}{d_2 k(x_2 + C)}. \tag{36}$$

then the controller can satisfy the reaching condition

$$\dot{S}(t) = -\varepsilon S^2 \operatorname{sgn}(S(t)) - kS(t). \tag{37}$$

*Proof* : Recall (21), the time derivative of $S(t)$ along the trajectory of (5) and (6) under the control (34) is given as

$$\begin{aligned} \dot{S}(t) &= d_1 \dot{x}_1(t) + d_2 \dot{x}_2(t) + d_3 (q/p)(x_1(t))^{q/p-1} \dot{x}_1(t) \\ &= d_1 x_2(t) + d_2 (k\omega - (x_2(t) + C)) p(t)) \\ &\quad + d_3 (q/p)(x_1(t))^{q/p-1} x_2(t). \end{aligned} \tag{38}$$

Substitute (34) into (38), we can get

$$\dot{S}(t) = -k_v S(t) - \varepsilon S^2 \operatorname{sgn}(S(t)).$$

So the controller (34) can satisfy the reaching condition (37). That is to say, the controller can force system state trajectories toward the terminal sliding surface infinite time and maintain them on the sliding surface after then. Meanwhile, the reaching rate is fast and chattering is low.

### 4.3. Simulation Results

In this section the effectiveness and performance of the terminal sliding mode controller (TSMC) is validated by simulations. Common sliding mode controller (SMVS, Sliding Mode Variable Structure) is also simulated for the purpose of comparison with suggested parameter values $\alpha = 0.96$, $\beta = -0.96$, $w = 2$ given in [15]. We consider the same dumbbell network topology with a single bottleneck link as Figure 1.

Copyright © 2009 SciRes. 

The network parameters are chosen the same as Part C of Section 3: the maximum buffer of each router is 500 packets and the link capacity is $C = 1250$packets/s . The desired queue length $q_d$ is 200 packets. The initial queue length is 400 packets. And the parameters of TSMC-AQM controller are chosen as $d_1 = 1$, $d_2 = 1$, $d_3 = 1000$, $q = 3$, $p = 5$, $k = 5$, $\varepsilon = 0.1$. From Theorem 3 we can calculate $t_r = 0.5506$ s.

The simulation is operated under variable network conditions. The link capacity *C* and the roundtrip time *R* are varied through the process. Figure 4 and Figure 5 show that the two controllers are insensitive to different TCP loads and link capacity, but TSMC has shorter regulating time and better steady performance than SMVS controller.

## 5. Conclusions

In this paper, we combine the Kelly's optimization scheme and the sliding mode control algorithm to analyze the convergence of the queue. We design two sliding mode algorithm: the linear and the terminal ones. The simulation results show that both of the algorithms can converge to the equilibrium point in finite time. Obviously, the terminal sliding mode control can obtain faster transients and less oscillatory responses under dynamic network conditions, which translates into higher link utilization, low packet loss rate and small queue fluctuations. And the proposed controller has better stability and robustness than common sliding mode controller, which would be meaningful for the congestion control of the Internet.

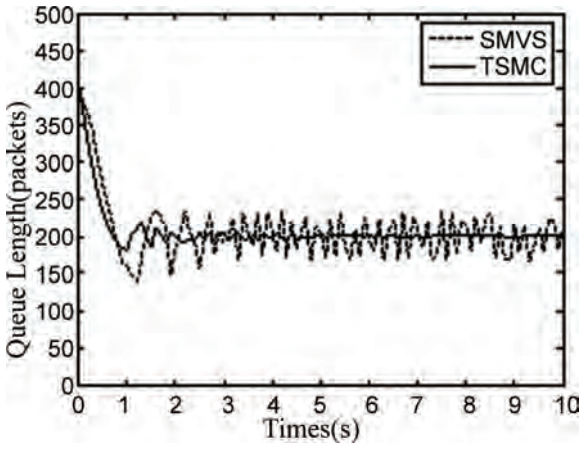



**Figure 4. The comparison with variable *C*.**

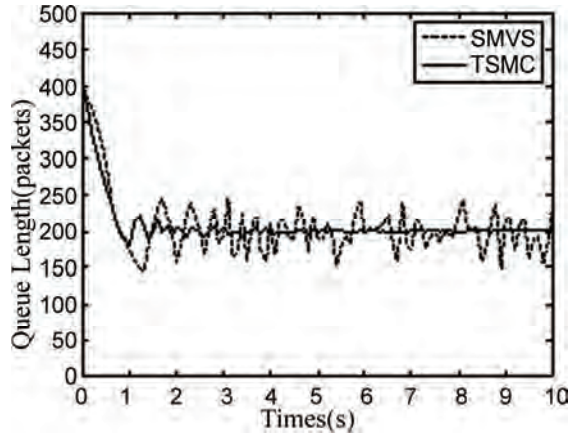



**Figure 5. The comparison with variable *R* .**

## 6. References

[1] B. Braden and D. Clark, "Recommendations on queue management and congestion avoidance in the Internet," RFC 2309, 1998.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transaction on Networking, Vol. 1, pp. 397–413, 1993.

[3] W. C. Feng, Kang G. Shin, D. D. Kandlur, *et al*., "The blue active queue management algorithms," IEEE/ACM Transactions on Networking, Vol. 10, No. 4, pp. 513–528, 2002.

[4] S. Athuraliya, S. H. Low, V. H. Li, *et al*., "REM: Active queue management," IEEE Network, Vol. 15, No. 3, pp. 48–53, 2001.

[5] S. Srisankar, Kunniyur, and R. Srikant, "An adaptive virtual queue (AVQ) algorithm for active queue management," IEEE/ACM Transactions on Networking, Vol. 12, No. 2, pp. 266–289, 2004.

[6] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," Proceedings of IEEE INFOCOM, Anchorage, Alaska, USA, IEEE Communications Society, pp. 1726–1734, 2001.

[7] H. Lim, K. J. Park, and C. H. Choi, "Virtual rate control algorithm for active queue management in TCP networks," IEEE Electronics Letters, Vol. 38, No. 16, pp. 873–874, 2002.

[8] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," Journal of the Operational Research Society, Vol. 49, No. 3, pp. 237–252, March 1998.

[9] Y. H. Roh and J. H. Oh, "Robust stabilization of uncertain input-delay systems by sliding mode control with delay compensation," Automatica, Vol. 35, pp. 1861–1865, 1999.

[10] F. Y. Ren, C. Lin, and X. H. Yin, "Design a congestion controller based on sliding mode variable structure control," Computer Communications, Vol. 28, pp. 1050–1061, 2005.

Copyright © 2009 SciRes. 

[11] P. Yan, Y. Gao, and H. AOzbay, "A variable structure control approach to active queue management for TCP with ECN," IEEE Transactions on Control Systems Technology, Vol. 13, pp. 203−215, 2005.

[12] F. J. Yin, G. M. Dimirovski, and Y. W. Jing, "Robust stabilization of uncertain input delay for Internet congestion control," Proceedings of the American Control Conference, Minneapolis, Minnesota, USA, pp. 5576−5580, 2006.

[13] Y. Tang, "Terminal sliding mode control for rigid robots," Automatica, Vol. 34, pp. 51−56, 1997.

[14] S. H. Yu, X. H. Yu, B. Shirinzadeh, and Z. H. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," Automatica, Vol. 41, pp. 1957−1964, 2005.

[15] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low, "Congestion control for high performance, stability, and fairness in general networks," IEEE/ACM Transaction on Networking, Vol. 13, No. 1, pp. 43–56, 2005.

[16] R. Thommes and M. J. Coates, "Deterministic packet marking for congestion price estimation," In Proceeding of IEEE INFOCOM, Hong Kong, pp. 12−23, 2004.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 378-384
doi:10.4236/ijcns.2009.25042 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Notification Services for the Server-Based Certificate Validation Protocol

**Johannes BUCHMANN, Vangelis KARATSIOLIS**
*Technische Universität Darmstadt, Department of Computer Science, Cryptography and Computeralgebra, Darmstadt, Germany*
*Email*: {*buchmann,karatsio*}*@cdc.informatik.tu-darmstadt.de*
*Received May* 15, 2009; *revised June* 17, 2009; *accepted July* 22, 2009

## ABSTRACT

The Server-Based Certificate Validation Protocol allows PKI clients to delegate to a server the construction or validation of certification paths. The protocol's specification focuses on the communication between the server and the client and its security. It does not discuss how the servers can efficiently locate the necessary PKI resources like certificate or certificate revocation lists. In this paper we concentrate on this topic. We present a simple and effective method to facilitate locating and using various PKI resources by the servers, without modifying the protocol. We use the extension mechanism of the protocol for notifying the servers about PKI repositories, certificates, and revocations. We specify the tasks of the servers and certificate issuers and define the messages that are exchanged between them. A proof of concept is given by implementing an SCVP server, a client, and the proposed method in Java.

**Keywords:** SCVP, Certification Path, Certification Path Construction, Certification Path Validation, X.509 Certificate

## 1. Introduction

A Public Key Infrastructure (PKI) has lots of protocols and processes that support important functions of the infrastructure. The building of a certification path and its validation are two of them. The PKI clients need to perform these operations before they can securely use an X.509 certificate. But there are clients that are not able or they simply do not want to perform certification path construction or validation themselves.

For these clients a protocol has been specified by the IETF. This is the server-based certificate validation protocol (SCVP) [1]. This protocol allows clients to delegate the building or validation of a certification path to a server.

Once a request reaches the server, the server tries to build the certification path. For performing this task it needs to contact various repositories and download certificates and CRLs. It is not always possible to construct such a path, if the repositories are not reachable by the server. Further, it is not always possible to locate the correct certificates or CRLs. Moreover, certification authorities that operate an SCVP server need to configure this server in such a way that it is able to efficiently locate their resources.

In this paper we concentrate on such implementation issues. Especially we see how to use the extension mechanisms of the protocol to provide the SCVP server with important resources for its functioning. These are for example the trust anchors of a PKI, the revocation lists, or the location of repositories. We show how to create appropriate messages that are sent by a purpose-specific client charged with this task by the CA. We also present the prototype implementation of an SCVP server in Java. This SCVP server is notified about the PKI resources by a notification client using our proposed method.

### 1.1. Notation

For the rest of the paper we denote by $C^A_B$ the certificate issued to entity *B* by entity *A*. By *CP: [*$C^A_C$ *and* $C^C_E$*]* we

Copyright © 2009 SciRes. 

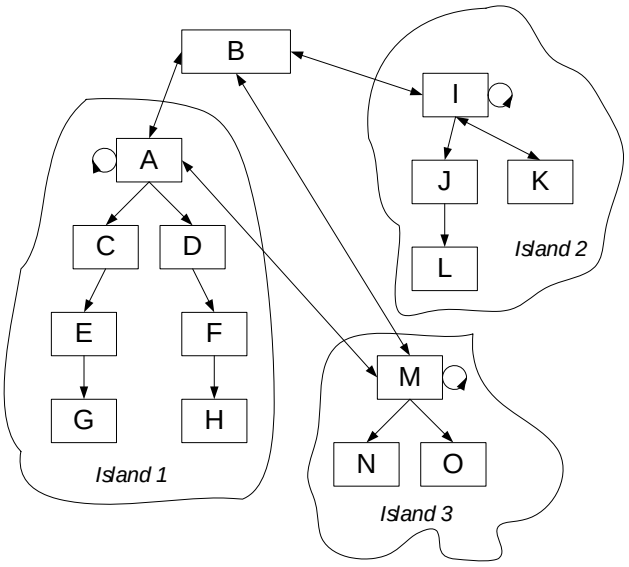



**Figure 1. An example of a PKI topology with three independent infrastructures (islands) and a CA acting as Bridge (entity B).**

denote the certification path which consists of the certificates $C^A_C$ and $C^C_E$. We will use the PKI topology depicted in Figure 1 in our examples. The boxes represent entities and the arrows represent certificates issued by one entity to another (in the arrow direction).

This paper is organised as follows: In Section 2 we briefly discuss certification path construction and validation. In Section 3 we describe the SCVP. In Section 4 we present the extension of the protocol for sending notifications to the server. In Section 5 we give the prototype implementation of an SCVP server which uses our proposed method. We conclude our work in Section 6.

## 2. Certification Path Building and Validation

One client receives a certificate. The client wants to verify whether the binding of the public key and the certificate's subject (found in the subject distinguished name and/or the subject alternative name) is valid [2]. For verifying that, the client needs, among others, all certificates in the certification chain between one of its trust anchors and the certificate in question. Suppose for example that the client wants to verify the certificate $C^E_G$ and it possesses one trust anchor. The trust anchor is entity *A*. In this case the certification path is *CP: [*$C^A_C$, $C^C_E$, *and* $C^E_G$*]*. The certificate of the trust anchor $C^A_A$ is not part of the certification path. Building this certification path can be easy. But more complicated paths are necessary to be constructed. For example, if the same certificate $C^E_G$ needs to be verified but the trust anchor is entity *K*, then the construction of this certification path is more complex. Guidelines for building certification paths are given in [3].

If a client does not want to perform certification path construction on its own then it can delegate this to the SCVP server. The server will then try to construct the path (following the guidelines from [3]).

The validation of the certification path is the next step in the verification. The most commonly used algorithm for this purpose is described in Section 6 of RFC 3280 [2]. This algorithm takes as input the certification path, the current time of the validation, the set of allowed policies, some other policy related parameters, and information about the trust anchor. The last is the name of the trust anchor, the algorithm of its public key (with optional parameters), and the public key itself. This information is trusted. The algorithm outputs the result of the validation and, in case of successful validation, the public key that has been validated (with parameters and algorithm) and policy related information. An SCVP server must implement this algorithm (see [1]).

Certification path validation assumes that a certification path already exists. Therefore validation of a certification path implies that a certification path building process has been already conducted.

For attribute certificates [4] these processes are similar. The default validation algorithm is described in [4].

## 3. The Server-Based Certificate Validation Protocol (SCVP)

SCVP [1] is a protocol specified by the IETF. The goal of the protocol is to allow clients that cannot perform certification path building or certification path validation to delegate this task to a server. A reason for doing this is that the clients cannot locate the resources themselves or they do not support the necessary protocols (for example OCSP [5]). The process of delegating the certification path building is also known as DPD (delegated path discovery) and this of delegating the validation as DPV (delegated path validation). They are defined along with their requirements in [6].

For delegating the above tasks, the client sends a CVRequest [1] (see Figure 2) to the server. This request can be signed or a MAC value can be calculated over the request and be sent with it. In these two cases the request is encapsulated in a CMS [7] message.

The query (of the type *Query*) contains the certificates for which the clients request the certification path to be built or validated. The specification of a query can be seen in Figure 3. It is possible to define whether the certification path should be built, validated, or validated with revocation checking. This is specified in the *checks*. The protocol also allows the client to specify the type of

Copyright © 2009 SciRes. 

```
CVRequest ::= SEQUENCE {
cvRequest Version     INTEGER DEFAULT 1,
query                 Query,
requestorRef          [0] GeneralNames OPTIONAL,
requestNonce          [1] OCTET STRING OPTIONAL,
requestorName         [2] GeneralName OPTIONAL,
responderName         [3] GeneralName OPTIONAL,
requestExtensions     [4] Extensions OPTIONAL,
signatureAlg          [5] AlgorithmIdentifier OPTIONAL,
hashAlg               [6] OBJECT IDENTIFIER
                      OPTIONAL,
requestorText         [7] UTF8String (SIZE (1..256))
                      OPTIONAL}
```

**Figure 2. CVRequest.**

```
Query ::= SEQUENCE {
queriedCerts          CertReferences,
checks                CertChecks,
wantBack              [1] WantBack OPTIONAL,
validationPolicy      ValidationPolicy,
responseFlags         ResponseFlags OPTIONAL,
serverContextInfo     [2] OCTET STRING OPTIONAL,
validationTime        [3] GeneralizedTime OP TIONAL,
intermediateCerts     [4] CertBundle OPTIONAL,
revInfos              [5] RevocationInfos OPTIONAL,
producedAt            [6] GeneralizedTime OPTIONAL,
queryExtensions       [7] Extensions OPTIONAL }
```

**Figure 3. Query.**

```
ValidationPolicy ::= SEQUENCE {
validationPolRef      ValidationPolRef,
validationAlg         [0] ValidationAlg OPTIONAL,
userPolicySet         [1] SEQUENCE SIZE (1..MAX) OF
                      OBJECT IDENTIFIER OPTIONAL,
inhibitPolicyMapping  [2] BOOLEAN OPTIONAL,
requireExplicitPolicy [3] BOOLEAN OPTIONAL,
inhibitAnyPolicy      [4] BOOLEAN OPTIONAL,
trustAnchors          [5] TrustAnchors OPTIONAL,
keyUsages             [6] SEQUENCE OF KeyUsage OP-
                      TIONAL,
extendedKeyUsages     [7] SEQUENCE OF KeyPurposeId
                      OPTIONAL,
specifiedKeyUsages    [8] SEQUENCE OF KeyPurposeId
                      OPTIONAL }
```

**Figure 4. Validation policy.**

objects that must be returned by the server. This is covered by the *wantBack* element.

For defining the policies that the server should use for validating a certificate, the *validationPolicy* element is used (see Figure 4).

```
notification OBJECT IDENTIFIER ::=
{1.3.6.1.4.1.8301.3.8.1.1}

Notification ::= SEQUENCE OF EXTENSIONS
```

**Figure 5. Notification request.**

To facilitate the building and validation of certification paths by the server, we extend the CVRequest by providing notifications about PKI resources within the request. Many aspects of the protocol are reused in order not to affect it significantly.

## 4. The Notification Messages

The notification message is a standard CVRequest. To distinguish it as a notification message it contains an extension (as this is defined in [2]) called *Notification*. The specification of the extension and its object identifier (OID) is found in Figure 5.

The Notification is a sequence of already existing extensions that are used in the X.509 based PKI. These extensions can hold all necessary information that is required for notifying the server for new resources. This is specified like that in order to minimise the effort of PKI practitioners to implement the proposed notification method. In addition by being a sequence of extensions it is possible to notify the server about various resources within one notification request. The Notification extension is non-critical.

There are six types of notification. These notify the server about: a) trust anchors, b) other certificates (for example of CRL signers), c) CRLs and delta-CRLs, d) repositories for revocation purposes, e) repositories for certificates, and f) cross certificates.

### 4.1. Notification about Trust Anchors

This type of notification notifies the server about the trust anchors of a PKI. Trust anchors are all entities that are allowed to issue certificates.[1] However, in practice only entities that possess a self-signed certificate are considered trust anchors. Therefore we propose to include only such certificates in this notification.

This Notification is an empty sequence. In the *trustAnchors* element of the ValidationPolicy the trust anchors of the PKI are sent. In our example, for the first PKI island (Island 1) entities *A*, *C*, *E*, *D*, and *F* are certification authorities. In this case the trustAnchors element may consist of five certificates. These are $C^{A}_{A}$, $C^{A}_{C}$, $C^{C}_{E}$, $C^{A}_{D}$, and $C^{D}_{F.}$ We propose to include only $C^{A}_{A}$. The other four certificates can be included in a notification about

[1]Such entities possess a certificate which contains the basic constraints extension and has the value true for the *cA* Boolean flag (see [9]).

Copyright © 2009 SciRes. 

cross certificates (see Subsection 4.6).

Trusting $C^{A}_{A}$ is a very critical operation. If this certificate is not a legitimate one, then the SCVP server may return wrong results. For this reason some PKIs may introduce an out-of-band mechanism that provides the SCVP server with information about which self-signed certificates are trusted or not. One technical realisation of this concept is to have a configuration file, signed by an administrator, which contains the fingerprints of known valid self-signed certificates. The SCVP server compares the fingerprint of the self-signed certificate provided in the request with those in the file. If a match is found then it accepts the certificate, otherwise it discards it. All other (non self-signed) certificates are verified before they are considered trust anchors.

### 4.2. Notification about Other Certificates

There are certificates that are required during a validation but do not belong to certification authorities. These are the certificates of CRL signers (entities that issue indirect CRLs), of OCSP signers, and of SCVP servers. These certificates are used for verifying signatures on revocation lists, on OCSP responses, and on SCVP responses respectively.

These certificates are sent within the *intermediateCerts* element of the Query. The Notification element is an empty sequence.

### 4.3. Notification about Revocation Lists

This type of notification is used for sending the CRLs or delta-CRLs to the SCVP server. To send the CRLs to the server the *revInfos* element of the Query of the type RevocationInfos (see Figure 6) is used. From this element only the *crl* and the *delta-crl* fields are used.

The Notification is an empty sequence. These notifications can also be used in a "push-mode". In this mode the CRLs are sent to the server as soon as they are issued. Such a mechanism is useful in certain environments. In this case the SCVP server has always fresh revocation information.

```
RevocationInfos ::= SEQUENCE SIZE
(1..MAX) OF RevocationInfo

RevocationInfo ::= CHOICE {
    crl         [0] CertificateList,
    delta-crl   [1] CertificateList,
    ocsp        [2] OCSPResponse,
    other       [3] OtherRevInfo }
```

**Figure 6. Revocation Infos.**

**Table 1. Elements of general names.**

| Type of resource | Element of GeneralName |
|---|---|
| LDAP | directoryName |
| X.500 | directoryName |
| Web or FTP | uniformResourceIdentifier |
| HTTP, WebDAV | uniformResourceIdentifier |
| DNS | dNSName |

### 4.4. Notification about Revocation Repositories

It may not be possible or desired that the CA or a CRL issuer sends every CRL to the SCVP server. In addition the location of an OCSP server may be unknown to it. In these cases the SCVP server can be notified about the location where these resources can be found. This is very helpful if the certificates issued by the CA do not contain the CRLDistributionPoint (for CRLs), FreshestCRL (for delta-CRLs), or Authority Information Access (for OCSP) extensions. But even if these values are present, once they are set in a certificate they cannot be changed. This is a problem if the resources have been moved or do not exist at all[1] and they cannot be accessed anymore.

The revocation resources can be located in diverse repositories. Examples of typical repositories that are used in a PKI are X.500 directories [8], LDAP directories [9], DNS servers [10], WebDAV [11], Web or FTP servers [12], or HTTP stores according to [13] specified additionally in [14] as an RFC. To notify about the location of a CRL the CRLDistributionPoint [2] extension is added to the sequence of extensions of the Notification. For the location of delta-CRLs the FreshestCRL [2] extension is added. In these extensions the GeneralNames [2] element is used for specifying the different locations. In Table 1 the elements of GeneralNames that are used for describing the resources are given.

For notifying about the location of OCSP servers, the Authority Information Access [2] extension is added to the list of extensions. It is possible to notify the SCVP server for more than one repository within one notification request.

### 4.5. Notification about Certificate Repositories

The CA may not wish to send any certificates to the SCVP server but just notify it about the repositories in which these are located.

In this case it sends a Notification request which contains the Subject Information Access extension (see [2]). This extension contains the *caRepository* access method. This specifies the location of the repository used by a CA. The value of the location is specified as GeneralName.

[1]Typical example is that of a CA stopping operation.

Copyright © 2009 SciRes. 

The same principles as in the case of the revocation resources apply here. It is possible to send a notification about more than one repository location by defining more AccessDescription elements inside the extension.

### 4.6. Notification about Cross Certificates

Cross certifications may occur any time and the number of cross certificates of a CA can be large. For notifying the servers about such certificates the SCVP notifier sends a Notification request with the Authority Information Access extension present. This extension contains the *caIssuers* access method (see [2]) which points to the location where cross certificates are stored. An example value for this location is: *ldap://host:389/CN=CA, C=DE, DC=Org, DC=COM/ cross CertificatePair; binary? sub? object Class=pkiCA*. This address is found in an LDAP directory and follows the LDAP URL format. This method allows the CA to notify the SCVP server only once, stating where past and possibly future cross certificates can be located.[1] A condition for this notification to function properly is that the CA publishes all cross certificates (issuedTo and issuedBy) in the directory.

An alternative to this approach is to send a Notification as an empty sequence with the cross certificates stored in the *intermediateCerts* element of the Query. These can be distinguished from the certificates discussed in Section 4.2, because the basic constraints extension identifies them as CA certificates. For example, entity *A* sends the certificates $C^A_M$, $C^M_A$, $C^A_B$, *and* $C^B_A$. Certificates $C^A_C$, $C^C_E$, $C^A_D$, *and* $C^D_F$ are also sent within this type of notification.

### 4.7. Summary of the Messages

A summary of the types of notification that can be sent to the SCVP server is given in Table 2. The type of resource for which the notification is performed is given in the first column. The Notification Extension column describes the contents of the Notification extension and the Influenced Element column the element of the CVRequest that is used in the request.

### 4.8. The Notification Client

The notification client is the entity inside a PKI which is responsible for notifying SCVP servers about the resources of the PKI. One choice for being a notification client is the online part of the CA. Another choice is the components that administrate the certificates and revocations and are usually employed for updating an OCSP or an LDAP server.

The CA issues a special certificate to the notification client. A notification client certificate is an X.509 certificate that has the extended key usage (see [9]) extension set. The value of this extension contains only one KeyPurposeId which is identified by the OID "1.3.6.1.4.1.8301.3.8.1.2". The extension is marked critical.

The notification client always sends signed requests to the SCVP server. The client can also check whether the SCVP server has accepted the information that was sent to it and has been successfully updated. This is very useful when the notification messages are used by a CA to update the backend of its own SCVP server. In this case it includes certain certificates within the notification, namely in the *queriedCerts* element of the Query. Typical choices for certificates to include in the query for testing whether the server can build and verify a certification path are valid certificates issued recently by the CA. Another choice is certificates that have been revoked. By asking for a validation of the latest revoked certificate the client can test whether the SCVP server has received the freshest CRL. To properly evaluate the result of the verification the client chooses proper values for the *checks* (see [1]) element of the Query.

**Table 2. Overview of notifications.**

| Type of resource | Notification Extension | Influenced Element |
|---|---|---|
| Trust Anchors | empty | trustAnchors |
| Other Certificates | empty | intermediateCerts |
| CRLs | empty | crl |
| Delta-CRLs | empty | delta-crl |
| CRL Repository | CRLDistributionPoints | none |
| Delta-CRL Repository | FreshestCRL | none |
| OCSP server | Authority Information Access | none |
| Cross Certifications | Authority Information Access | intermediateCerts (opt.) |
| General Repository | Subject Information Access | none |

### 4.9. Implementation Guidelines for the Server

We present a small catalogue of implementation guide lines for SCVP servers that need to be taken into account

[1]This is a supported by the value of the URL. The URL of this example allows a search of unlimited depth beneath this CA entry for all cross certificates in the directory. Other URLs may not be able to support this type of "dynamic" notification.

Copyright © 2009 SciRes. 

for a proper and secure use of the notification requests.

- The notification client must present a valid certificate that has a critical extended key usage extension which contains only the OID "1.3.6.1.4.1.8301.3.8.1.2".
- All notification requests (Notification extension present) that contain a valid signature are accepted. The server may ignore notification requests in some cases, for example when it is overloaded. Notification requests that are neither signed nor have a valid signature are rejected.
- Self-signed certificates provided in a notification request, may be considered trustworthy only if there is an out-of-band mechanism that ensures that these are indeed trusted. This depends on the PKI. All other certificates are verified.
- Information retrieved by the server or provided by the notification client should be stored in a local repository. A database or an LDAP directory can be used for this purpose. When a CVRequest reaches the server, this should try first to access its backend and if no information is found or is not recent enough, then it should try to contact external resources.
- Optionally the SCVP server can forward a notification request to other SCVP servers.

## 5. Design and Implementation

We have designed and implemented a prototype SCVP server and client as well as the proposed notification request. Some features like attribute certificates and delta-crls are not supported in the current implementation.

The SCVP server is implemented as a Java servlet. The servlet container is Apache Tomcat 6.0. The backend of the server is the file system. That is all certificates and revocation lists that are used by the server to answer requests are stored on the hard disc. The server signs its responses with keys stored either in software in the PKCS#12 format or in hardware by using a smart card. The connection to the smart card is realised with the classes contained in *javax.smartcardio* package which are available since Java 6.0. The communication with the card reader and the smart card is done over PC/SC.

The SCVP server operates for the first PKI of our example (Island 1 in Figure 1). The server starts operation without having any certificate or CRL stored in its backend at all. The backend is updated exclusively by notification requests. The test scenario is to notify the server about the trust anchor of the PKI and the other intermediate certificates. Afterwards the client sends a regular CVRequest about a certificate which has not been revoked. The server should be able to build and return a valid certification path. Then, this certificate is revoked, the CA issues a CRL, and the notification client informs the server about the new CRL by sending a notification request as described in our method. The expected answer is that the certificate is revoked. For concentrating only on the performance of the communication all certificates, CRLs, and requests are pre-produced and are just sent to the server.

For testing the implementation and the efficiency of the notification method the client sends 1000 requests to the server. Half of them regard not revoked certificates while the other half revoked ones. When the server signs its requests using keys stored in software it takes approximately 57 ms to answer a request. That is to accept it, verify and process it, create and sign the response and finally send this back to the client. When keys (1024 bits RSA) stored in a smartcard are used the required time is about 480 ms. The server runs on an Intel Core Duo with 1.6 GHz.

The server starts operating without any certificates or CRLs stored in its backend. However, by using the notification method described in the paper it is possible to update its backend and enable it to produce useful and reasonable answers. In addition, when certificates are revoked the server is immediately notified about it and responds taking these revocations into account. Moreover, this server can be used by any PKI that wishes to add SCVP services without modifying the current PKI. Old certificates and CRLs need to be sent once in the beginning and newly produced ones need to be sent to the server as a notification request. Only a notification client should be implemented and customised according to the requirements of the PKI.

## 6. Conclusions

In this paper we presented a simple method for notifying the SCVP servers about PKI resources. We showed the necessary steps that a CA and the SCVP server perform and the messages exchanged between them. This method can be used to notify general purpose SCVP servers as well as the own SCVP server of a CA. This method is very useful when an SCVP server may not be able to locate the resources of a PKI. For example the certificates are stored in a database in which the server does not have access. This is common when PKIs of different organisations are involved. Moreover, this method can be used for forwarding notification requests to other SCVP servers. We also provided a prototype implementation of an SCVP server and client as well as an implementation of the proposed method in Java. It was shown that the method is effective for notifying an SCVP server about certificates and revocation lists.

Copyright © 2009 SciRes. 

## 7. References

[1] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk, "Server-based certificate validation protocol (SCVP)," IETF Request for Comments, Vol. 5055, December 2007.

[2] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," IETF Request for Comments, Vol. 3280, April 2002.

[3] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas, "Internet X.509 public key infrastructure: Certification path building," IETF Request for Comments, Vol. 4158, September 2005.

[4] S. Farrell and R. Housley, "An internet attribute certificate profile for authorization," IETF Request for Comments, Vol. 3281, April 2002.

[5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 internet public key infrastructure online certificate status protocol–OCSP," IETF Request for Comments, Vol. 2560, June 1999.

[6] D. Pinkas and R. Housley, "Delegated path validation and delegated path discovery protocol requirements," IETF Request for Comments, Vol. 3379, September 2002.

[7] R. Housley, "Cryptographic message syntax (CMS)," IETF Request for Comments, Vol. 3852, July 2004.

[8] "Recommendation X.500 ITU-T information technology – open systems interconnection – the directory: Overview of concepts, models and services," August 2005.

[9] J. Sermersheim, "Lightweight directory access protocol (LDAP): The protocol," IETF Request for Comments, Vol. 4511, June 2006.

[10] S. Josefsson, "Storing certificates in the domain name system (DNS)," IETF Request for Comments, Vol. 4398, March 2006.

[11] D. W. Chadwick and S. Anthony, "Using WebDAV for improved certificate revocation and publication," In Proceedings of Public Key Infrastructure: 4th European PKI Workshop: Theory and Practice, EuroPKI, Lecture Notes in Computer Science, Vol. 4582, pp. 265–279, June 2007.

[12] R. Housley and P. Hoffman, "Internet X.509 public key infrastructure operational protocols: FTP and HTTP," IETF Request for Comments, Vol. 2585, May 1999.

[13] P. Gutmann and A. Reliable, "Scalable general-purpose certificate store," In Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC'00), pp. 278–287, December 2000.

[14] P. Gutmann, "Internet X.509 public key infrastructure operational protocols: Certificate store access via HTTP," IETF Request for Comments, Vol. 4387, February 2006.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 385-391
doi:10.4236/ijcns.2009.25043 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Research on Financial Distress Prediction with Adaptive Genetic Fuzzy Neural Networks on Listed Corporations of China

**Zhibin XIONG**
*School of Mathematical Sciences, South China Normal University, Guangzhou, China*
*Email*: *zxiong3@gmail.com*
*Received March* 27, 2009; *revised May* 10, 2009; *accepted June* 28, 2009

## ABSTRACT

To design a multi-population adaptive genetic BP algorithm, crossover probability and mutation probability are self-adjusted according to the standard deviation of population fitness in this paper. Then a hybrid model combining Fuzzy Neural Network and multi-population adaptive genetic BP algorithm—Adaptive Genetic Fuzzy Neural Network (AGFNN) is proposed to overcome Neural Network's drawbacks. Furthermore, the new model has been applied to financial distress prediction and the effectiveness of the proposed model is performed on the data collected from a set of Chinese listed corporations using cross validation approach. A comparative result indicates that the performance of AGFNN model is much better than the ones of other neural network models.

**Keywords:** Multi-Population Adaptive Genetic BP Algorithm, Fuzzy Neural Network, Cross Validation, Financial Distress

## 1. Introduction

In recent years, neural networks (NNs), especially back-propagation NNs (BPNN), are developed and applied quickly to financial distress prediction because of their excellent performances of treating non-linear data with learning capability [1−2]. However, the shortcoming of neural networks is also significant due to a "black box" syndrome and the difficulty in dealing with qualitative information, which limited its applications in practice [3]. Besides, the common NNs also suffer from relatively slow convergence speed and occasionally involve in a local optimal solution.

On the other hand, fuzzy logic as a rule-based development in artificial intelligence can not only tolerate imprecise information, but also make a framework of approximate reasoning. Thus, a number of fuzzy neural networks have been developed to overcome the "black box" syndrome [4−5]. By the same token, genetic algorithm as an effective non-linear modeling system in artificial intelligence not only searches optima speedily but also searches optima globally [6−9]. Therefore, it is possible that combining neural networks with fuzzy logic and genetic algorithm could merge their advantages and meanwhile overcome the disadvantages mentioned above.

In this study, a new genetic algorithm combined with neural network, named as Adaptive Genetic Fuzzy Neural Network (AGFNN), is presented to predict financial distress on the data collected from a set of Chinese listed corporations, and the results indicate that the performance of AGFNN model is much better than the ones of other NN models.

The rest of this paper is organized as follows. Section 2 introduces the structure of the AGFNN model and associated algorithm. Section 3 describes the data source and methodology. In Section 4, the proposed model is predicted financial distress using the data from Chinese listed corporations, and its performances are compared with the other NN models. Finally, some concluding remarks are drawn from Section 5.

Copyright © 2009 SciRes. 

## 2. Architecture of AGFNN and Its Algorithm

### 2.1. Architecture of AGFNN

In this study, the proposed AGFNN model is based on the fuzzy neural networks [6]. A fuzzy neural network consists of a set of fuzzy if-then rules that describe the input-output mapping relationship of the network. The antecedents of fuzzy rules partition the input space into a number of linguistic term sets while the consequent constituent can be chosen as a fuzzy membership function (Mamdani model), a singleton value, or a function of a linear combination of input variables (TSK model).

For simplicity, the singleton consequent of fuzzy rules is adopted in this paper. The fuzzy rule with singleton consequent can be given in the following form:

Rule $k$: if $x_1$ is $A_{1k}$ and $x_2$ is $A_{2k}$ … and $x_n$ is $A_{nk}$, then

$$y = b_k \tag{1}$$

where $x_i$ is the input variable, $y$ is the output variable, $A_{ik}$ is the linguistic term of the precondition part, $b_k$ is the constant consequent part, and $n$ is the number of input variables.

The structure of a fuzzy neural network is shown in Figure 1, where $n$ and $m$ are the number of input variables and the number of fuzzy sets respectively. It is a four layer network structure. We use $I_i^{(l)}, O_i^{(l)}$, $l = 1,2,3,4$ denote the input and output of the $i^{\text{th}}$ node in layer $L$ respectively. The functions of the nodes in each layer are described as follows:

**Layer 1:** The number of nodes is $n$ in this layer. The nodes only transmit input values to layer 2:

$$I_i^{(1)} = x_i, O_i^{(1)} = I_i^{(1)}, i = 1,2,...,n; \tag{2}$$

**Layer 2:** Nodes in this layer correspond to one linguistic label of the input variables in layer 1; that is, the membership value specifying the degree to which an input value belongs to a fuzzy set is calculated in this layer.

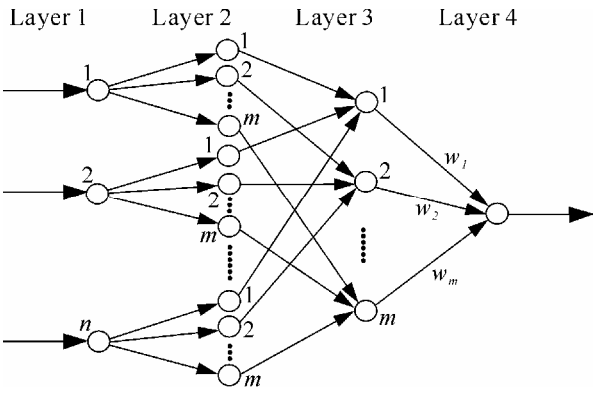



**Figure 1. Structure of AGFNN.**

The input and output in this layer are formulated as follows:

$$I_{ij}^{(2)} = O_i^{(1)}, O_{ij}^{(2)} = \mu_{ij}(x_i) = \exp\left[-\left(\frac{x_i - m_{ij}}{\sigma_{ij}}\right)^2\right] \tag{3}$$
$$i = 1,2,...,n, j = 1,2,...,m$$

where $\mu_{ij}(x_i)$ is fuzzy membership function, $m_{ij}$ and $\sigma_{ij}$ are, respectively, the center and the width of the Gaussian membership function $\mu_{ij}$.

**Layer 3:** There are m nodes in this layer. The output of each node in this layer is determined by the fuzzy AND operation. Here, the product operation is utilized to determine the firing strength of each rule. The input and output in this layer are formulated as follows:

$$I_j^{(3)} = \prod O_{ij}^{(2)} = \prod \mu_{ij},\ O_j^{(3)} = I_j^{(3)}, \tag{4}$$
$$j = 1,2,...,m, i = 1,2,...,n$$

**Layer 4:** The single node in this layer computes the overall output as the sum of all incoming signals. The input and output in this layer are formulated as follows:

$$I^{(4)} = \sum_{j=1}^{n} w_j O_j^{(3)},\ O^{(4)} = I^{(4)} \tag{5}$$

where the $w_j$ is the weight associated the $j$-th node in layer 3 with the single node in layer 4 (output layer).

### 2.2. Algorithm of AGFNN

In this study, a modified algorithm, multi-population adaptive genetic BP algorithm (MAGBPA), is proposed to optimize the parameters of this model included the weight, the center and the width of the membership function. In order to improve the performance of evolutionary algorithm, First, this paper employs a multi-population genetic algorithm where a large population is divided into smaller subpopulations and subpopulations cooperate and compete with each other. Secondly, a adaptive genetic algorithm where crossover probability pc and mutation probability pm are self-adjusted according to the standard deviation of population fitness. pc and pm are defined as follows:

$$P_c = P_{c1} + \frac{(P_{c2} - P_{c1})(\delta - \delta_{\min})}{\delta_{\max} - \delta_{\min}} \tag{6}$$

$$P_m = P_{m1} + \frac{(P_{m2} - P_{m1})(\delta - \delta_{\min})}{\delta_{\max} - \delta_{\min}} \tag{7}$$

Copyright © 2009 SciRes. 

where $p_{c1}$ and $p_{c2}$ are the minimum and maximum value of crossover probability set in advance respectively. $p_{m1}$ and $p_{m2}$ are the maximum and minimum value of mutation probability set in advance respectively. $\delta_{\max}$ and $\delta_{\min}$ are the maximum and minimum value of standard deviation of population fitness set in advance respectively. $\delta$ is value of standard deviation of currently population fitness.

However, while GA is very effective at global search can quickly isolate global minimum, it may be inefficient at actually finding that minimum. Therefore, back-propagation (BP) operator is introduced into GA. That is, BP is used for the fine-tuned search when the GA is used to isolate the global minimum. The detailed procedure of algorithm consists of the following steps:

**Step 1:** Let $t = 0$ and randomly produce an initial population $P(t)$ which divided into $N$ ($N > 1$) subpopulation averagely. In this study, the size of population is selected from 30-100.

**Step 2:** Use real numbers to code values of the model's parameters. The values of $w$ are limited in the interval of [-100, 100]. The values of $m$ and the values of $\sigma$ are limited in the interval of [-2, 2] and interval of (0, 2] respectively. The initial parameter values are given randomly in terns of distribution of $e^{-|r|}$.

**Step 3:** Calculate the value of fitness for each individual of each subpopulation. We use the reciprocal of SSE as the function of evaluation, that is: $f = 1/\sum_j E_j^2$ .at same time, calculate $\delta_0$ (the value of standard deviation of initial population fitness) and we can regard $\delta_0$ as the initial values of $\delta_{\max}$ and $\delta_{\min}$ .

**Step 4:** Execute Step 5 and Step 6 in each subpopulation independently.

**Step 5:** In each subpopulation, genetic operators are executed. Roulette-wheel is used in the selection and adaptive strategies are used in crossover and mutation. The selection of each individual depends on the probability that is proportional to its degree of fitness. Self-adjusted crossover probability $p_c$ and a seif-adjusted mutation probability $p_m$ are used in this study according to the Equation (6) and the Equation (7).

**Step 6:** A number of individuals are distributed among subpopulations by means of migration operator after the subpopulations evolve independently for a certain number of generations (isolation time). A fixed migration rate $p_{mr}$ is employed, and the selection of individuals for migration is based on the value of fitness. The migration of topology is neighborhood topology that migration is made only between nearest neighbors. These exchanged individuals participate in the evolution process and produce new subpopulations.

**Step 7:** Calculate the value of $\delta$ when the new population is produced. If $\delta > \delta_{\max}$ , then $\delta_{\max} = \delta$ ; if $\delta < \delta_{\min}$ , then $\delta_{\min} = \delta$ .

**Step 8:** Once the new subpopulations are produced, use BP algorithm to make each individual learn from the data sample. Update the parameter values by using the follow formulas:

$$m_{ij}(t+1) = m_{ij}(t) - \eta \frac{\partial E_p}{\partial m_{ij}}, \tag{8}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \frac{\partial E_p}{\partial \sigma_{ij}}, \tag{9}$$

$$w_j(t+1) = w_j(t) - \eta \frac{\partial E_p}{\partial w_j}, \tag{10}$$

where $\frac{\partial E_p}{\partial m_{ij}}$, $\frac{\partial E_p}{\partial \sigma_{ij}}$ and $\frac{\partial E_p}{\partial w_{ij}}$ are determined by the follow equations

$$\frac{\partial E_P}{\partial m_{ij}} = \frac{\partial E_P}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial m_{ij}} = (y - Y) w_j \prod_{l=1, l\neq i}^{n} \mu_{lj} \bullet \exp[-(\frac{x_i - m_{ij}}{\sigma_{ij}})^2] \frac{2(x_i - m_{ij})}{\sigma_{ij}^2} \tag{11}$$

$$\frac{\partial E_P}{\partial \sigma_{ij}} = \frac{\partial E_P}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial \sigma_{ij}} = (y - Y) w_j \prod_{l=1, l\neq i}^{n} \mu_{lj} \bullet \exp[-(\frac{x_i - m_{ij}}{\sigma_{ij}})^2] \frac{2(x_i - m_{ij})^2}{\sigma_{ij}^3} \tag{12}$$

$$\frac{\partial E_P}{\partial w_j} = \frac{\partial E_P}{\partial y} \frac{\partial y}{\partial w_j} = (y - Y) \prod_{i=1}^{n} \mu_{ij} \tag{13}$$

**Step 9:** Check the termination criterion. In this study, we use two measures that the mean squared error (*MSE*) and the maximal evolution generations. That is, if the *MSE*< $\varepsilon$ or the number of maximal evolution generations is *M*, where ε and *M* are values set in advance, then stop algorithm, otherwise let $t = t+1$ and go to Step 3 until termination criterion is reached.

The diagram of multi-population adaptive genetic BP algorithm is shown in Figure 2.

## 3. Data Source and Pretreatment

The data of samples used in this study are selected from the listed corporations of China. Those sample co- rporations can be divided into two categories: ST (special treatment) corporations and the normal corporations. The

Copyright © 2009 SciRes. 

main reason that listed corporations become ST corporations is due to the bad financial status. So, the ST corporations denote the corporations in financial distress and the normal corporations denote the corporations in financial non-distress in this study.

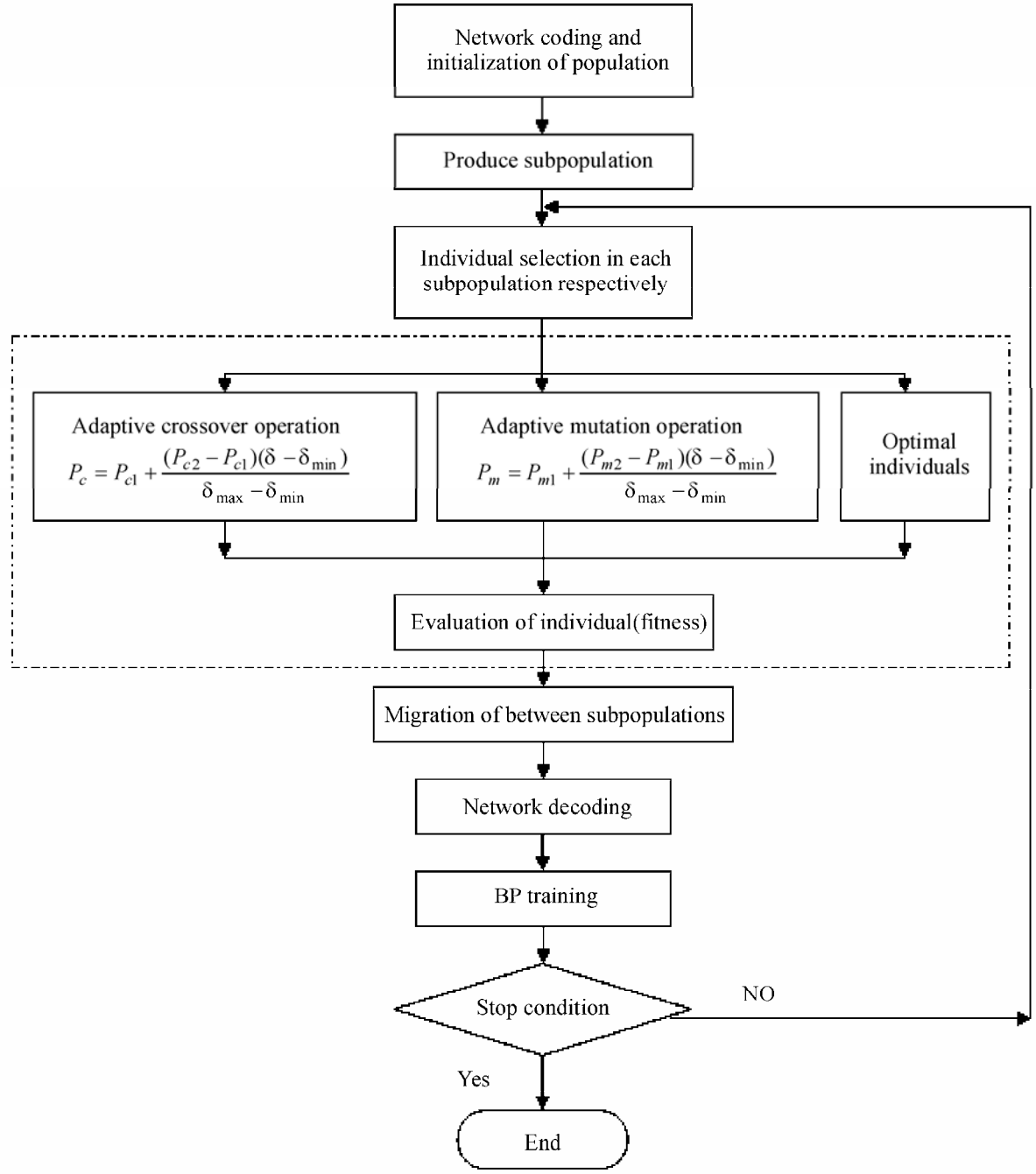



**Figure 2. Diagram of multi-population adaptive genetic BP algorithm.**

Jain and Nag (1997) observed that classification accuracy based on balanced samples represent a poor metric for assessing the effectiveness of NNs in predicting financial distress and bankruptcy. Because the proportion of financial failures among publicly traded entities is very small, it is mathematically possible for a model to have a predictive accuracy that exceeds 50% (better than a decision rule based on a fair coin test) and still perform very poorly when faced with real-world data. So they suggested that NN models should be validated with an

Copyright © 2009 SciRes. 

unbalanced sample that includes a realistically small proportion of failed companies [10]. According to this suggestion, this paper selects 188 observations which include 47 ST and 141 normal corporations from 2004 to 2005. The detail of the number of selected corporations is shown in Table 1.

**Table 1. Number of selected corporations.**

| ST Periods | 2004 | 2005 | Sum |
|---|---|---|---|
| ST-corp | 26 | 21 | 47 |
| Normal-corp | 78 | 63 | 141 |
| | | Total: | 188 |

The data set is divided into two subsets: one is a training sample set with 120 corporations including 30 ST corporations and 90 normal corporations, used to design the common NNs model and the GFNN model; another is a test sample set with 68 corporations including 17 ST and 51 normal corporations to test the performance of models.

In this study, 10 financial variables are selected in this data according to the prior studies [4–5], which include: net profit to total assets (X1), ratio of main business profit (X2), return on equity (X3), total liabilities to total assets (X4), quick ratio (X5), interest coverage ratio (X6), working cash to total liability (X7), turnover of total assets (X8), turnover of accounts receivable (X9) and growth ratio of main business income (X10).

Function $Y = \frac{2(X - \min x)}{(\max x - \min x)} - 1$ is used to treat input data, where $X$ is input matrix, max $x$ and min $x$, respectively, denote the maximum element and minimum element of matrix $X$. So, the number of input layer nodes is 10. The value of parameter m is set to 3. The number of layer 2 nodes and the number of layer 3 nodes are 30 and 3 respectively. The output variable is a single variable $y$, that is, output layer has one node. $y$ = 1 denotes the corporation is in distress, $y$ = 0 denotes the corporation is normal in this study. The actual structure of AGFNN is a 10-30-3-1 network model.

## 4. Empirical Results

In order to minimize the possible bias associated with the random sampling of the training and testing samples, researchers tend to use $n$-fold cross-validation scheme in evaluating the classification capability of the built model. In $n$-fold cross-validation, the entire dataset is randomly split into $n$ mutually exclusively subsets (also called folds) of approximately equal size with respect to the ratios of different populations. The classification model will then be trained and tested $n$ times. Each time the model is built using ($n$-1) folds as the training sample and the remaining single fold is retained for testing. The training sample is used to estimate the credit risk model's parameters while the retained holdout sample is used to test the generalization capability of the built model. The overall classification accuracy of the built model is then just the simple average of the $n$ individual accuracy measures [11]. As cross-validation is the preferred procedure in testing the out-of-sample classification capability when the dataset size is small and the size of bad credit corporations is only 47, the four-fold cross-validation will be adopted in this study. Therefore there are 47 corporations in each fold of the dataset.

In this study, suppose that the population size is 80, the number of subpopulations is 4, initial crossover probability is 0.75, initial mutation probability is 0.025, $p_{c1}$ and $p_{c2}$ are set 0.7 and 0.9 respectively; $p_{m1}$ and $p_{m2}$ are set 0.05 and 0.001 respectively; migration rate is 0.4, isolation time is 20 generations, the number of learning with BP algorithm in each generation is 5 times and the learning rate $\eta$ is 0.06.

Termination criterion of sample learning as follows:

(1) $\varepsilon < 0.001$ or (2) the maximum number of genetic generations = 2000.

The built-in multi-population adaptive genetic BP algorithm based fuzzy neural networks program attached in the toolbox of MATLAB is used for data processing in this study.

Four networks credit risk models were built and the classification results of the corresponding testing samples were summarized in Table 2. From the results in Table 2, we can observe that the average correct classification rates for the four folds are 93.62, 89.36, 91.49 and 93.62%, respectively, with the mean equals to 92.02%.

In the meanwhile, two models, a BPNN model (employ BP algorithm) and a classic model-ANFIS (adaptive network-based fuzzy inference system) model, are designed for comparison in this study*.

**Table 2. Cross-validation testing results of AGFNN.**

| Folder number | Financial distress prediction results | | |
|---|---|---|---|
| | (1-1) | (2-2) | Average correct classification rate |
| 1 | 94.29% (33/35) | 91.67% (11/12) | 93.62% (44/47) |
| 2 | 91.43% (32/35) | 83.33% (10/12) | 89.36% (42/47) |
| 3 | 94.29% (33/35) | 83.33% (10/12) | 91.49% (43/47) |
| 4 | 94.44% (34/36) | 90.91% (10/11) | 93.62% (44/47) |
| Mean | 93.62% (132/141) | 87.23% (41/47) | 92.02% (173/188) |

Here a Class 1 corporation is defined as a corporation with financial non-distress while a Class 2 corporation is the one with financial distress.

*See the Literature 12 and 13 for a more detailed design of the BPNN and ANFIS models..

Copyright © 2009 SciRes. 

**Table 3. Cross-validation testing results of BPNN.**

| Folder number | Financial distress prediction results | | |
|---|---|---|---|
| | (1-1) | (2-2) | Average correct classification rate |
| 1 | 77.14% (27/35) | 75% (9/12) | 76.6% (36/47) |
| 2 | 82.86% (29/35) | 66.67% (8/12) | 78.72% (37/47) |
| 3 | 80% (28/35) | 66.67% (8/12) | 76.6% (36/47) |
| 4 | 80.56% (29/36) | 72.73% (8/11) | 78.72% (37/47) |
| Mean | 80.14% (113/141) | 70.21% (33/47) | 77.66% (146/188) |

**Table 4. Cross-validation testing results of ANFIS.**

| Folder number | Financial distress prediction results | | |
|---|---|---|---|
| | (1-1) | (2-2) | Average correct classification rate |
| 1 | 82.86% (29/35) | 83.33% (10/12) | 82.98% (39/47) |
| 2 | 88.57% (31/35) | 75% (9/12) | 85.11% (40/47) |
| 3 | 85.71% (30/35) | 75% (9/12) | 82.98% (39/47) |
| 4 | 86.11% (31/36) | 81.82% (9/11) | 85.11% (40/47) |
| Mean | 85.82% (121/141) | 78.72% (37/47) | 84.04% (158/188) |

**Table 5. Summarized testing results of the three constructed models**

| Credit model | Financial distress prediction results | | |
|---|---|---|---|
| | (1-1) | (2-2) | Average correct classification rate |
| BPNN | 80.14% (113/141) | 70.21% (33/47) | 77.66% (146/188) |
| ANFIS | 85.82% (121/141) | 78.72% (37/47) | 84.04% (158/188) |
| AGFNN | 95.04% (134/141) | 82.98% (39/47) | 92.02% (173/188) |

The testing results of the four built BPNN models can be summarized in Table 3. From the results in Table 3, we can conclude that the average correct classification rates for the four folds are 76.6, 78.72, 76.6 and 78.72%, respectively, with the mean equals to 77.66%. Similarly, the prediction results of the four built ANFIS models are summarized in Table 4. From the results in Table 4, we can conclude that the average correct classification rates for the four folds are 82.98, 85.11, 82.98 and 85.11%, respectively, with the mean equals to 84.04%.

In order to evaluate the effectiveness of the proposed AGFNN model, the classification results are also compared with those using BPNN and ANFIS models. Table 5 summarizes the average classifications results of BPNN, ANFIS and AGFNN models. As shown on the Table 5, AGFNN model outperforms BPNN and ANFIS models whether on identifying financial normal corporations or identifying financial distress corporations. Moreover, we can see that the average correct classification rate of BPNN and ANFIS models are 77.66% and 84.04% respectively, while the average correct classification rate of AGFNN model is 92.02% on Table 5. So, it can be conclude that the AGFNN model has the best financial distress prediction capability in terms of the correct classification rate from Table 5.

## 5. Conclusions

In this paper, an improved method for the financial distress prediction has been proposed. This method, namely hybrid model, combines fuzzy neural network and multi-population adaptive genetic back-propagation algorithm, named as Adaptive Genetic Fuzzy Neural Network (AGFNN). In this model, a modified learning algorithm, multi-population adaptive genetic BP algorithm (MAGBPA), is proposed to adjust the parameters for the desired outputs. In the MAGBPA method, multi- population adaptive genetic algorithm is used to obtain a rough solution quickly and avoid local optimal solution, and then BP algorithm is used to fine-tune the results. In the end of this study, we investigate the performances of the BPNN model, ANFIS model and AGFNN model on financial distress prediction using the cross-validation approach, based on a set of financial data selected from China listed corporations from 2004 to 2005. The empirical results are shown on a series of tables (from Table 2 to Table 5). The results indicate that the performance of AGFNN is much better than the BPNN and ANFIS models and show that the proposed AGFNN model is promising in financial distress prediction.

## 6. Acknowledgement

This work is supported by the National Science Foundation of China (NSFC), Grant 70371029.

Copyright © 2009 SciRes. 

## 7. References


[1] P. Jackson and W. Perraudin, "Regulatory implications of credit risk modeling," Journal of Banking and Finance, Elsevier Science Inc., Vol. 24, pp. 1−14, 2000.

[2] T. G. Calderon and J. J. Cheh, "A roadmap for future neural networks research in auditing and risk assessment," International Journal of Accounting Information Systems, Elsevier Science Inc. Vol. 3, pp. 203−235, 2002.

[3] A. F. Shapiro, "The merging of neural networks, fuzzy logic, and genetic algorithms," Insurance, Mathematics and Economics, Elsevier Science Inc., Vol. 31, pp. 115−131, 2002.

[4] R. Malhotra and D. K. Mahotra, "Differentiating between good credits and bad credits using neuro-fuzzy systems," European Journal of Operational Research, Elsevier Science Inc., Vol. 136, pp. 190–211, 2002.

[5] S. Piramuthn, "Financial credit risk evaluation with neural and neuro-fuzzy systems," European Journal of Operational Research, Elsevier Science Inc., Vol. 112, pp. 310–321, 1999.

[6] C. J. Lin and Y. J. Xu, "A self-adaptive neural fuzzy network with group-based symbiotic evolution and prediction applications," Fuzzy Sets and Systems, Elsevier Science Inc., Vol. 157, pp. 1036−1056, 2006.

[7] R. Sikora and S. Piramuthu, "Framework for efficient feature selection in genetic algorithm based data mining," European Journal of Operational Research, Elsevier Science Inc., Vol. 180, pp. 723−737, 2007.

[8] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," Systems, Man and Cybernetics, IEEE Transactions on, Vol. 24, No. 4, pp. 656−667, 1994.

[9] X. T. Guo and Y. Zhu, "Evolutionary neural networks based on genetic algorithms," Journal of Qinghua University (National Science Edition), Vol. 40, No. 10, pp. 116−119, October 2000.

[10] B. A. Jain and B. N. Nag, "Performance evaluation of neural network decision models," Journal of Manage Information Systems, Elsevier Science Inc., Vol. 14, pp. 201−216, 1997.

[11] R. A. Johnson and D. W. Wichern, "Applied multivariate statistical analysis," 5th Edition, Prentice-Hall, Upper Saddle River, N. J., 2002.

[12] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing, matlab curriculum series," Prentice-Hall, Englewood Cliffs, N. J., 1997.

[13] Z. B. Xiong and R. –J. Li, "Credit risk evaluation with fuzzy neural networks on listed corporations of China," Proceedings of IWVDVT, China, pp. 397−402, May, 2005.


Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 392-399
doi:10.4236/ijcns.2009.25044 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Enhancing Delay in MANET Using OLSR Protocol

**N. ENNEYA, K. OUDIDI, M. ELKOUTBI**
*E.N.S.I.A.S, Laboratory SI2M, University Mohammed V-Souissi, Rabat, Morocco*
*Email*: {*enneya,oudidi,elkoutbi*}*@gmail.com*
*Received October* 30, 2008; *revised January* 22, 2009; *accepted March* 31, 2009

## ABSTRACT

The performance of a Mobile Ad hoc Network (MANET) is closely related to the capability of the implemented routing protocol to adapt itself to unpredictable changes of topology network and link status. The Optimized Link State Routing (OLSR) protocol is a one key of the proactive routing protocols for MANETs. It is based on the multi-point relays (MPRs) technique to reach all nodes in the network with a limited number of broadcasts. In this paper, we propose new versions of the original OLSR protocol based on a new mobility parameter, in the goal to enhance and adapt it in the presence of the mobility. For this objective we define new three criterions for MPRs selection. The first criteria take for selection, just the mobility of nodes at one-hop. The two others criterions are based on both mobility of nodes at one-hop and two-hops.

**Keywords:** Ad Hoc Networks, OLSR Protocol, Multipoint Relays, Node Mobility Degree, Mobility Quantification

## 1. Introduction

A Mobile Ad hoc Network (MANET) is a collection of mobile nodes (MNs) that cooperatively communicate with each other without any pre-established infrastructures such as a centralized access point. These nodes may be computers or Devices such as laptops, PDAs, mobile phones, pocket pc with wireless connectivity are commonly used. Due to the fact that MNs change their physical location by moving around, the network topology may change unpredictably. This causes changes of link status between each MN and its neighboring. Thus, MNs which join and/or leave the communication range of MN in the network will surely change its relationship with its neighbors by detection of a new link breakages and/or link additions. In the same way, the change of the all routes printed by this MN is also based on the relationship. This change of routes is made with an overhead traffic in the process of maintenance routes assured by the implemented routing protocol in a MANET. For resume, the performance of a MANET is closely related to the capability of the routing protocols to adapt themselves to unpredictable changes of topology network and link status [23,24].

One of the most important aspects of the communication process is the design of routing protocols used to establish and maintain multi-hop routes to allow data communication between nodes. Several researches have been done in this area, and many multi-hop routing protocols have been developed. The Optimized Link State Routing (OLSR) protocol [1,2], Dynamic Source Routing protocol (DSR) [5], Ad Hoc on Demand Distance Vector protocol [6], Temporally Ordered Routing Protocol (TORA) [12], and others protocols that establish and maintain routes on a best-effort basis. There are three main categories of MANET routing protocols: Proactive (table-driven), Reactive (on-demand) and Hybrid. Proactive protocols build their routing tables continuously by broadcasting periodic routing updates through the network; reactive protocols build their routing tables on demand and have no prior knowledge of the route they will take to get to a particular node. Hybrid protocols create reactive routing zones interconnected by proactive routing links and usually adapt their routing strategy to the amount of mobility in the network.

In this paper, we present a new quantitative measure of node mobility reflecting the mobility degree at each node in the MANET. This node mobility degree is re-

Copyright © 2009 SciRes. 

lated to the link status change in the vicinity of the communication range. Therefore, based on this mobility quantification at each MN in the MANET, we have proposed three versions of the original OLSR protocol to enhance and adapt it in the presence of high mobility, i.e. high topology and link status changes.

The rest of this paper is organized as follows. Section 2 gives an overview of the original OLSR protocol. Section 3, presents our proposed node mobility degree. Section 4 presents performance metrics for evaluating performance of routing protocols. In Section 5, simulations and results are given. The last section concludes and presents some future works.

## 2. Optimized Link State Routing Protocol

### 2.1. Overview

The optimized link state routing (OLSR) protocol [1] is a proactive routing protocol that employs an efficient link state packet forwarding mechanism called *multipoint relaying.* This protocol optimizes the pure link state routing protocol. Optimizations are done in two ways: by reducing the size of the control packets and by reducing the number of links that are used for forwarding the link state packets. The reduction in the size of link state packets is made by declaring only a subset of the links in the link state updates. These subsets of links or neighbors that are designated for link state updates and are assigned the responsibility of packet forwarding are called *multipoint relays.* The optimization by the use of multipoint relaying facilitates periodic link state updates. The link state update mechanism does not generate any other control packet when a link breaks or when a link is newly added. The link state update optimization achieves higher efficiency when operating in highly dense networks. The Figure 1(a) shows the number of message transmissions required when the typical flooding-based approach is employed. In this case, the number of message transmissions is approximately equal to the number of nodes that constitute the network. The set consisting of nodes that are multipoint relays is referred to as *MPRset.* Each given node in the network selects an *MPRset* that processes and forwards every link state packet that this node originates (see Figure 1(b)). The neighbor nodes that do not belong to the *MPRset* process the link state packets originated by node *P* but do not forward them. Similarly, each node maintains a subset of neighbors called *MPR selectors,* which is nothing but the set of neighbors that have selected the node as a multipoint relay. A node forwards packets that are received from nodes belonging to its *MPRSelector* set. The members of both *MPRset* and *MPRSelectors* keep changing over time. The members of the *MPRset* of a node are selected in such a manner that every node in the node's two-hop neighborhood has a bidirectional link with the node.

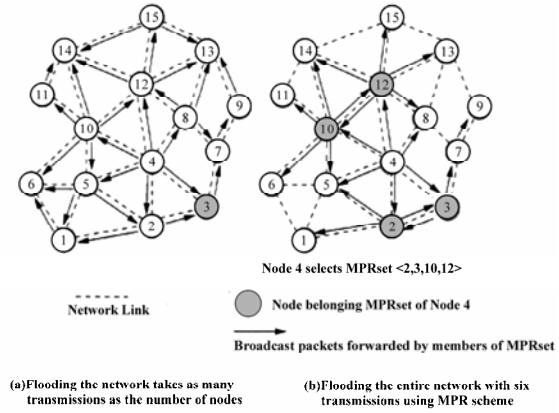



**Figure 1. Example of MPRs selection in OLSR protocol.**

The selection of nodes that constitute the *MPRset* significantly affects the performance of OLSR because a node calculates routes to all destinations only through the members of its *MPRset.* Every node periodically broadcasts its *MPRSelector* set to nodes in its immediate neighborhood. In order to decide on the membership of the nodes in the *MPRset,* a node periodically sends Hello messages that contain the list of neighbors with which the node has bidirectional links and the list of neighbors whose transmissions were received in the recent past but with whom bidirectional links have not yet been confirmed. The nodes that receive this Hello packet update their own two-hop topology tables. The selection of multipoint relays is also indicated in the Hello packet. A data structure called neighbor table is used to store the list of neighbors, the two-hop neighbors, and the status of neighbor nodes. The neighbor nodes can be in one of the three possible link status states, that is, unidirectional, bidirectional, and multipoint relay. In order to remove the stale entries from the neighbor table, every entry has an associated timeout value, which, when expired, removes the table entry. Similarly a sequence number is attached with the *MPRset* which gets incremented with every new *MPRset.*

The *MPRset* need not be optimal, and during initialization of the network it may be same as the neighbor set. The smaller the number of nodes in the *MPRset,* the higher the efficiency of protocol compared to link state routing. Every node periodically originates topology control (TC) packets that contain topology information with which the routing table is updated. These TC packets contain the *MPRSelector* set of every node and are flooded throughout the network using the multipoint relaying mechanism. Every node in the network receives several such TC packets from different nodes, and by using the information contained in the TC packets, the topology table is built. A TC message may be originated by a node earlier than its regular period if there is a

Copyright © 2009 SciRes. 

change in the *MPRSelector* set after the previous transmission and a minimal time has elapsed after that. An entry in the topology table contains a destination node which is the *MPRSelector* and a last-hop node to that destination, which is the node that originates the TC packet. Hence, the routing table maintains routes for all other nodes in the network.

### 2.2. MPR Selection Algorithm

The computation of the MPR set with minimal size is a NP-complet problem [16]. For this end, the standard MPR selection algorithm currently used in the OLSR protocol implementations is as follows:

For a node *x*, let *N(x)* be the neighborhood of *x*. *N(x)* is the set of nodes which are in the range of *x* and share with x a bidirectional link. We denote by *N2(x)* the two-neighborhood of *x*, i.e, the set of nodes which are neighbors of at least one node of *N(x)* but that do not belong to *N(x)* (see Figure 2).

Based on the above notations, the standard algorithm for MPR selection is defined as follows:

**1)** $U \leftarrow N^2(x)$

**2)** $MPR(x) \leftarrow \varnothing$

**3)** *while* $\exists v:\ v \in U \wedge \exists! w \in N(x) : v \in N(w)$ *do*

a) $U \leftarrow U - N(w)$

b) $MPR(x) \leftarrow MPR(x) \cup \{w\}$

**4)** *while*$(U \neq \varnothing)$ *do*

a) *choose* $w \in N(x)$ *such as* : $CRITERIA(w) = |N(w) \cap U| = \max(|w' \cap U| : w' \in N(x))$

b) $U \leftarrow U - N(w)$

c) $MPR(x) \leftarrow MPR(x) \cup \{w\}$

**5)** *return* $MPR(x)$

## 3. Proposed Node Mobility Degree

Each node in a mobile ad-hoc network can be found in four states with its neighbor nodes: the node moves and its neighbors are static, the node is static and its neighbors move, the node and its neighbors move, the node and its neighbors are static. Consequently, these four possible states result in a change of the link status of the node with its neighbors. So, as the nodes move in the mobile ad-hoc network, the link status changes in time.

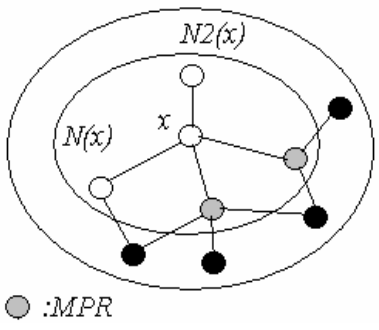



**Figure 2. Example of MRRset calculation.**

Based on this observation, we define a mobility measure representing the degree of node mobility in the network. This mobility measure has no unit and don't depend upon simulation artifacts such as mobility model parameters or movement patterns. Moreover its evaluation is done at discrete time intervals.

We define the mobility degree of a mobile node *i* at a time *t* by the following formula:

$$M_i^{\lambda}(t) = \lambda \frac{NodesOut(t)}{Nodes(t-\Delta t)} + (1-\lambda)\frac{NodesIn(t)}{Nodes(t)} \qquad \textbf{(1)}$$

where:

$NodesIn(t)$: The number of nodes that joined the communication range of $i$ during the interval $[t-\Delta t, t]$.

$NodesOut(t)$: The number of nodes that left the communication range of $i$ during the interval $[t-\Delta t, t]$.

$Nodes(t)$: The number of nodes in the communication range of $i$ at time t.

$\lambda$: The mobility coefficient between 0 and 1 defined in advance.

This node mobility degree is quantified locally and independently of the localization of a given node in the network. We represent this local and relative quantification by the change of the neighbors of each node. The node mobility degree at a given time *t* for node $i$ in the mobile ad-hoc network is defined as the change in its neighbors compared to the previous (state) at time $t-\Delta t$. Thus, mobile nodes that join and/or leave the neighbors of node $i$ will surely have an impact on the evaluation of its mobility degree. Moreover, we have chosen the mobility coefficient $\lambda$ between 0 and 1in order to have the node mobility degree at interval [0,1].

For illustration, let us take an example when node $i$ is on the state shown in (Figure 3(a)) with 10 neighbors, and during interval $\Delta t$, its neighbors will undergo the state changes shown in (Figure 3(b)): four nodes (with blue color) will leave the communication range, and two nodes (with red color) will join it. Consequently the node will be after $\Delta t$ (at time *t*) in the state (Figure 3(c)) with six changes. At the end of each time interval, the node will be able to make an evaluation of the change of its neighbors represented by this relative mobility, which is in our example equal to 13/40=32.5% (with $\lambda = 1/2$).

Each node in the mobile ad-hoc network can make an autonomous and automatic evaluation of its mobility at regular time intervals (this evaluation can be periodically done while exchanging the Hello messages). Mor- eover the calculation and recalculation of the node mobility is

Copyright © 2009 SciRes. 

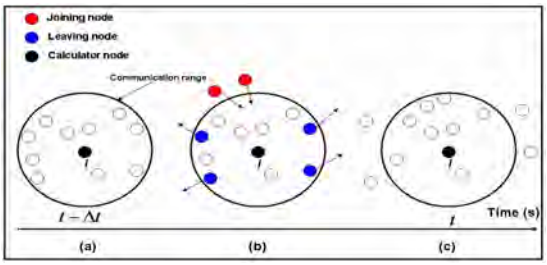



**Figure 3. Node mobility degree quantification.**

fast, and does not require enough consumption of resources (CPU and memory).

## 4. Our Improvement

Mobility is a crucial problem in MANETs, and until now, the majority of routing protocols have shown some weaknesses to face a high mobility in some parts of the network. Our objective consists in positively using the mobility, in order to adapt and improve the performance of the OLSR protocol.

### 4.1. Link Mobility Estimation

Some OLSR experiments [4,13] show that links must be more stable and less mobile to avoid fragile connections which involves data loss and frequent route changes. The OLSR protocol maintains constantly the shortest paths to reach all possible destinations in the network. So, it is more judicious to estimate the quality of links before adding them in the topological information that serves to calculate the best routes. The quality of a link can be estimated based on the power of the received signal. This information is provided by some wireless cards. If this information is not available, OLSR protocol estimates the link quality based on the number of control messages lost. A link failure can be detected using the timer expiry or by the link layer that informs upper layers of the failure with a neighbor node after reaching the maximal number of retries.

With an aim to estimate the quality of links in terms of mobility, we define the mobility of a link *L* between two nodes *A* and *B* as the average mobility of the involved nodes (see Figure 4), as showed in following equation:

$$M^{\lambda}_{L(A,B)} = \frac{M^{\lambda}_{A}(t) + M^{\lambda}_{B}(t)}{2} \quad \textbf{(2)}$$


**Figure 4. The link mobility of the link *L(A,B)* is (40%+ 50%)/2=45%.**

This evaluation of the link mobility alone is not significant because we can have a normal value of the link mobility with a high mobility value of one of the involved nodes. The dependence between the mobility of nodes composing a link (in the network core) at the time *t* can be seen as mobility dependence of link *L(A,B)* as follows:

$$P^{\lambda}_{L(A,B)}(t) = | M^{\lambda}_{A}(t) - M^{\lambda}_{B}(t) | \quad \textbf{(3)}$$

Therefore, a *reliable symmetric link* in terms of mobility can be seen as a link satisfying the two following conditions:

1) The *average mobility of the link L(i,j)* is lower than a threshold THRESHOLD_Link which depends on the characteristics of the wireless network (network density, network mobility, network scalability, network dimension, ...):

$$M^{\lambda}_{L(i,j)}(t) \le \text{THRESHOLD\_Link} \quad \textbf{(4)}$$

2) The *mobility dependence of link L(i,j)* is near to zero :

$$P^{\lambda}_{L(i,j)}(t) \to 0 \quad \textbf{(5)}$$

The choice of such a link satisfying these two conditions ensures the link to have a low mobility, with a strong dependence between the involved nodes.

### 4.2. Proposed Mobility Criterions

In this section, we propose three new criterions for the operation of MPRs selection. The first criteria is direct because it selects as MPRs set, neighbor nodes with less mobility (Figure 5 (a)). Precisely the node selected as MPR node is a node where its mobility is the smallest (Equation 6). The two other criterions are based on the estimation of links quality between neighbors at one-hop and the neighbors at two-hop (Figure 5 (b)). The quality of the link in terms of mobility is given by the two conditions cited in the previous sub-section. So, the new selection of the MPR set is a compromise between the number of links towards the nodes at two-hops and its reliability in terms of mobility. The selection of a

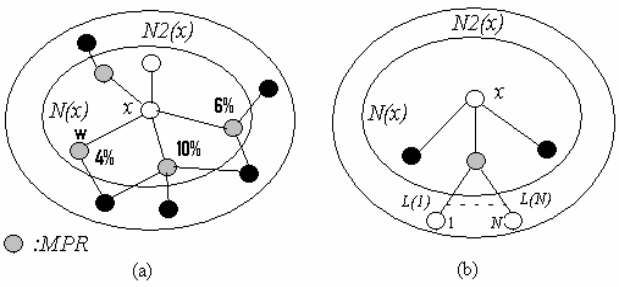



**Figure 5. Criterions evaluation.**

Copyright © 2009 SciRes. 

neighbor as a MPR node can be viewed as an operation of maximization of the selection criteria. The second criteria suggested is based on sum (Equation (7)) and the third is based on the product (Equation (8)). The principal advantage of these three criterions is the facility on calculation and doesn't require enough of resources in memory and CPU. Indeed, their evaluation is based on data base of neighbor nodes at one-hop and two-hop used by the OLSR protocol.

$$DIR-CRITERIA(w) = \min_{w \in N(x)} M_w^{\lambda}(t) \quad \textbf{(6)}$$

$$SUM-CRITERIA(w) = 1 - \frac{\sum_{i=1}^{N} M_{L(w,i)}^{\lambda}(t)}{N} \quad \textbf{(7)}$$

$$PRD-CRITERIA(w) = 1 - \prod_{i=1}^{N} M_{L(w,i)}^{\lambda}(t) \quad \textbf{(8)}$$

## 5. Metrics of Performance

In this paper we have considered the most important metrics for analyzing and evaluating performance of MANET routing protocols during simulation. These considered metrics are:

*Normalized Routing Overhead (NRL):* It represents the ratio of the control packets number propagated by every node in the network to the data packets number received by the destination nodes. This metric reflect the efficiency of the implemented routing protocols in the network.

*Packet Delivery Fraction (PDF):* This is a total number of delivered data packets divided by total number of data packets transmitted by all nodes. This performance metric will give us an idea of how well the protocol is performing in terms of packet delivery by using different traffic models.

*Average End-to-End delay (Avg-End-to-End):* This is the average time delay for data packets from the source node to the destination node. This metric is calculated by subtracting "time at which first packet was transmitted by source" from "time at which first data packet arrived to destination". This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC layer, propagation and transfer times.

## 6. Simulations and Results

In this section we have compared the performance of the original OLSR protocol based on the MPR selection standard algorithm, and the two modified OLSR protocols related to the direct and product criterions: DIR-OLSR and PRD-OLSR protocols. In this study we have eliminated the sum criteria for his hard cost in terms of MPRs nodes number [18,19].

### 6.1. Simulation Environment

For simulating the original OLSR protocol and the modified OLSR protocols related to our proposed criterions, we have used the OLSR protocol implementation [21] which runs in version 2.9 of Network Simulator NS2 [20] and uses the ad-hoc networking extensions provided by CMU, with a radio range of 250m.

We use a network consisting of 50 mobile nodes to simulate a high-density network. These nodes are randomly moved in an area of 1000m by 1000m according to the Random Waypoint (RWP) mobility model [22]. Moreover, to simulate a high dynamic environment (the worst case), we have consider the RWP mobility model with a pause time equal to 0. All simulations run for 300s.

A random distributed CBR (Constant Bit Rate) traffic model is used which allows every node in the network to be a potential traffic source and destination. The CBR packet size is fixed at 512 bytes. The application agent is sending at a rate of 10 packets per second whenever a connection is made. All peer to peer connections are started at times uniformly distributed between 5s and 290s seconds. The total number of connections and simulation time are 10 and 500s, respectively.

For each presented sample point, 50 random mobility scenarios are generated. The simulation results are thereafter statistically presented by the mean of the performance metrics. This reduces the chances that the observations are dominated by a certain scenario which favors one protocol over another. As we have interest in the case of high mobility (i.e. high link status and topology changes) we have reduced the HELLO interval and TC interval at 0.5s and 3s, respectively, for quick updates of the neighbors and topology data bases.

In particular, for the PRD-OLSR protocol related to the product criteria, we have choose THRESHOLD_Link= 0.05 as a threshold for evaluating the average mobility of links.

### 6.2. Results and Discussion

To show how the modified version of the OLSR protocol is more adapted to the link status and topology changes comparing to the original OLSR protocol, we have made there performance comparison based on the three performance metrics cited in Section 5. Moreover, with the supposed configuration cited above, we have run simulations in different mobility levels by varying maximum speed of nodes between 0m/s (no mobility) to 50m/s (very high mobility) in steps of 10m/s. For given the same importance of mobile nodes leaving and joining the

Copyright © 2009 SciRes. 

communication range at each node in the network we have choose the mobility coefficient equal to $\lambda$ =1/2.

According to the Figure 6, the original OLSR, PRD-OLSR and DIR-OLSR protocols ensure in the whole the same packet delivery fraction for all maximum speeds. Indeed, it can be seen that the number of packets dropped along the path is quite similar for all maximum speed being approximately 42% at worst. Moreover, the ratio is worse for a continuously changing network (i.e. high maximum speed) than for the static path conditions, because the number of link failures grows along with the mobility. However, it is interesting to notice that even with static topology conditions, sending nodes do not achieve 100% packet delivery but only 81%-83%. This clearly shows the impact of the network congestion and packet interference as the load on the network increases.

Figure 7 shows that PRD-OLSR protocol ensures a good enhancement in terms of delay comparing to the DIR-OLSR and original OLSR protocols, where have globally the same delay for all maximum speeds. Precisely, the original OLSR protocol delay is around 2.7 seconds with higher mobility rate (maximum speed equal to 50m/s) and decreases to almost 1.2 seconds with static topology conditions. For DIR-OLSR protocol the delay gets more than twice as large being almost 2.65 sec for high mobility and surprisingly increasing to over 1.2 seconds when the mobility is decreased. For the intermediate speed (from 10m/s to 40m/s) al lightweight difference between them is found. This allows us to conclude that original OLSR and DIR-OLSR protocols ensures approximatively the same delay.

Unlike to the protocols above (i.e. original OLSR and DIR-OLSR protocols), the PRD-OLSR protocol delay is about 2.6s (enhancement by 0.05s and 0.1s comparing to DIR-OLSR and original OLSR, respectively) with high mobility, increasing to almost 0.9s-1.1s (unlike the DIR-OLSR and original OLSR protocols that their minimum delay is found at 1.2s) with lower maximum

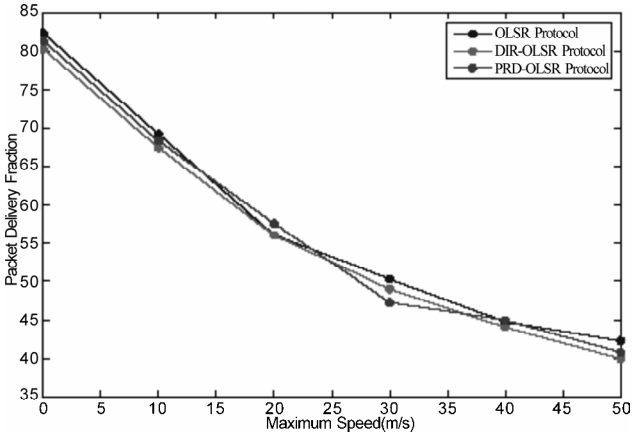



**Figure 6. Comparison of the three versions of the OLSR protocol in terms of packet delivery fraction.**

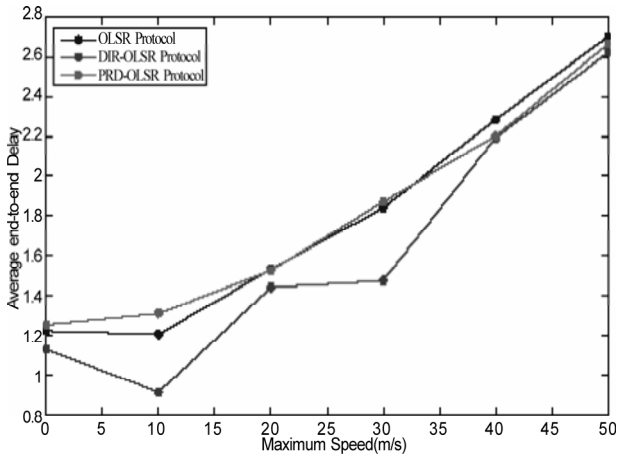



**Figure 7. Comparison of the three versions of the OLSR protocol in terms of Average end-to-end delay.**

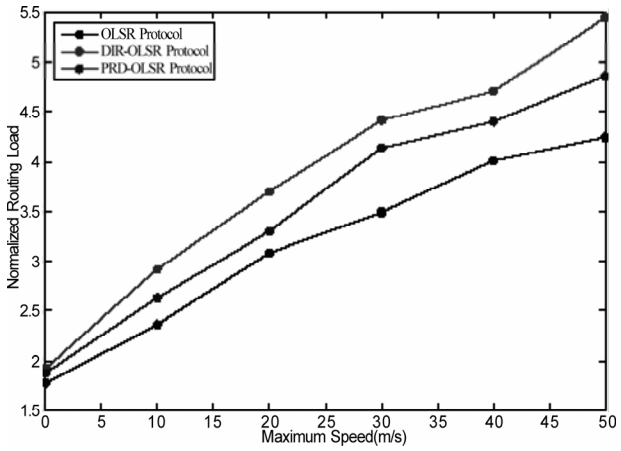



**Figure 8. Comparison of the three versions of the OLSR protocol in terms of normalized routing load.**

speed. Moreover, this enhancement is more shown for all intermediate maximum speeds and particularly for the two maximum speeds (10m/s and 30m/s). In short, we can say that the PRD-OLSR protocol is more adapted to all levels of mobility from 0m/s (no mobility) to 50m/s (very high mobility).

Figure 8 illustrates the normalized routing load (*NRL*) introduced into the network for the three versions of OLSR protocol, where the number of routing packets is normalized against sent data packets. A fairly stable normalized control message overhead would be a desirable property when considering the performance as it would indicate that the actual control overhead increases linearly with maximum speed of nodes due to the number of messages needed to establish and maintain connection. The OLSR protocol produces the lowest amount of *NRL* when compared to PRD-OLSR and DIR-OLSR protocols during all maximum speed values. Moreover, the PRD-OLSR protocol produces a lightweight routing

Copyright © 2009 SciRes. 

load comparing to the DIR-OLSR protocol that produces more routing load. In the worst case (at the maximum speed value equal to 50m/s), the *NRL* increases to 5.5% for DIR-OLSR protocol, 4.8% for PRD-OLSR and 4.25% for the original OLSR. Precisely, comparing to the original OLSR protocol, the PRD-OLSR and DIR-OLSR protocols produce 12.94% and 29.41% routing packets, respectively. This explains that our two proposed criterions request more routing packets to establish and maintain routes in the network.

## 8. Conclusions and Perspectives

This paper presents two versions of the original OLSR protocol, in the goal to adapt and enhance its performance to the dynamic nature of MANETs characterized by the link status and topology changes. These versions are based on a mobility degree that is quantified and evaluated in time by each mobile node in the network.

In the future works, we plan to continue this study by considering different configurations of MANETs for well understanding the behavior of each OLSR protocol version. Moreover, it is important to study the impact of the mobility coefficient $\lambda$ ($\lambda$ =1/2 in this work) by varying them into (0.00, 0.25, 0.75, 1.00). Finally, to implement an extension of the OLSR protocol supporting QoS, assuming that QoS requirements are expressed in terms of less mobility.

## 9. References

[1] T. Clausen and P. Jacquet ,"Optimized link state routing protocol (OLSR)," RFC 3626 Experimental, October 2003.

[2] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation," In Proceedings of the IEEE Symposium on Wireless Personal Mobile Communications, September 2001.

[3] F. Bai and A. Helmy, "A survey of mobility models in wireless ad hoc networks," Wireless Ad Hoc and Sensor Networks, Chapter 1, Kluwer Academic Publishers, pp. 1−29, June 2004.

[4] A. Laouiti, P. Muhlethaler, A. Najid, and E. Plakoo, "Simulation results of the OLSR routing protocol for wireless network," 1st Mediterranean Ad-Hoc Networks Workshop (Med-Hoc-Net), Sardegna, Italy, 2002.

[5] D. B. Johnson, D. A. Maltz, and Y. C. Hu. "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Internet-Draft, Draft-Ietf-Manet-Dsr-10.txt, Work in Progress, July 2004.

[6] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, Experimental, July 2003.

[7] Y. Ge, T. Kunz, and L. Lamont, "Proactive QoS routing in ad-hoc networks," The 2nd International Conference on Ad-Hoc Networks and Wireless, Montreal, Canada, October 2003.

[8] T. Plesse, J. Lecomte, C. Adjih, M. Badel, *et al.*, "OLSR performance measurement in a military mobile ad-hoc network," Ad-Hoc Networks Journal Special Issue on Data Communication and Topology Control in Ad-Hoc Networks, October 2004.

[9] T. Clausen, P. Jacquet, and L. Viennot, "Investigating the impact of partial topology in proactive MANET routing protocols," The 5th International Symposium on Wireless Personal Multimedia Communications, 2002.

[10] A. Tonnesen, "Implementing and extending the optimized link State Routing Protocol," Master Thesis, Department of Informatics, University of Oslo, August 2004.

[11] OOLSR, "Implementation of the OLSR, optimized link state routing protocol," Hipercom Project, http://hipercom.inria.fr/oolsr/, November 2004.

[12] V. Park and M. Corson, "Temporally-ordered routing algorithm (TORA): Version 1 functional specification," Internet-Draft, IETF, Draft-Ietf-Manet-Tora-Spec-04.txt, July 2001.

[13] A. Laouiti and C. Adjih, "Measures des performances du protocole OLSR," IEEE SETIT2003 Tunisia, March 2003.

[14] J. Haerri, C. Bonnet, and F. Filali, "OLSR and MPR: Mutual dependencies and performances," In Proceedings of Med-Hoc Net 2005, June 2005.

[15] A. Busson, N. Mitton, and E. Fleury, "Analysis of the multipoint relays selection in OLSR and implications," In Proceedings of Med-Hoc Net 2005, June 2005.

[16] Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in Proceedings of the Hawaii International Conference on System Sciences (HICSS'02), Big Island, Hawaii, January 2002.

[17] S. Obilisetty, A. Jasti, and R. Pendse, "Link stability based enhancements to OLSR (LS-OLSR)," Vehicular Technology Conference, IEEE 62nd, pp. 28−25, September, 2005.

[18] N. Enneya, A. Baayer, and M. Elkoutbi, "New criterion for MPR selection in OLSR protocol," in Procceding of IASTED Wireless and Optical Communications, Monteral, Canada, pp. 416−421, May 30−June 1, 2007

[19] N. Enneya, K. Oudidi, and M. Elkoutbi, "New mobility metrics for MPRs selection in the OLSR protocol," 9th African Conference on Research in Computer Science and Applied Mathematics (CARI'08), Rabat, Morocco, October 27−30, 2008.

[20] The VINT Project, "The network simulator ns-2," http://www.isi.edu/nsnam/ns/, Page accessed on January 2008.

Copyright © 2009 SciRes. 

[21] F. J. Ros, "UM-OLSR version 8.8.0," University of Murcia, Spain, http://masimum.dif.um.es/?Software: UM-OLSR, January 2008.

[22] B. J. David and A. M. Daviv, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing. Kluwer Academic Publishers, pp. 153−181, 1996.

[23] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," IEEE Pers. Communication, Vol. 8, No. 1, pp 16−28, 2001.

[24] B. J. Kawak, N. O. Song, and L. E. Miller, "A standard measure of mobility for evaluating mobile ad hoc network performance," IEICE TRANSACTION COMMUNICATION, Vol. E86-B, No. 11, pp. 3236−3243, November 2003.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 400-406
doi:10.4236/ijcns.2009.25045 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Network Delay Model for Overlay Network Application

**Tian JIN, Haiyan JIN**
*School of Electronics and Information Engineering, BeiHang University, Beijing, China*
*Department of Enterprise Management, North China Electric Power University, Beijing, China*
*Email*: *jintian@buaa.edu.cn, jhy*99@263.*net*
*Received May* 6, 2009; *revised* June 11, 2009; *accepted July* 8, 2009

## ABSTRACT

This paper presents a method to model network delay for overlay network application. The network topology measurement technology and network AS information is used to build up model of network delay via AS and geographic distance. Based on global Internet measurement result, we calculated the parameters of the model. Furthermore, the model verification is done by comparing on AS-MMI protocol and HMTP protocol.

**Keywords:** Network Model, Autonomous System, Overlay Network

## 1. Introduction

The network delay measurement in large scale overlay system become impossible due to the Internet expansion. So, in this paper we introduce a method of modeling network delay with AS and geographic distance. This method use network topology information to aid us know more information about the connectivity of Internet, and reduce the time and range of network delay measurement [1,2]. Contrast with network bandwidth, delay is more stable. Thus, the system can share a much more stable result of network topology measurement.

First, we will show how to measure the Internet topology. And then, we will show how to modeling the delay of network by the topology and geographic information. At last, we will demonstrate the protocol performance by using the model and ping result. That result shows our model of Internet can reflect network delay quite well.

## 2. Related Work

By knowing the location of a client host, an application, such as a Web service, could send the user location-based targeted information, classify users based on location, and improve the performance of overlay application.

Previous work on the measurement-based geolocation of Internet hosts uses the positions of reference hosts [3, 4], called landmarks, with well-known geographic location as the possible location estimates for the target host. These might limit the accuracy of the resulting location estimation, because the closest reference host may still be far from the target.

Some applications such as GeoTrack, GeoPing and GeoCluster have developed to map IP-to-location information, but none of them has detailed model. In this paper, we will show how the model could be measured.

## 3. Topology Measurement

Some researches report that there are 36888 routers and 42269 links in 10 major AS. And number of AS over Internet has exceeded 18000.

Due to the bandwidth inside an AS is much more than inter-connection of AS, we can use the AS relationship to describe the network connectivity. In multimedia communication, network delay always happens between AS inter-connections. So, the network delay can be measured by the AS count of packet passed. According to Oregon University OIX project, we can gather the router information all over the world, and summarize the Internet connectivity relationship.

Though research in topology measurement is very popular, but most of them do not take geography into consideration [5]. With the development of network, the inter-connections between AS will become more and more short. The delay of network will mostly depend on

Copyright © 2009 SciRes. 

**Table 1. Major AS network router and links.**

| AS Num | Name | Place | Router | Link |
|---|---|---|---|---|
| 1221 | Telstra | (Australia) | 2,796 | 3,000 |
| 1239 | Sprintlink | (US) | 8,355 | 9,500 |
| 1755 | Ebone | (Europe) | 596 | 500 |
| 2914 | Verio | (US) | 7,336 | 6,800 |
| 3257 | Tiscali | (Europe) | 865 | 700 |
| 3356 | Level3 | (US) | 3,446 | 6,700 |
| 3967 | Exodus | (US) | 900 | 1,100 |
| 4755 | VSNL | (India) | 121 | 69 |
| 6461 | Abovenet | (US) | 2,259 | 1,400 |
| 7018 | AT&T | (US) | 10,214 | 12,500 |

geographic delay. So, the network delay can be modeled by AS and geographic distance. We will introduce the technique of deducing network delay by analyzing route information and generating network topology. The key issues contains, how to analyze AS information; how to generate AS connectivity and how to model geographic distance of AS.

The network topology measurement take four stages: The first is gathering Internet route table, analyzing BGP route and mapping between IP and AS. The second stage is analyzing BGP route and AS path. We can deduce AS connectivity at this stage and calculate shortest connectivity path of AS. The third stage is finding AS geographic information according to AS registry information. The last stage is calculating AS communication delay by AS and geographic information.

### 3.1. Gathering Route Information

BGP is an external gateway routing protocol of TCP/IP. It is designed for solving large scale network route problem. The BGP route is synchronized all over Internet, so it can reflect the topology of current Internet.

The path information of two Autonomous Systems is recorded in route table of all core routers. We can get the AS connectivity by analyzing these information. If the information is gathering from difference routers over the world, the topology will be more accurate.

Oregon University's OIX project provided summary of some core router's BGP route table. We can extract mapping between IP and AS from that table. For instance, following mapping can be obtained by previous route Table 2:

**Table 2. IP-AS mapping from route table.**

| IP Address | AS Num |
|---|---|
| 6.1.0.0/16 | 668 |
| 6.2.0.0/22 | 668 |
| 6.3.0.0/18 | 668 |
| 6.4.0.0/16 | 668 |

**Table 3. AS peers listed in route table.**

| AS Num | Peer AS Num |
|---|---|
| 4538 | 9407 |
| 9407 | 7660 |
| 7660 | 11537 |
| 11537 | 668 |
| 19782 | 11537 |

### 3.2. AS Information Analyzing

In BGP route table, AS path is the route path at AS level. So, analyzing AS path information, we can deduce the connectivity of AS. For instance, we can summarize the following connectivity of AS from the route Table 3:

According to BGP route data of Oregon University at year 2004, there are 18431 AS and 39886 links in Internet. The data briefly described how the Internet is connected at that time.

### 3.3. Analyzing AS Connectivity

We can calculate the shortest path between two AS by using shortest tree algorithm. The route protocol (such as OSPF) has taken the shortest path into consideration. The result will reflect the theoretical minimal distance between two AS, which is similar as actual distance [6,7]. We use the number of AS passed in communication (also called as AS length) to represent the distance of AS. The BGP route in Internet might always change, but most BGP route change does not interfere with AS length. The length of AS path is stable in most time.

We can deduce AS length by previous route example:

The distance of AS can be calculated via following algorithm:

Define $Path(u)$ is, the set of AS directly connect with AS u.

Define $\mathrm{Len}(u,v)$ is, distance of AS u and AS v.

If "u v" or "v u" exist in AS path of BGP route Table 4,

**Table 4. AS distance according to route table.**

| AS Num | AS Num | Dis | AS Num | AS Num | Dis |
|---|---|---|---|---|---|
| 4538 | 9407 | 1 | 9407 | 19782 | 3 |
| 4538 | 7660 | 2 | 7660 | 11537 | 1 |
| 4538 | 11537 | 3 | 7660 | 668 | 2 |
| 4538 | 668 | 4 | 7660 | 19782 | 2 |
| 4538 | 19782 | 4 | 11537 | 668 | 1 |
| 9407 | 7660 | 1 | 11537 | 19782 | 2 |
| 9407 | 11537 | 2 | 668 | 19782 | 2 |
| 9407 | 668 | 3 | | | |

Copyright © 2009 SciRes. 

then $u \in Path(v)$.

If $u \in Path(v)$, then $\mathrm{Len}(u,v)=1$. If $u \notin Path(v)$, then $\mathrm{Len}(u,v)=0$.

If $w \in Path(v)$, and $\mathrm{Len}(u,v) \neq 0$, and $\mathrm{Len}(u,w)=0$, then $\mathrm{Len}(u,w)=\mathrm{Len}(u,v)+1$.

If $w \in Path(v)$, and $\mathrm{Len}(u,v) \neq 0$, and $\mathrm{Len}(u,w) \neq 0$, then $\mathrm{Len}(u,w)=\min(\mathrm{Len}(u,v)+1, Len(u,w))$.

According to BGP route data from Oregon University's OIX project at year 2004, the average distance of AS is 3.765305, AS distance of 18431 AS is shown in Table 5:

### 3.4. Using Geographic Information

With the development of Internet and speed, the geographic ratio in communication delay will be increased. So, we must take the geographic delay into account [8]. For simple calculation, we can get the AS number by the IP address, and then get country information by AS registry. And the geographic distance can be calculated by the longitude and latitude information of that country. For mote accurate calculation, we can use city information rather than country information for calculation.

The information of IP Address and city, AS number and country information can be retrieval from whois server. The latitude and longitude information can be retrieval from NetGeo and other projects [6,7]. With the earth's radius and following formula, we can calculate the theoretical distance of two nodes:

DISTANCE=R*ARCOS[SIN(A)SIN(C)
+COS(A)COS(C)COS(B-D)];

R is earth's mean radius: 6371km.

(A, B) is latitude and longitude of node 1.

(C, D) is latitude and longitude of node 2.

The network delay consists of geographic delay and AS communication delay. So, we can calculate the backbone link delay and geographic delay by topology measurement result. The following formula shows the network delay model:

RTT＝T(N) + P*D

The P is related with current network condition.

D is the geographic distance between two nodes.

T(N) is the delay of AS length N between two nodes.

Based on measurement result, value of P and T is P=20us/km, T(2)=10ms, T(3)=55ms, T(4)=78ms, T(5)= 92ms. The Internet is in a changing state, so previous parameter will also change with Internet's development. That value only reflects current Internet measurement result from Chapter 4.

**Table 5. Summary of AS distance.**

| Dis | Count | Ratio | Dis | Count | Ratio |
|---|---|---|---|---|---|
| 1 | 79772 | 0.0235% | 8 | 133926 | 0.0394% |
| 2 | 18585690 | 5.4715% | 9 | 11467 | 0.0034% |
| 3 | 121181350 | 35.6748% | 10 | 792 | 0.0002% |
| 4 | 135241350 | 39.8139% | 11 | 65 | 0.0000% |
| 5 | 52057273 | 15.3252% | 12 | 8 | 0.0000% |
| 6 | 10960601 | 0.32267% | 13 | 1 | 0.0000% |
| 7 | 1431035 | 0.4213% | | | |

## 4. Network Delay Analyze

The topology measurement is related with reality network, so we can not setup the topology measurement test with simulation. We use PlanetLab [9] node as source of reality measurement, and use ping (or extended tcpping [10]) to measure reality network delay. The comparison will confirm the relationship between AS length and network delay.

Due to the ping firewall in Internet, some ping measurement will not reach to some host. To get the better measurement result, we use TCP instead of ICMP for node delay test [1,2,11], and use connection confirm time instead of RTT time to represent network delay.

We use ScriptRoute's packet data service to measure the network delay between node installed ScriptRoute. The TCP port 3355 is used for tcpping measurement instead of ICMP ping. The node from 83 AS and 129 nodes joined the test, they are distributed as Figure 1:

We select 14 nodes' result, and make detail analyses for network topology model in Table 6 and Table 7.

According to AS distance calculation algorithm at previous chapter, we can get the AS distance of the nodes in Table 8 and Table 9.

After we summarize AS distance, geographic distance and RTT information from previous table, we can calculate network delay data from 81 nodes. Among the result, most of AS distance is 2 to 4, which is 88.9% of all data. The distribution of AS distance is shown at Table 10.

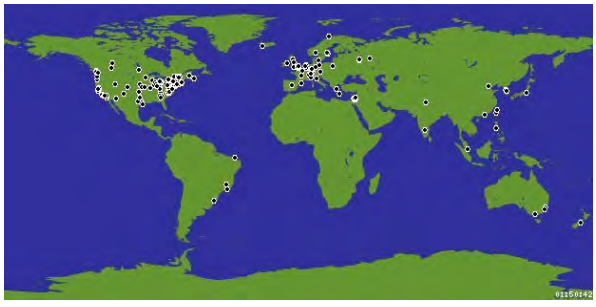


**Figure 1. Topology measurement node distribution.**

Copyright © 2009 SciRes. 

**Table 6. Topology measurement node list.**

| | IP | AS | Area | Lat. | Long. |
|---|---|---|---|---|---|
| **S1** | 128.208.4.155 | 73 | us | 47.65 | -122.31 |
| **S2** | 129.97.75.240 | 549 | ca | 45.35 | -72.52 |
| **S3** | 142.103.2.2 | 271 | ca | 49.26 | -123.23 |
| **S4** | 212.192.241.155 | 2848 | ru | 55.65 | 37.5 |
| **S5** | 165.132.126.58 | 4665 | kr | 37.53 | 127 |
| **S6** | 140.109.17.180 | 9264 | tw | 25.02 | 121.37 |
| **S7** | 130.161.40.154 | 1128 | nl | 52.02 | 4.35 |
| **S8** | 132.72.23.10 | 378 | il | 31.5 | 34.75 |
| **S9** | 198.32.154.195 | 11537 | us | 40.72 | -73.99 |
| **S10** | 195.37.16.101 | 680 | de | 48.58 | 13.47 |
| **S11** | 140.192.37.134 | 20130 | us | 41.88 | -87.63 |
| **S12** | 206.117.37.5 | 226 | us | 33.98 | -118.46 |
| **S13** | 128.83.143.153 | 18 | us | 30.28 | -97.74 |
| **S14** | 202.141.62.35 | 23731 | in | 29.85 | 77.9 |

**Table 7. The geographic distance (km).**

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| **S1** | 0 | 3754 | 192 | 8375 | 8322 | 9740 | 7831 |
| **S2** | 3754 | 0 | 3776 | 7022 | 10594 | 12065 | 5424 |
| **S3** | 192 | 3776 | 0 | 8213 | 8156 | 9581 | 7714 |
| **S4** | 8375 | 7022 | 8213 | 0 | 6621 | 7359 | 2190 |
| **S5** | 8322 | 10594 | 8156 | 6621 | 0 | 1490 | 8615 |
| **S6** | 9740 | 12065 | 9581 | 7359 | 1490 | 0 | 9497 |
| **S7** | 7831 | 5424 | 7714 | 2190 | 8615 | 9497 | 0 |
| **S8** | 10920 | 8765 | 10774 | 2694 | 8121 | 8289 | 3356 |
| **S9** | 3864 | 528 | 3914 | 7511 | 11056 | 12535 | 5840 |
| **S10** | 8497 | 6175 | 8372 | 1807 | 8415 | 9158 | 751 |
| **S11** | 2787 | 1274 | 2861 | 8003 | 10512 | 12001 | 6595 |
| **S12** | 1554 | 4069 | 1744 | 9790 | 9578 | 10910 | 8958 |
| **S13** | 2848 | 2757 | 3006 | 9559 | 11156 | 12587 | 8158 |
| **S14** | 11163 | 11135 | 10972 | 4271 | 4570 | 4299 | 6347 |

| | R8 | R9 | R10 | R11 | R12 | R13 | R14 |
|---|---|---|---|---|---|---|---|
| **S1** | 10920 | 3864 | 8497 | 2787 | 1554 | 2848 | 11163 |
| **S2** | 8765 | 528 | 6175 | 1274 | 4069 | 2757 | 11135 |
| **S3** | 10774 | 3914 | 8372 | 2861 | 1744 | 3006 | 10972 |
| **S4** | 2694 | 7511 | 1807 | 8003 | 9790 | 9559 | 4271 |
| **S5** | 8121 | 11056 | 8415 | 10512 | 9578 | 11156 | 4570 |
| **S6** | 8289 | 12535 | 9158 | 12001 | 10910 | 12587 | 4299 |
| **S7** | 3356 | 5840 | 751 | 6595 | 8958 | 8158 | 6347 |
| **S8** | 0 | 9156 | 2607 | 9951 | 12212 | 11511 | 4104 |
| **S9** | 9156 | 0 | 6587 | 1145 | 3958 | 2432 | 11651 |
| **S10** | 2607 | 6587 | 0 | 7346 | 9673 | 8910 | 5740 |
| **S11** | 9951 | 1145 | 7346 | 0 | 2825 | 1575 | 11902 |
| **S12** | 12212 | 3958 | 9673 | 2825 | 0 | 1990 | 12712 |
| **S13** | 11511 | 2432 | 8910 | 1575 | 1990 | 0 | 13313 |
| **S14** | 4104 | 11651 | 5740 | 11902 | 12712 | 13313 | 0 |

Distance calculated by longitude and latitude (km)

**Table 8. Topology experiment AS distance.**

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| **S1** | 0 | 4 | 3 | 4 | 4 | 3 | 4 |
| **S2** | 4 | 0 | 3 | 5 | 5 | 3 | 5 |
| **S3** | 3 | 3 | 0 | 5 | 4 | 2 | 5 |
| **S4** | 4 | 5 | 5 | 0 | 4 | 3 | 4 |
| **S5** | 4 | 5 | 4 | 4 | 0 | 3 | 4 |
| **S6** | 3 | 3 | 2 | 3 | 3 | 0 | 3 |
| **S7** | 4 | 5 | 5 | 4 | 4 | 3 | 0 |
| **S8** | 4 | 5 | 3 | 3 | 4 | 3 | 3 |
| **S9** | 2 | 4 | 2 | 2 | 3 | 1 | 2 |
| **S10** | 3 | 4 | 2 | 3 | 3 | 2 | 3 |
| **S11** | 4 | 4 | 3 | 4 | 4 | 2 | 3 |
| **S12** | 3 | 3 | 2 | 3 | 3 | 2 | 3 |
| **S13** | 4 | 5 | 4 | 5 | 4 | 3 | 5 |
| **S14** | 4 | 4 | 3 | 4 | 4 | 3 | 4 |

| | R8 | R9 | R10 | R11 | R12 | R13 | R14 |
|---|---|---|---|---|---|---|---|
| **S1** | 4 | 2 | 3 | 4 | 3 | 4 | 4 |
| **S2** | 5 | 4 | 4 | 4 | 3 | 5 | 4 |
| **S3** | 3 | 2 | 2 | 3 | 2 | 4 | 3 |
| **S4** | 3 | 2 | 3 | 4 | 3 | 5 | 4 |
| **S5** | 4 | 3 | 3 | 4 | 3 | 4 | 3 |
| **S6** | 3 | 1 | 2 | 2 | 2 | 3 | 2 |
| **S7** | 3 | 2 | 3 | 3 | 3 | 5 | 4 |
| **S8** | 0 | 2 | 2 | 4 | 3 | 4 | 5 |
| **S9** | 2 | 0 | 2 | 2 | 2 | 2 | 3 |
| **S10** | 2 | 2 | 0 | 3 | 3 | 4 | 3 |
| **S11** | 4 | 2 | 3 | 0 | 3 | 4 | 3 |
| **S12** | 3 | 2 | 3 | 3 | 0 | 3 | 3 |
| **S13** | 4 | 2 | 4 | 4 | 3 | 0 | 5 |
| **S14** | 5 | 3 | 3 | 3 | 3 | 5 | 0 |

Copyright © 2009 SciRes. 

**Table 9. RTT time (ms) measured by tcpping.**

| R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|
| 0 | 71.349 | 30.971 | 199.615 | 148.035 | 246.747 | 158.807 |
| 71.349 | 0 | 66.913 | 139.948 | 197.421 | 310.892 | E |
| 30.976 | 66.913 | 0 | 194.833 | 162.055 | 255.681 | 140.498 |
| 198.063 | 139.948 | 191.213 | 0 | 309.407 | 316.383 | 65.828 |
| 152.266 | 197.421 | 145.103 | 309.407 | 0 | 120.777 | 284.129 |
| 246.775 | 310.892 | 255.564 | 347.639 | 135.715 | 0 | 300.238 |
| 158.807 | E | 140.498 | 65.828 | 284.129 | 300.238 | 0 |
| 223.482 | 117.406 | 213.658 | 105.605 | 253.891 | 270.334 | 70.366 |
| 56.205 | E | 86.534 | 169.715 | E | 225.518 | E |
| 184.487 | 117.094 | 171.863 | 63.994 | 324.602 | 331.154 | 28.366 |
| 46.925 | 12.234 | 31.679 | 138.64 | 248.451 | 186.586 | 102.759 |
| 26.13 | 91.096 | 51.684 | 194.72 | 173.54 | 157.166 | 169.848 |
| 60.169 | E | 90.399 | 166.525 | 189.145 | 229.427 | E |
| 743.486 | 770.898 | 887.795 | 912.26 | E | 938.597 | 909.354 |
| **R8** | **R9** | **R10** | **R11** | **R12** | **R13** | **R14** |
| 223.543 | 56.205 | 184.542 | 50.386 | 26.056 | 60.169 | 742.947 |
| 117.406 | E | 117.094 | 12.234 | 91.096 | E | 770.898 |
| 202.403 | 86.534 | 172.014 | 49.707 | 56.315 | 90.399 | 917.678 |
| 104.881 | 169.715 | 65.103 | 133.095 | 185.149 | 166.525 | 922.6 |
| 368.949 | E | 319.806 | 178.883 | 176.751 | 189.145 | 896.655 |
| 372.332 | 225.518 | 331.148 | 197.726 | 196.177 | 229.427 | 938.417 |
| 70.366 | E | 28.366 | 102.759 | 169.848 | E | 909.354 |
| 0 | 195.149 | 83.075 | 136.026 | 218.953 | 150.46 | 921.899 |
| 195.149 | 0 | E | 29.099 | 32.648 | E | E |
| 83.398 | E | 0 | 135.035 | 187.613 | 157.074 | 912.963 |
| 174.036 | 29.099 | 126.183 | 0 | 55.733 | 28.134 | 728.829 |
| 164.449 | 32.648 | 165.351 | 45.122 | 0 | 35.5 | 684.068 |
| 150.46 | E | 157.074 | 28.134 | 35.5 | 0 | 776.616 |
| 936.419 | E | 907.139 | 803.945 | 751.519 | 776.616 | 0 |

**Table 10. Topology experiment result summary.**

| AS Distance | Count | Ratio | Avg. Geographic Distance (km) | Avg. Network Delay (ms) |
|---|---|---|---|---|
| 1 | 1 | 1.23% | 12534 | 225.518 |
| 2 | 13 | 16.05% | 6455 | 138.98 |
| 3 | 33 | 40.74% | 6799 | 206.03 |
| 4 | 26 | 32.10% | 7058 | 257.67 |
| 5 | 8 | 9.88% | 8660 | 333.25 |

Copyright © 2009 SciRes. 

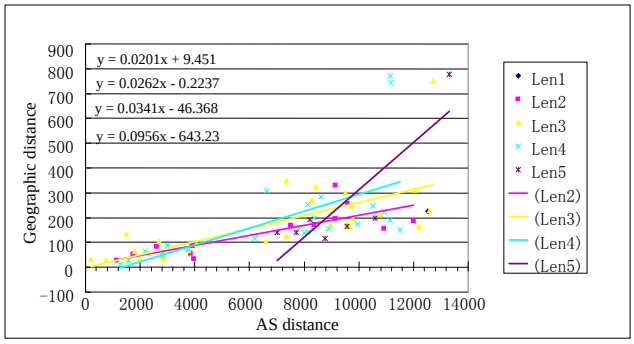



**Figure 2. Network delay model.**

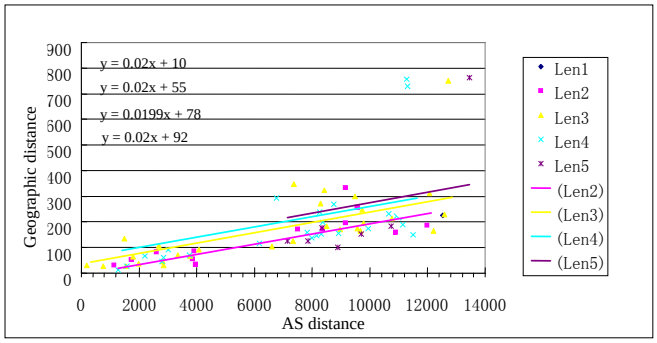



**Figure 3. Normalized network delay model.**

We can find from summary that the average geographic distance and network delay are dramatically increased when AS distance is increased. And the ratio of AS distance is similar as global value, in which ratio of AS distance from 1~5 is (0.0235%, 5.4715%, 35.6748%, 39.8139%, 15.3252%). From the result, we can draw the conclusion that there is relationship among AS distance, geographic distance and network delay.

Then we divided the measurement result into several groups based on AS distance. We can get the model of geographic distance and network delay in different AS distance by using linear regression method.

Due to the network delay cause by geographic distance is direct proportion with geographic distance, the different Linear Regression line should have same slope. So, we slightly adjust all the Linear Regression line's slope into same value -0.02. The procedure is called as normalization. The model after normalization is:

From the result, the network delay can be calculated via following formula:

$$RTT = T(N) + P*D$$

Inside, P is the parameter reflect geographic distance, P=20us/km.

D is the geographic distance of two nodes.

T(N) is the network delay caused by AS distance N, where T(2)=10ms, T(3)=55ms, T(4)=78ms, T(5)=92ms.

The result shows network delay seemed to be random, but it is related with geographic distance and AS distance. Commonly speaking, average network RTT delay will increase 0.02ms when geographic distance increase 1 kilometer; average network RTT delay will increase 11~45ms when AS distance increase 1.

## 5. Model Verification

We use the model of network delay in overlay network protocol design to test it [12]. We design an AS-MMI protocol based on MMI protocol for large scale multimedia communication. The gateway selection algorithm of AS-MMI protocol will use the model. By contrast, we use HMTP [13,14] for comparison, which will use partly measured RTT delay from previous table. The two protocols consider Shortest Path Tree (SPT) algorithm as best solution.

The performance experiment takes SPT algorithm for reference. And there are four factors to evaluate the time and quality of spanning tree.

Tree Cost means the cost of a tree is the sum of delays on the tree's links. Tree cost is a convenient metric to measure total network resource consumption of a tree.

Tree Delay means the delay from one member to another along the tree. The ratio between tree delay and unicast shortest path delay is delay ratio.

Tree Time means the total time used to build the span-

Copyright © 2009 SciRes. 

**Table 11. Spanning tree result.**

| | *HMTP* | *AS-MMI* |
|---|---|---|
| Tree Cost | 1074.317 | 1009.897 |
| Tree Cost Ratio | 1.67 | 1.57 |
| Tree Delay | 563.677 | 712.624 |
| Tree Delay Ratio | 1.18 | 1.49 |
| Tree Time | 2817.704 | 1009.897 |
| Hit Node | 4 | 5 |
| Hit Ratio | 33.3% | 41.7% |

ning tree. The time can reflect the effectiveness of protocol.

Hit Ratio: If the delay of the node and its parent node is no more than 10% of delay in SPT algorithm. We considerate this node as a Hit node, which means the node's delay is quite low. The ratio of hit node reflects how many nodes delay is acceptable.

Verification result is shown in Table 11:

We can find from previous table, AS-MMI protocol use much more short time in build up spanning tree than HMTP protocol. In spanning tree cost and hit ratio, the result of two protocols is similar. But the AS-MMI protocol will cause a little higher tree delay. That means the network model used by AS-MMI protocol can reflect most node's network delay, and tree build speed is much faster. But the AS distance is only theoretical distance which might be different as actual distance, some nodes' network delay might have errors. These nodes' delay caused tree delay increase dramatically, but does not have much effect interfere with tree cost and hit ratio.

Overall, the network model used by AS-MMI protocol can provide a fast method to check the network delay between different nodes. And the result shows it is accurate for most nodes of AS-MMI protocol.

## 6. Acknowledgment

Many thanks to the network and research support from State Key Laboratory of Software Development Environment, BeiHang University.

## 7. References

[1] M. Jain and C. Dovrolis, "End to end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," Technical Report, University of Delaware, February 2002.

[2] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," In Proceedings of Passive and Active Measurements (PAM) Workshop, March 2002.

[3] A. Ziviani, S. Fdida, J. F. de Rezende, and O. C. M. B. Duarte, "Improving the accuracy of measurement-based geographic location of Internet hosts," Computer Networks, Vol. 47, No. 4, pp. 503–523, March 2005.

[4] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in Proceedings ACM SIGCOMM, San Diego, CA, pp. 173–185, August 2001.

[5] M. Kwon and S. Fahmy, "Topology-aware overlay networks for group communication," In ACM NOSSDAV, 2002.

[6] P. Francis, S. Jamin, V. Paxon, L. Zhang, D. Gryniewicz, and Y. Jin, "An architecture for a global internet host distance estimation service," Proceedings 18th IEEE Infocom'99, 1999.

[7] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "Idmaps: A global internet host distance estimation service," IEEE/ACM Transactions on Networking, October 2001.

[8] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," In ACM SIGCOMM Internet Measurement Workshop, 2002.

[9] Planet Lab Project, http://www.planet-lab.org/.

[10] ScriptRoute Project, http://www.scriptroute.org/.

[11] C. Dovrolis, P. Ramanathan, and D. Moore. "What do packet dispersion techniques measure?" Proceedings of IEEE INFOCOM, April 2001.

[12] S. Shi, "Design of overlay networks for internet multicast," Ph. D. dissertation, Washington University, August 2002.

[13] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: A framework delivering multicast to end users," Proceedings of IEEE INFOCOM '02, 2002.

[14] B. Zhang, W. Wang, S. Jamin, D. Massey, and L. Zhang, "Universal IP multicast delivery," Computer Networks: The International Journal of Computer and Telecommunications Networking, pp. 781−806, April 2006.

[15] T. Cicic, "Networklevel multicast deployment and recovery," Ph. D. dissertation, October 2001.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 407-421
doi:10.4236/ijcns.2009.25046 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# UWB-Based Localization in Wireless Sensor Networks

**Di WU[1, 2], Lichun BAO[1], Renfa LI[2]**
[1]*Donald Bren School of ICS, University of California, Irvine, USA*
[2]*School of Computer and Communication, Hunan University, Changsha, China*
*Email*: *dwu3@ics.uci.edu, lbao@ics.uci.edu, scc_lrf@hnu.cn*
*Received March* 11, 2009; *revised April* 21, 2009; *accepted May* 30, 2009

## ABSTRACT

Localization has many important applications in wireless sensor networks, such as object searching and tracking, remote navigation, location based routing etc. The distance measurements have been based on a variety of technologies, such as acoustic, infrared, and UWB (ultra-wide band) media for localization purposes. In this paper, we propose UWB-based communication protocols for distance estimation and location calculation, namely a new UWB coding method, called U-BOTH (UWB based on Orthogonal Variable Spreading Factor and Time Hopping), an ALOHA-type channel access method and a message exchange protocol to collect location information. U-BOTH is based on IEEE 802.15.4a that was designed for WPANs (wireless personal area networks) using the UWB technology. We place our system in coal mine environments, and derive the corresponding UWB path loss model in order to apply the maximum likelihood estimation (MLE) method to compute the distances to the reference sensors using the RSSI information, and to estimate the coordinate of the moving sensor using least squares (LS) method. The performance of the system is validated using theoretic analysis and simulations. Results show that U-BOTH transmission technique can effectively reduce the bit error rate under the path loss model, and the corresponding ranging and localization algorithms can accurately compute moving object locations in coal mine environments.

**Keywords:** Orthogonal Variable Spreading Factor (Ovsf), Time Hopping (Th), Ultra-Wide Band (Uwb), Localization, Ranging

## 1. Introduction

Large-scale economic wireless sensor networks (WSNs) become increasing attractive to environmental monitoring, control and interaction applications. Object tracking and localization is one of the key challenges for these applications [1]. Various solutions have been proposed based on two ranging techniques: 1) time of arrival (TOA) [2], such as GPS, 2) the path loss model based on radio RSSI signal strength [3] or acoustic signal strength [4] attenuation in relation to the signal propagation distance. Sometimes, range-free techniques are also applied to estimation locations, such as hop count or centroid methods [5].

However, most of these localization methods require generic signal propagation and network formation assumptions. In this paper, we place our localization method in coal mine environments for monitoring and tracking human and vehicle locations using multiple reference points installed in the WSNs. This approach is especially valid given the practical value of the localization system in helping people in the frequent emergency situations and reducing the high costs of coal mine operations.

Coal mine environments present extremely harsh conditions for wireless communications. First, the power of the transmitter underground must be reduced to the lowest level to avoid sparkling gas explosions. Secondly, signal propagations are especially prone to multipath effects. Third, wireless networks are more dynamic than surface networks due to signal attenuation, movements etc. Last but not the least, wireless sensor network in

* This work was sponsored in parts by the National Natural Science Foundation of China under Grant No. 60673061 and the Raytheon Company under Grant No. RC-42621.

Copyright © 2009 SciRes. 

coal mines is a multiple users system, and the MUI (multiple users' interference) has dramatic impacts on the precision of a localization system. Coding is an important method to depress MUI.

UWB (ultra-wide band) transmission and coding technologies provide an ideal solution to the coal mine environment. On one hand, UWB systems can provide high bandwidth data transmissions; on the other hand, UWB exhibits excellent characteristics to reduce co-channel interference. IEEE 802.15.4a is the *de facto* standards to provide low power long distant low data rate service for real-time communication and precise ranging and localization applications [6,7].

There are many UWB localization algorithms proposed in the past [8−11]. Wang *et al*. demonstrated the use of UWB in coal mines to realize short-distance high-rate applications such as video monitoring, as well as localization and monitoring [12].

Of the different UWB transmission techniques, Impulse Radio Ultra-wideband (IR-UWB) provides a desirable platform to enable efficient and precise localization solutions in coal mines environments [13]. Different coding algorithms for IR-UWB communication systems have been proposed so far, such as DS-UWB (Direct Sequence UWB) and TH-UWB (Time Hopping UWB) [14]. However, none was shown to guarantee high quality localization. The simple DS-UWB cannot even meet the localization requirements when the multipath and multi-user interference exist. In this paper, we apply the Orthogonal Variable Spread Factor (OVSF) coding algorithm in IR-UWB networks to solve the multi-user interference problem.

Other than TOA (time-of-arrival), TDOA (time difference of arrival) and AOA (angle of arrival) based ranging techniques, ranging based on the path loss model is an intuitive method, especially in low-cost WSNs. The path loss model defines the signal propagation characteristics, and determines the received signal strength. Therefore, given the received signal strength (RSSI), we can estimate the distance between the receiving node and other reference points using computational methods, rather than expensive hardware [15]. Several channel models were proposed to evaluate UWB systems in different propagation environments in the IEEE 802.15.4a. However, these models relied on insufficient measurements and fixed parameters, and cannot reflect the real channel characteristics. In [16], a statistical path loss model was established for channels in the residential areas based on over 300,000 frequency response measurements. This approach shows good agreement with measured data, but requires a highly complex modeling and simulation procedures, as mentioned in IEEE 802.15.4a. Li *et al*. analyzed the propagation mode of UWB signal in coal mine, and proposed a free-space propagation alike model based on the existing residential indoor model [17]. So far, channel path loss modeling in coal mine environment remains a complex task. In this paper, we will provide a more practical and accurate coal mine model using UWB medium based on IEEE 802.15.4a.

Once the range or distance information is available between the mobile target and the reference points in the WSNs, the location of the mobile target is fairly easy to derive. Trilateration is a common approach. Savarese *et al*. presented a trilateration algorithm based on least squares (LS) method in large-scale WSNs [18]. We apply similar method, but because the number of reference points involved in the localization algorithm could be limited, the LS method is adapted to run multiple iterations in order to reduce the power consumption of the reference points, and to provide accurate location coordinates.

The rest of the paper is organized as follows. Section 2 describes the basic assumptions of the localization system, and some of the symbols used in this paper. Section 3 presents a new IR-UWB coding method, called U-BOTH (UWB based on Orthogonal Variable Spreading Factor and Time Hopping), and provides the signal processing model of UWB in coal mine environments. Section 4 specified a WSN communication protocol in order to collect reference point location information. According to the path loss model and the RSSI information gathered by the mobile target, Section 5 and Section 6 present the ranging and localization algorithms using the maximum likelihood estimation (MLE) and the least squares methods, respectively. Section 7 evaluates the system using simulations. Section 8 concludes the paper.

## 2. Assumptions and Notation

Although we focus on coal mine wireless sensor network (WSN) deployment, the results can be easily adapted to other deployment scenarios. The key difference between various WSN deployments is the signal propagation characteristics, reflected in the path loss model used for range calculations. In each of these deployment scenarios, we assume that a number of reference nodes are deployed the network, and have already acquired their exact location coordinate through other means, such as initial location calibrations. The task in our localization computation is to derive the position coordinate of a mobile target object by running the localization algorithms on the target. We do not elaborate on the application of the coordinate information in this paper.

Figure 1 illustrate such a WSN in which a target node, denoted by triangle, moves across the network. The tar

Copyright © 2009 SciRes. 

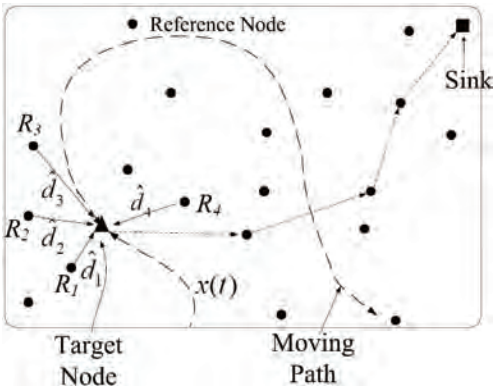



**Figure 1. Wireless sensor network localization using UWB.**

get node collects the coordinate information of reference nodes $R_1$, $R_2$, $R_3$ and $R_4$ denoted by dots, and their corresponding signal strength, by which to calculate the ranges between itself and the four reference nodes, $\hat{d}_1$, $\hat{d}_2$, $\hat{d}_3$ and $\hat{d}_4$. Afterward, the target node derives its own coordinate, and sends to a sink, denoted by the square in Figure 1.

In order to enable communication between the target and reference nodes in the network, we assume that each and every node in the WSN is able to communicate through U-BOTH, proposed in this paper.

For convenience, the notation used in this paper is summarized in Table 1.

**Table 1. Notation and meaning.**

| Notation | Meaning |
|---|---|
| $T_f$ | The frame time. |
| $T_c$ | The chip time. |
| $T_b$ | The bit time. |
| $N_s$ | The number of pulses for every bit. |
| $N_c$ | The number of chips for every frame. |
| $d_j^n$ | The OVSF code of transmitter $n$. |
| $SF$ | The spreading factor of OVSF code. |
| $N_s$ | The period of OVSF code. |
| $E_{TX}^n$ | The transmission energy of transmitter $n$. |
| $E_{RX}^n$ | The received energy of transmitter $n$. |
| $p_0(t)$ | The energy normalized pulse waveform. |
| $c_j^n$ | The time-hopping code with period $N_s$. |
| $a_{\lfloor j/N_s \rfloor}^n$ | The indication of information bit $b$. |
| $r_\mu(t)$ | The input useful signal of the receiver. |
| $r_{mui}(t)$ | The input multiple users interference signal of the receiver. |
| $n(t)$ | The input additive white Gaussian noise of the receiver. |
| $m(t)$ | The correlation temple of the receiver. |
| $Z_\mu$ | The output useful signal of the receiver. |
| $Z_{mui}$ | The output multiple users interference of the receiver. |
| $Z_n$ | The output additive white Gaussian noise of the receiver. |
| $N_0$ | The noise spectral density. |
| $\tau$ | The delay of the other transmitter's interfering pulse. |
| $\mu_x$ | The mean value of variable $x$. |
| $\sigma_x$ | The standard deviation of a random variable $x$. |
| $erfc(x)$ | The complementary error function of value $x$. |
| $\mathrm{Pr}_b$ | The bit error rate (BER). |

Copyright © 2009 SciRes. 

## 3. Physical Layer Model

### 3.1. UWB Signal Spreading and Modulation

First of all to achieve accurate localization in wireless communication, we need a reliable physical layer communication technique that reduces bit error rate (BER), while mitigating the multi-users-interference (MUI) and Gaussian noise interference. Our physical layer is a UWB system based on time-hopping (TH) signal transmission as well as OVSF (orthogonal variable spread factor) for spreading out the symbols.

OVSF (Orthogonal Variable Spread Factor) was extensively used in CDMA systems to provide variable spreading codes [19]. Shorter OVSF code lengths are usually optimized for short distance, high data rate transmission in less crowed environments due to its smaller spreading factor. On the other hand, time hopping (TH) is one of many signal modulation methods used by UWB. We will apply the time-hopping pulse position modulation (TH-PPM) algorithm to encode UWB pulse streams, and OVSF direct sequence to spread the user data bit stream, called U-BOTH (UWB modulation Based on OVSF and Time Hopping).

Figure 2 illustrates the utilization of time hopping (TH) pulse position modulation and OVSF spreading to encode a single bit in the user data stream. First, U-BOTH sends each bit in the bit time, denoted by $T_b$. Then it modulates the bit 1 using a TH code, 12110021, in which each digit denotes a chip slot position within a frame time, $T_f$, to send a broadband radio pulse. The number of pulses is denoted by $N_s$. Therefore, each bit duration is $T_b = T_f \times N_s$. Each chip slot lasts for $T_c$, sufficient to send a short UWB pulse signal.

After the initial pulse position modulation using UWB signals, the pulse sequence is again applied with OVSF code so that the phases are shifted by $\pi$ to provide orthogonality between multiple users. The length of the OVSF code is called the spread factor *SF*, which is equal to $N_s$.

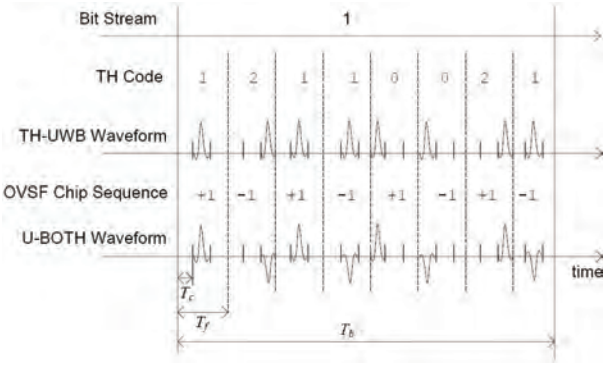



**Figure 2. U-BOTH: Interference resistant UWB modulation using time hopping and OVSF.**

In our system, the TH code is a pseudo-random sequence generated from foreknown seeds, such as node IDs. And the OVSF codes are selected from a well- defined set of orthogonal spreading codes.

To formally analyze the system in this paper, we represent the transmitted signal by the nth transmitter in Equation (1):

$$s^{(n)}(t) = \sum_{j=-\infty}^{+\infty} d_j^n a_{\lfloor j/N_s \rfloor}^n \sqrt{E_{TX}^n}\, p_0\left(t - jT_f - c_j^n T_c\right), \quad \textbf{(1)}$$

in which, $d_j^n = \pm 1$ is the OVSF code with the period $N_s$, $E_{TX}^n$ is the energy of the *n*-th transmitter, $p_0(t)$ is the energy normalized pulse waveform, $c_j^n \in [0, N_c - 1]$ is the TH code with period $N_s$ and $a_{\lfloor j/N_s \rfloor}^n$ indicates the data stream bit. If the data bit is 1, $a_{\lfloor j/N_s \rfloor}^n = +1$. Otherwise, $a_{\lfloor j/N_s \rfloor}^n = -1$.

At the receiver side, the received signal consists of three source of information:

$$r(t) = r_u(t) + r_{mui}(t) + n(t),$$

in which, $r_u(t)$ is the desired user signal, $r_{mui}(t)$ is co-channel interference from multiple users, and $n(t)$ is the additive white Gaussian noise (AWGN).

Denote the pulse energy of the *n*-th transmitter as $E_{RX}^n$. Without loss of generality, we assume that the first user's transmission is the desired signal at the receiver for simplicity, then Equation (2) provides the desired signal function at the receiver:

$$r_u(t) = \sum_{j=-\infty}^{+\infty} d_j^1 a_{\lfloor j/N_s \rfloor}^1 \sqrt{E_{RX}^1}\, p_0\left(t - jT_f - c_j^1 T_c\right). \quad \textbf{(2)}$$

We define the correlation template of the receiver:

$$m(t) = \sum_{j=iN_s}^{(i+1)N_s - 1} d_j^1 p_0\left(t - jT_f - c_j^1 T_c\right); i \in (-\infty, +\infty). \quad \textbf{(3)}$$

### 3.2. Single User System Analysis

As the first step, we assume that the channel is the AWGN multipath-free channel, and that the transmitter and the receiver are synchronized. In a single user signal processing system, the input of the receiver has two parts: $r_u(t)$ and $n(t)$, and the output of the receiver in time interval [*0, $T_b$*] is represented by:

$$Z = Z_\mu + Z_n = \int_0^{T_b} \left(r_u(t) + n(t)\right) m(t)\, dt. \quad \textbf{(4)}$$

Copyright © 2009 SciRes. 

In Equation (4), the useful output signal is:

$$Z_\mu = \sum_{j=0}^{N_s-1} \int_{jT_f + c_j^1 Tc}^{jT_f + c_j^1 Tc + Tc} d_j^1 d_j^1 a_{\lfloor j/N_s \rfloor}^1 \sqrt{E_{RX}^1}\, \omega(t)\, dt ,$$

where $\omega(t) = p_0\left(t - jT_f - c_j^1 T_c\right) p_0\left(t - jT_f - c_j^1 T_c\right)$.

Because $d_j^1 d_j^1 = 1$, $p_0(t)$ is the energy normalized pulse waveform, we have

$$\begin{aligned} Z_\mu &= \sum_{j=0}^{N_s-1} \int_0^{T_c} a_{\lfloor j/N_s \rfloor}^1 \sqrt{E_{RX}^1}\, p_0(t)\, p_0(t)\, dt \\ &= N_s a_{\lfloor j/N_s \rfloor}^1 \sqrt{E_{RX}^1} \int_0^{T_c} p_0(t)\, p_0(t)\, dt \\ &= a_{\lfloor j/N_s \rfloor}^1 N_s \sqrt{E_{RX}^1} . \end{aligned}$$

In Equation (4), the output noise signal is:

$$Zn = \sum_{j=0}^{N_s-1} \int_0^{T_c} d_j^1 p_0(t) n(t)\, dt = \sum_{j=0}^{N_s-1} d_j^1 n_j ,$$

where $n_j$ is Gaussian random variable with mean 0 and variance $N_0/2$. Because $d_j^1$ is not a random variable, the variance of $Z_n$ is:

$$D(Z_n) = D\left(\sum_{j=0}^{N_s-1} d_j^1 n_j\right) = N_s \frac{N_0}{2},$$

$$Zn \sim N\left(0, N_0 N_s / 2\right).$$

Suppose that the statistical probabilities of data bit $b$ = 0 and $b$ = 1 are equal, we obtain the BER (bit error rate) of the single user system in AWGN channel as follows:

$$\mathrm{Pr}_b = \frac{1}{2} P(Z > 0 \mid b = 0) + \frac{1}{2} P(Z < 0 \mid b = 1) = P(Z > 0 \mid b = 0).$$

Because $a_{\lfloor j/N_s \rfloor}^n = -1$ if b = 0, then the useful output is $Zu = a_{\lfloor j/N_s \rfloor}^1 N_s \sqrt{E_{RX}^1} = -N_s \sqrt{E_{RX}^1}$. Using Equation (4), the BER become:

$$\begin{aligned} \mathrm{Pr}_b &= P(Z > 0 \mid b = 0) = P\left(-N_s \sqrt{E_{RX}^1} + Z_n > 0\right) \\ &= P\left(Z_n > N_s \sqrt{E_{RX}^1}\right). \end{aligned}$$

It can be rewritten by complementary error function $erfc(x)$ as follow:

$$\mathrm{Pr}_b = \frac{1}{2} erfc\left(\sqrt{\frac{N_s E_{RX}^1}{N_0}}\right).$$

where $erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp\left(-t^2\right) dt$.

Because U-BOTH is a rate variable system using OVSF, we analyze the relation between BER and the bit rate. Suppose the system's OVSF code is a code tree of 6 layers [20], and the spreading factor is 2, 4, 8, 16, 32, 64, respectively. Further suppose the basic rate of our system is $R_0$, then the corresponding bit rate of U-BOTH is $R_b = iR_0$ ($i$ = 32, 16, 8, 4, 2, 1, respectively).

Denote the bit rate as $R_b$, where $R_b = iR_0$, $i$ = 1, 2…32, we can get the relation between BER and the bit rate:

$$\mathrm{Pr}_b = \frac{1}{2} erfc\left(\sqrt{\frac{SF \cdot E_{RX}^1}{N_0}}\right) = \frac{1}{2} erfc\left(\sqrt{\frac{64 R_0 \cdot E_{RX}^1}{R_b N_0}}\right), \quad \textbf{(5)}$$

Equation (5) shows that the BER decrease when the spreading factor *SF* increases or when the bit rate decreases. Therefore, we can adjust *SF* to adapt different environments with various noise levels while maintaining the same bandwidth of the signal. This is the main reason we adjust OVSF codes in our system.

### 3.3. Multi-User Interference Analysis

In multi-user communication system, the received signal includes multi-user interference $Z_{mui}$ and noises. The $Zu + Zn$ part is the same as Equation (4), but the multi-user interference $Z_{mui}$ is additional. Because the phase and delay _ of interfering pulses is random as shown in Figure 3, we have to compute the interference's variance.

Suppose that $\tau^n$ is uniformly distributed over $[0, T_f)$, then the interference variance of the desired signal, i.e. the signal from the 1st user, caused by transmitter $n$ is [14]:

$$\sigma_{bit}^2 = \frac{N_s}{T_f} \int_0^{T_f} \left(\sqrt{E_{RX}^n} \int_0^{Tc} d_j^1 d_i^n p_0\left(t - \tau^n\right) p_0(t)\, dt\right)^2 d\tau^n .$$

Therefore, the total interference variance $\sigma_{mui}^2$ from all other transmitters is:

$$\sum_{n=2}^{N_\mu} \left( \frac{N_s E_{RX}^n}{T_f} \int_0^{T_f} \left( \int_0^{T_c} d_j^1 d_i^n p_0\left(t - \tau^n\right) p_0(t)\, dt \right)^2 d\tau^n \right).$$

Because the delay $\tau$ for all transmitters has the same distribution, we get the following formula:

$$\sigma_{mui}^2 = \frac{N_s}{T_f} \sum_{n=2}^{N_\mu} E_{RX}^n \left( \int_0^{T_f} \left( \int_0^{Tc} d_j^1 d_i^n p_0\left(t - \tau^n\right) p_0(t) dt \right)^2 d\tau^n \right)$$

Copyright © 2009 SciRes. 

$$= \sigma_M^2 \frac{N_s}{T_f} \sum_{n=2}^{N_\mu} E_{RX}^n \ ,$$

in which,

$$\sigma_M^2 = \int_0^{T_f} \left( \int_0^{T_c} d_j^1 d_i^n p_0\left(t - \tau^n\right) p_0\left(t\right) dt \right)^2 d\tau = \int_0^{T_f} R^2\left(\tau\right) d\tau \ .$$

According to [14], and noticing that $R_b = \frac{1}{N_s T_f}$ and $N_s = SF = \frac{64R_0}{R_b}$, Equation (6) gives the BER in multi-user interference environments.

$$\Pr_b = \frac{1}{2} erfc \left( \sqrt{\frac{1}{2} \left( \left( \frac{2N_s E_{RX}^1}{N_0} \right)^{-1} + \left( \frac{N_s E_{RX}^1}{\sigma_M^2 \frac{1}{T_f} \sum_{n=2}^{Nu} E_{RX}^n} \right)^{-1} \right)^{-1}} \right)$$

$$= \frac{1}{2} erfc \left( \sqrt{\frac{1}{2} \left( \left( \frac{128R_0 E_{RX}^1}{R_b N_0} \right)^{-1} + \left( \frac{E_{RX}^1}{\sigma_M^2 R_b \sum_{n=2}^{N_\mu} E_{RX}^n} \right)^{-1} \right)^{-1}} \right). \quad \textbf{(6)}$$

## 4. Network Protocol Operations

### 4.1. Protocol Operation

Our localization algorithms depend on a two-step process—the first step is for the target node to acquire the coordinate and signal strength information from reference nodes in the network using U-BOTH based communication protocols, and the second step is for the target node to calculate the distances to the reference nodes, and infer its own coordinate.

In order to get the necessary coordinate information from adjacent reference nodes, the following protocol steps are taken:

1) The mobile node broadcasts a location request message.
2) All the reference nodes send back packet with their own coordinates. During each short sampling period, a moving node can receive tens of responses from each of the reference nodes.

In wireless sensor networks, code assignments are categorized into transmitter-oriented, receiver-oriented or a per-link-oriented code assignment schemes (also known as TOCA, ROCA and POCA, respectively) [21, 22]. Depending on the ways of assigning the OVSF-TH codes and encoding the MAC data frames for transmissions, we propose two different ways to implement multiple access protocols using U-BOTH.

a) *ROCA-Based Protocol Operations*: The first approach is based on the receiver-oriented code assignment (ROCA), in which case the data packet transmissions are encoded using the unique OVSF-TH code assigned to the receiver. Beside ROCA, there is a common OVSF-TH code for bootstrapping and coordination purposes.

In ROCA scheme, when a target node needs to find out its coordinate, it sends a location request message using the common OVSF-TH code to the reference nodes. The request message includes the request command, and the receiver's OVSF-TH code. Upon receiving the request message, each reference nodes sends back a response message using the receiver's OVSF-TH code using a random backoff mechanism.

The response message includes each reference node's identification information, and their coordinates.

b) *TOCA-Based Protocol Operations*: The second approach is based on transmitter-oriented code assignment

(TOCA), in which case each packet transmission is encoded using two OVSF-TH codes — one is a common OVSF-TH code to encode the common physical layer frame header, and the other transmitter-specific code is to encode the physical layer frame payload. The frame head includes the transmitter-oriented OVSF-TH code for encoding the frame payload.

Because the physical layer headers are sent on a common OVSF-TH code, the physical layer header transmissions resemble those of ALOHA networks with regard to packet collision. Because the headers are usually short, the collision probability is low.

On the other hand, because the data frame payload is transmitted on unique OVSF-TH codes, the interference between the payload and other frame headers and payloads is dramatically reduced.

In both ROCA- and TOCA-based systems, packets from the reference nodes can be lost. However, this does not affect the overall performance of our localization algorithms because they tolerate such losses.

### 4.2. Location Calculation Algorithms

We make use of our OVSF-TH-UWB system and provide a UWB sensor localization network for mining applications to monitor environment and mineworker. We suppose moving nodes and other monitored nodes are the target nodes and every reference node knows its location. Suppose target node has a UWB RFID Tag equipped with transmitter and receiver to assist distributed localization with reference nodes. Data sink collect all the

Copyright © 2009 SciRes. 

real-time localization information and send out to the monitoring center outside the mining area, as shown in Figure 1.

There are two steps in order to estimate the coordinate of the mobile node:

a) *Ranging*: Ranging is to estimate the approximate distance between the target node and the reference nodes. Target node estimates the distance from it to each reference node according to the RSSI values, using maximum likelihood estimation.

In this paper, we take ROCA scheme during ranging, target node broadcasts range-initiate (RI) packets for range estimates, and neighbor reference nodes reply with range-response (RR) packets, which include their coordinate and signal strength. To collect ranging and location information as much as possible in a short time, we take a dual-channel mechanism joint with common code and receiver code provided by OVSF-TH code. The receiver OVSF-TH code is generated by the unique MAC ID of receiver. Suppose target node *i* broadcast a RI packet via common channel code $C_0$ at $t = t_1$, and begin to listen the RR packets sent to its unique OVSF-TH code. This process initiate a window of time from $t = t_1$ to $t = t_1 + Tw$. The window-length $Tw$ is much larger than RR duration $T_{RR}$, which allows multiple RR gathered within window duration from adjacent reference nodes.

Following are the ranging and localization algorithm:

1) Target node *i* broadcasts RI packet at $t = t_1$ on common code $C_0$. This RI includes node *i*'s MAC ID and OVSFTH code Ci. When there is no interference, the RI arrives at a reference node *j*.
2) Reference node *j* delays $n\Delta_T$ after receiving the RI, and then reply a RR packet transmitted on the OVSF-TH code *Ci*. The RR packet includes node *j*'s MAC ID, transmitted signal strength and node *i*'s MAC ID. Where n is a random positive integer belong to [1, *K*], *K* is the average number of reference nodes in the transmitted range of target node. The delay $n\Delta_T$ is used to avoid interference between RR packets.
3) If $t_1 < t < t_1 + Tw$, node *j* continue to delay $n\Delta_T$ and then transmit RR packet via *Ci*.
4) Other reference nodes in the transmitted range of node *i* send RR packets by the same ways as node *j* in step 2 and 3.
5) If $t = t_1 + Tw$, node *i* stop receiving RR packets.]
6) When $t = t_1 + Tw$, node *i* received multiple RR packets from different reference nodes and several RR packets from the same reference node. It uses all the recorded information of *PLi* to estimates the distances to different reference nodes by MLE based RSSI ranging indicated in Equation (14) in Section 5.

b) *Localization*: Coordinate calculation, which is to determine the coordinate of the mobile node according to the coordinate information of the reference nodes and the corresponding distance from the target node to the reference nodes. Using the coordinates of the reference nodes and the estimated distance information, calculate the coordinate of the target node. It is implemented as follows:

1) When node *i* estimated all the distances to different reference nodes in its transmitted range, it applies these ranging results to Equation (15) in Section 6 and then computes its location based on the least squares algorithm.
2) When node *i* estimated its coordinate, it broadcasts ACK packet via its common channel code *C0*, informing finish of ranging. Its newest location is aware to neighbor nodes by the ACK and will arrive at the data sink through the localization routing.

According to above procedures of network protocol operations, after getting the reference coordinates and the respective signal strength information, a target node calculates its coordinate in two steps — ranging and localization. In order to fully take advantage of U-BOTH physical model in Section 3, we also design specific ranging algorithm and localization algorithm for U-BOTH in following sections.

## 5. Ranging Algorithm

Ranging is to estimate the approximate distance between the target node and the reference nodes. We use the maximum likelihood estimation for such calculations. First of all, we need to establish the path loss model of the UWB channel in order to inversely derive the distance information from received signal qualities.

### 5.1. The Path Loss Model

It is well-known that the path loss model can be expressed by the log-distance path loss law in many indoor or outdoor environments, as shown by Equation (7).

$$PL(d) = \left( PL_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right)\right) + S; \quad d \geq d_0, \qquad \textbf{(7)}$$

in which

- *d0* is the reference distance (e.g. 1 meter in UWB medium),
- *PL0* means the path loss in dB at *d0*,
- *d* is the distance between the transmitter (Tx) and receiver (Rx),
- $\gamma$ refers to the path loss exponent which depends on channel and environment,

Copyright © 2009 SciRes. 

- *S* is the log-normal shadow fading in dB. Usually, *S* is a Gaussian-distributed random variable with zero mean and standard deviation $\sigma$ .

The tunnel's environment in coal mine can be regarded as a special type of indoor environments, considering various kinds of concrete environmental factors in coal mine [17]. Thus, we adopt the UWB path loss model in coal mine based on residential indoor propagation model.

According to the residential indoor models, the values of

$\gamma$ , $\sigma$ and S in Equation (7) are specific in each propagation environment, and could be treated as random variables [16].

Accordingly, the UWB path loss is commonly modeled as:

$$PL\left(\overline{d}\right)\Big|_{dB} = \left(PL_0 + 10\mu_\gamma \log_{10} d\right) + \left(10n_1\sigma_\gamma \log_{10} d + n_2\mu_\sigma + n_2n_3\sigma_\sigma\right), \quad \textbf{(8)}$$

in which $\mu_\sigma$ is the mean value of shadow fading's standard deviation $\sigma_S$ .

The probability density function (*pdf*) is:

$$p\left(PL\middle|d\right) = \frac{e^{-\frac{\left(PL-\left(PL_0+10\mu_\gamma \log_{10} d\right)\right)^2}{2\left(100\sigma_\gamma{}^2\left(\log_{10} d\right)^2+\mu_\sigma{}^2+\sigma_\sigma{}^2\right)}}}{\sqrt{2\pi\left(100\sigma_\gamma{}^2\left(\log_{10} d\right)^2+\mu_\sigma{}^2+\sigma_\sigma{}^2\right)}}. \quad \textbf{(9)}$$

However, an unknown distance variable d appears in both the denominator and the exponent's denominator in Equation (9), so it is hard to develop further statistical analysis. In addition, the path loss exponent $\gamma$ in the aforementioned model is a random variable, and requires sufficient measurements on the spot in various residential environments before effectively being applied in generic scenarios. Especially, the standard deviation

$\sigma$ of the log-normal shadow fading *S* usually changes from locations to locations. Even if at the same location, it may change because of the time-varying channel. Thus, we need practical method to estimate *S*, especially for moving targets.

Therefore, in order to apply above path loss model, IEEE

802.15.4a Task Group provided Channel Model 1-9 by taking limited real measurements to determine the values of $\gamma$ , $\sigma$ and other variables in different situations. When deploying real UWB networks, people could approximately choose the corresponding channel model with the parameters specified in IEEE 802.15.4a.

We propose a UWB coal mine propagation model based on many other modeling methods for the application of ranging and localization. In this model, the mean value of the path loss exponent $\gamma$ is given for different tunnel environments, and the log-normal shadow fading *S* is represented through a random variable as follows:

$$S = n_1\sigma ,$$

$$\sigma = \mu_\sigma + n_2\sigma_\sigma .$$

Then, the indoor UWB path loss could be expressed as:

$$PL\left(\overline{d}\right)\Big|_{dB} = \left(PL_0 + 10\gamma \log_{10} d\right) + \left(n_1\mu_\sigma + n_1n_2\sigma_\sigma\right), \quad \textbf{(10)}$$

according to Equation (7).

In Equation (10), $n_1$ and $n_2$ are zero-mean Gaussian variables of unit standard deviation, $n_1, n_2 \sim N[0,1]$ . (PL0+10 log10 d) is the median path loss, and $\left(n_1\mu_\sigma + n_1n_2\sigma_\sigma\right)$ represents the random variation about the median path loss. According to the "$3\sigma$ principle" in Gaussian distribution, the probability of a Gaussian variable lying in the range $(\mu-3\sigma, \mu+3\sigma)$ is 99.73%, even though the range of Gaussian random variable is $(-\infty, +\infty)$ . That is, 99.73% value of variables $n_1$ and $n_2$ is within the range of (-3, +3). Furthermore, it dramatically simplifies computation if we use truncated Gaussian distributions for $n_1$ and $n_2$ so as to keep $\gamma$ and $\sigma$ from taking on impractical values. According to [16], we confine $n_1, n_2 \in [-2, +2]$ , and $n_1n_2 \in [-4, +4]$ .

In aforementioned model, $\left(n_1\mu_\sigma + n_1n_2\sigma_\sigma\right)$ is not exactly Gaussian because $n_1n_2$ is not Gaussian variable. However, the product is very small with respect to whole expression of path loss. Thus, the path loss *PL(d)* is approximately a Gaussian-distributed random variable with:

$$n_1\mu_\sigma + n_1n_2\sigma_\sigma \sim N\left(0, \mu_\sigma{}^2 + \sigma_\sigma{}^2\right),$$

$$PL\left(d\right) \sim N\left(PL_0 + 10\gamma \log_{10} d, \mu_\sigma{}^2 + \sigma_\sigma{}^2\right).$$

The probability density function (*pdf*) of the path loss *PL(d)* is:

$$p(PL\middle|d) = \frac{e^{-\frac{\left(PL-\left(PL_0+10\gamma \log_{10} d\right)\right)^2}{2\left(\mu_\sigma{}^2+\sigma_\sigma{}^2\right)}}}{\sqrt{2\pi\left(\mu_\sigma{}^2+\sigma_\sigma{}^2\right)}}. \quad \textbf{(11)}$$

Compared with the model in [16], our UWB coal mine propagation model given by Equation (11) is more convenient to carry out parameter estimation and statistic analysis because it simplifies the statistics and *PDF*. What

Copyright © 2009 SciRes. 

is more, when considering the random influence of the log-normal shadow fading, this model is generic than current models in IEEE 802.15.4a.

### 5.2. Ranging Algorithm Based on Maximum Likelihood Estimation

The distance between the transmitter Tx and the receiver Rx in Equation (10) can be calculated by the general ranging method between two nodes using the RSSI information:

$$\hat{d} = 10^{\frac{PL(d) - PL_0 - n_1\mu_\sigma - n_1 n_2 \sigma_\sigma}{10\gamma}}$$

Receiver computes the distance between the transmitter Tx and the receiver Rx using random values $n_1$ and $n_2$ in the truncated range. This method takes into account the influence of real log-normal shadow fading on ranging and decreases the ranging error compared to the models in IEEE 802.15.4a.

However, the random variables $n_1$ and $n_2$ selected by the transmitter Tx are not exactly those in the real time-variant channel. In order to avoid the ranging errors caused by the large deviation between the simulated $n_1$ and $n_2$ values and the real $n_1$ and $n_2$ values in each round of ranging estimation, we propose an iterative ranging based on MLE (maximum likelihood estimation) in UWB wireless sensor networks.

Suppose *PLi* is the *i*th observation value, we get the joint conditional *pdf* $p(PL|d)$ using Equation (12).

$$p(PL \mid d) = \prod_{i=1}^{N} \frac{e^{-\frac{\left(PL_i - (PL_0 + 10\gamma \log_{10} d)\right)^2}{2\left(\mu_\sigma{}^2 + \sigma_\sigma{}^2\right)}}}{\sqrt{2\pi\left(\mu_\sigma{}^2 + \sigma_\sigma{}^2\right)}}. \quad \textbf{(12)}$$

The necessary condition to compute the MLE of *d* is:

$$\frac{\partial \ln p(PL \mid d)}{\partial d} = \frac{10N\gamma}{\left(\mu_\sigma{}^2 + \sigma_\sigma{}^2\right) d \ln 10}\left(\frac{1}{N}\sum_{i=1}^{N} PL_i - PL_0 - 10\gamma \log_{10} d\right) = 0. \quad \textbf{(13)}$$

We solve Equation (13) and have:

$$\hat{\log_{10}} d = \frac{1}{10N\gamma}\sum_{i=1}^{N} PL_i - \frac{PL_0}{10\gamma}.$$

Therefore, the MLE based RSSI UWB ranging is:

$$\hat{d} = 10^{\frac{1}{10N\gamma}\sum_{i=1}^{N} PL_i - \frac{PL_0}{10\gamma}}. \quad \textbf{(14)}$$

## 6. Localization Algorithm

When computing the location of a wireless sensor node, there are two types of nodes, the reference node and the target node. Suppose that we have three reference nodes with coordinates $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$, respectively. The target node computes its coordinate $(x, y)$ using trilateration method with the coordinates of reference nodes and their ranges $d_1, d_2, d_3$, to the target node using the following equations:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 = d_3^2 \end{cases}$$

In practical situations, three reference nodes are usually insufficient to accurately derive the target coordinate due to ranging errors from thermal noise and other interferences. The least squares algorithm uses multiple reference nodes and the corresponding ranges to improve accuracy in the presence of error. It first creates following equations:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \quad \textbf{(15)}$$

where $(x_i, y_i)$ and $d_i$ ($i$ = 1,2,…,$n$)are the coordinate of the reference nodes, and the distances to the target node.

These equations can be linearized by subtracting the last low and performing some minor arithmetic shuffling, resulting in the following relations *AI* = *b*:

$$A = 2\begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ \vdots & \vdots \\ (x_{n-1} - x_n) & (y_{n-1} - y_n) \end{bmatrix},$$

$$I = \begin{bmatrix} x \\ y \end{bmatrix},$$

Copyright © 2009 SciRes. 

$$b=\begin{bmatrix} x_1^2-x_n^2+y_1^2-y_n^2+d_n^2-d_1^2 \\ x_2^2-x_n^2+y_2^2-y_n^2+d_n^2-d_2^2 \\ \vdots \\ x_{n-1}^2-x_n^2+y_{n-1}^2-y_n^2+d_n^2-d_{n-1}^2 \end{bmatrix}.$$

This work employs the following solution:

$$\hat{I}=\left(A^TA\right)^{-1}A^Tb. \quad \textbf{(16)}$$

Although the least squares algorithm could reduce the localization error, it requires a large amount of reference nodes within the communication radius of target node. Therefore, in the mining application of UWB wireless sensor networks, it is necessary to balance the cost and the accuracy.

In the following, we explore the relation between localization error and ranging error and then propose a modified algorithm based on the least squares algorithm.

Assume that the estimated distance between the target node and the *i*th reference node is $\hat{d}_i=d_i+\Delta_i$, where $\Delta_i$ is the ranging error, $i\in\{1,2,...,n\}$. From Equation (16), we have $\hat{I}=(A^TA)^{-1}A^Tb=Mb$, where

$$M=(A^TA)^{-1}A^T=\begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & .... & M_{1,n-1} \\ M_{2,1} & M_{2,2} & M_{2,3} & .... & M_{2,n-1} \end{bmatrix},$$

$$b=\begin{bmatrix} b_1 & b_2 & b_3 & .... & b_{n-1} \end{bmatrix}^T$$

$$=\begin{bmatrix} x_1^2-x_n^2+y_1^2-y_n^2+d_n^2-d_1^2+ \\ x_2^2-x_n^2+y_2^2-y_n^2+d_n^2-d_2^2+_2 \\ \vdots \\ x_{n-1}^2-x_n^2+y_{n-1}^2-y_n^2+d_n^2-d_{n-1}^2+ \end{bmatrix}$$

$$\begin{bmatrix} \Delta_n^2-2d_n\Delta_n-\Delta_1^2-2d_1\Delta_1 \\ \Delta_n^2-2d_n\Delta_n-\Delta_2^2-2d_2\Delta \\ \vdots \\ \Delta_n^2-2d_n\Delta_n-\Delta_{n-1}^2-2d_{n-1}\Delta_{n-1} \end{bmatrix}.$$

Denote $B_i=x_i^2-x_n^2+y_i^2-y_n^2+d_n^2-d_i^2$, we get the coordinate $(x,y)$ of target node:

$$x=\sum_{i=1}^{n-1}M_{1,i}B_i+\left(\Delta_n^2+2d_n\Delta_n\right)\sum_{i=1}^{n-1}M_{1,i}+\sum_{i=1}^{n-1}\left(-M_{1,i}\left(\Delta_i^2+2d_i\Delta_i\right)\right). \quad \textbf{(17)}$$

$$y=\sum_{i=1}^{n-1}M_{2,i}B_i+\left(\Delta_n^2+2d_n\Delta_n\right)\sum_{i=1}^{n-1}M_{2,i}+\sum_{i=1}^{n-1}\left(-M_{2,i}\left(\Delta_i^2+2d_i\Delta_i\right)\right). \quad \textbf{(18)}$$

Usually, $\Delta_i\sim N(0,\sigma_i^2)$, and are mutually independent. Therefore,

$$D(x)=\left(2\sigma_n^4+4d_n^2\sigma_n^2\right)\left(\sum_{i=1}^{n-1}M_{1,i}\right)^2+\sum_{i=1}^{n-1}\left(M_{1,i}^2\left(2\sigma_i^4+4d_i^2\sigma_i^2\right)\right). \quad \textbf{(19)}$$

$$D(y)=\left(2\sigma_n^4+4d_n^2\sigma_n^2\right)\left(\sum_{i=1}^{n-1}M_{2,i}\right)^2+\sum_{i=1}^{n-1}\left(M_{2,i}^2\left(2\sigma_i^4+4d_i^2\sigma_i^2\right)\right). \quad \textbf{(20)}$$

From above deduction, it is shown that localization error could be reduced proportionally to the reduction in the ranging error. According to the relation, if we guarantee the accuracy of $d_i$ in every equation, the trilateration and the least squares algorithm could be kept similarly accurate even if the number of equations is limited. The proposed ranging solution using MLE based on RSSI in this paper improves the ranging accuracy between target node and every reference node through iterative ranging, and reduces the ranging error caused by ranging based on a single round sampling. Furthermore, it reduces the influence of fixed value of $\gamma$ given in our path loss model on ranging and localization in time-variant channel. Hence, our localization algorithm can provide accurate ranging and localization with less UWB reference nodes.

## 7. Simulation Results

In order to verify our localization algorithms based on U-BOTH system for WSNs, we carried out simulation in the following scenarios:

1) With regard to the BER (bit error rate), we evaluate U-BOTH system performance in single and multi-user scenarios.
2) Using the sample network deployment as shown in Figure 4, we evaluate the ranging accuracy by comparing the results with the Cramer-Rao lower bounds.
3) Using the same sample network as shown in Figure 4, we evaluate the impact of the number of itera-

Copyright © 2009 SciRes. 

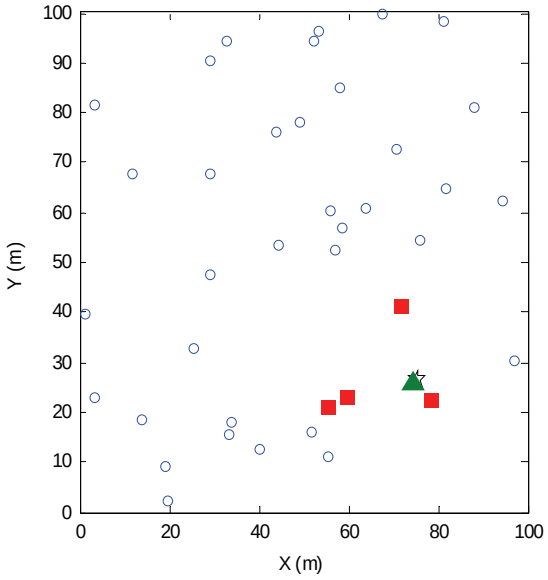



**Figure 4. Wireless sensor network simulation for localization.**

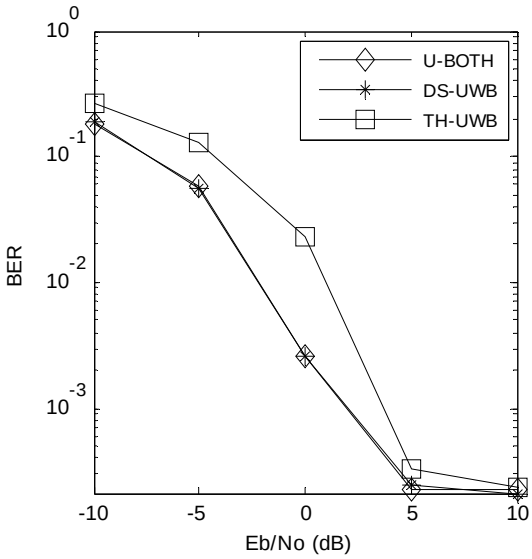



**Figure 5. Bit error rate in a single user system with additive white Gaussian noise (AWGN).**

tions in calculating the coordinate of a specific target node, indicated by the triangle in the diagram.

## 7.1. U-BOTH System Performance

We assume the channel is AWGN multipath-free single user channel; the transmitter and the receiver are synchronized perfectly. Then we randomly generate 2000 bits, every bit uses 4 pulses to repeat coding (*Ns* = 4).

Figure 5 illustrates the BER of the received signal using U-BOTH system, in contrast to DS-UWB that only uses direct sequence spreading, and TH-UWB that uses time-hopping pulse position modulation alone for UWB transmissions. We can see that the BER of U-BOTH and the DS-UWB system which use the $\pi$ -phase shift keying modulation are lower than TH-UWB. This is because the distance of two signals in binary phase shift keying (BPSK) modulation is $2\sqrt{E_{pulse}}$ , but $\sqrt{2E_{pulse}}$ in TH-UWB [23].

Secondly, we let $E_b = N_0 = 0$ dB, $Ns = 4$ and generated 2000 bits randomly. Figure 6 shows the relative performance of U-BOTH, TH-UWB and DS-UWB systems in multiple access scenarios. In this case, the received signal includes by noise and co-channel interference. In Figure 6, although both the BER and the variance of error bits increase as the number of users increases, the performance of our U-BOTH system is still better than DS-UWB and TH-UWB, proving that the UWB coding based OVSF-TH effectively handle the burst errors.

## 7.2. Evaluation of the Ranging Algorithms

In order to evaluate the performance of the ranging algorithms, we compare the parameter estimation errors against the Cramer-Rao lower bound (CRLB).

Denote $\hat{d}$ as the unbiased estimation of the parameter $d$ from

Equation (13), then the mean square error (MSE) of $\hat{d}$ is

$$MSE\left[\hat{d}\right] = E\left[\left(\hat{d}-d\right)^2\right] = E\left[\left(\hat{d}-E\left(d\right)\right)^2\right] = \mathrm{var}\left[\hat{d}\right]. \quad \textbf{(21)}$$

Hence in unbiased condition, the MSE of $\hat{d}$ is equal to the variance. The lower bound of the MSE based on UWB RSSI ranging could be represented by the CRLB:

$$CRLB\left(d\right) = \left(-E\left(\frac{\partial^2 \ln p(PL\,|\,d)}{\partial d^2}\right)\right)^{-1} = \frac{\left(\mu_\sigma{}^2 + \sigma_\sigma{}^2\right) d^2 \ln^2 10}{100 N \gamma^2}.$$

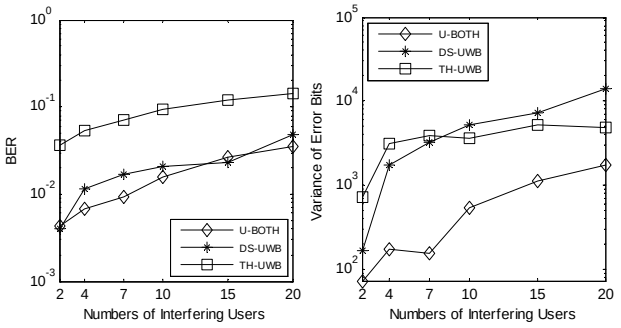



**Figure 6. Bit error rate and the variance of the number of error bits of 2000 generated bits.**

Copyright © 2009 SciRes. 

From Equation (13), we know that $\frac{\partial \ln p(PL \mid d)}{\partial d}$ can not be expressed in the form $K(d)[\hat{d}(PL)-d]$. So, the lower bound of MSE can not reach the CRLB. However, we can use the

MLE based RSSI ranging to enable the MSE to approach the CRLB. The MSE of UWB ranging is:

$$E\left[\left(\hat{d}-d\right)^2\right]=\int\left(\hat{d}-d\right)^2 p\left(PL \mid d\right)dPL\,. \qquad \textbf{(22)}$$

As *PLi* are mutually independent, we take Equation (12) and Equation (14) into Equation (22), and get the MSE of MLE based ranging using RSSI information:

$$e^{\frac{2\left(\mu_\sigma^2+\sigma_\sigma^2\right)\ln^2 10}{100N\gamma^2}+2\log_{10} d\ln 10}-2de^{\frac{\left(\mu_\sigma^2+\sigma_\sigma^2\right)\ln^2 10}{200N\gamma^2}+\log_{10} d\ln 10}+d^2\,. \qquad \textbf{(23)}$$

Therefore, the MSE of estimated distance $\hat{d}$ is the function of real distance d and the number of iterations *N*.

Based on the data in [7,11,16], we set $n_1$, $n_2 \in$ [-2, 2], $n_1 n_2 \in$ [-4, 4] and adopt values of UWB path loss model for simulations as shown in Table 2.

**Table 2. Portion of the simulation parameters.**

| Notation | Meaning | Value | |
|---|---|---|---|
| | | LOS | NLOS |
| $d_0$ | The reference distance | 1 *m* | 1 *m* |
| $PL_0$ | The path loss at reference distance | 47 *dB* | 51 *dB* |
| $\gamma$ | The path loss exponent | 1.7 | 3.5 |
| $\mu_\sigma$ | The mean value of shadow fading's standard deviation $\sigma$ | 1.6 | 2.7 |
| $\sigma_\sigma$ | The standard deviation of shadow fading's standard deviation $\sigma$ | 0.5 | 0.98 |

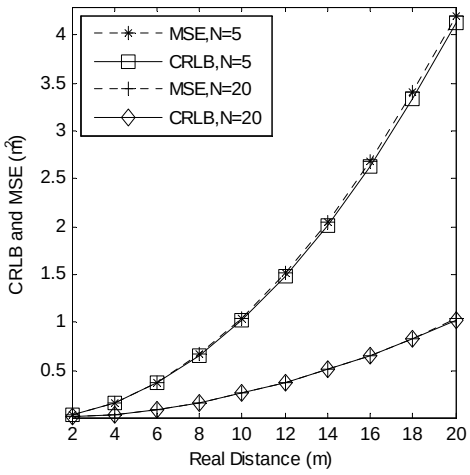



**Figure 7. Comparisons between the cramer-rao lower bound (CRLB) and the mean square error (MSE) of ranging estimations at different distances.**

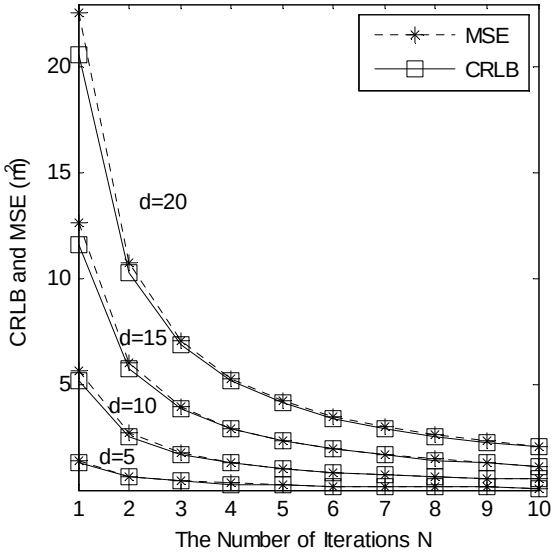



**Figure 8. Comparisons between the cramer-rao lower bound (CRLB) and the mean square error (MSE) of ranging estimations, regarding the number of iterations *N*.**

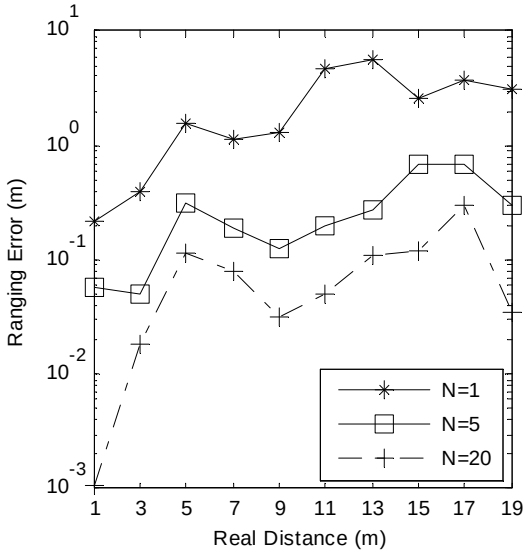



**Figure 9. The impact of the number of iteration *n* to the ranging errors at different distances.**

Figure 7 illustrates the relation between d and the CRLB, and the relation between d and the MSE. On one hand, it shows that the CRLB and the MSE increase when *d* increases, on the other, the MSE of ranging and the CRLB are always very close. When *d* is very small, they even overlap with one another. The more iterations we have for ranging, the smaller difference between the CRLB and the MSE (When $d = 4m$, $N = 20$, the CRLB is $0.0412m^2$, the corresponding MSE is $0.0414m^2$. When $d = 20m$, $N = 20$, the CRLB is 1.0310 $m^2$, the corresponding MSE is 1.0357 $m^2$).

Copyright © 2009 SciRes. 

In Figure 7, we can see that when $N = 20$, $d > 20m$, the MSE of ranging estimation grows higher than $1m^2$. Therefore, it is necessary to filter out large $d$ values in order to achieve higher ranging accuracy.

In the MLE based ranging using the RSSI values, the number of iterations $N$ is an important parameter. Figure 8 gives the relation between $N$ and the CRLB and the MSE, respectively. When $N$ increases, the CRLB and the MSE decrease rapidly. The MSE of ranging approximates the CRLB when $N$ is large enough (e.g. when $d = 5m$, $N = 10$, the CRLB is 0.1289 $m^2$, the corresponding MSE is 0.1300 $m^2$). Accordingly, we validate Equation (23), and prove the validity of our MLE method.

Figure 9 compares the MLE based ranging errors when the numbers of iterations $N$ are 1, 5, and 20. Even if thermal noises and other interferences cause the error to fluctuate randomly, we can see that higher numbers of iterations dramatically increase the accuracy of ranging computations.

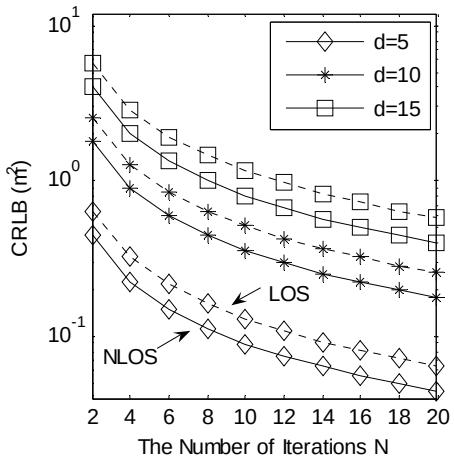



**Figure 10. The cramer-rao lower bound (CRLB) of ranging estimations in los (line of sight) and nlos (non-line of sight) environments.**

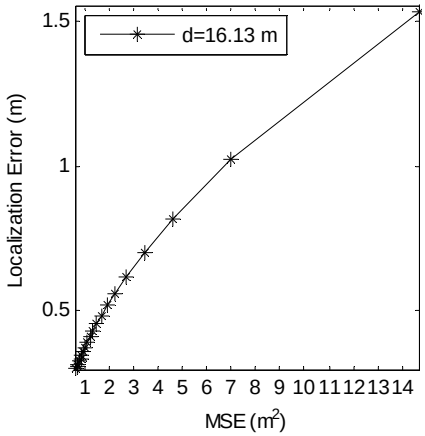



**Figure 11. Realtion between the mean square error (MSE) of ranging estimation and the localization error.**

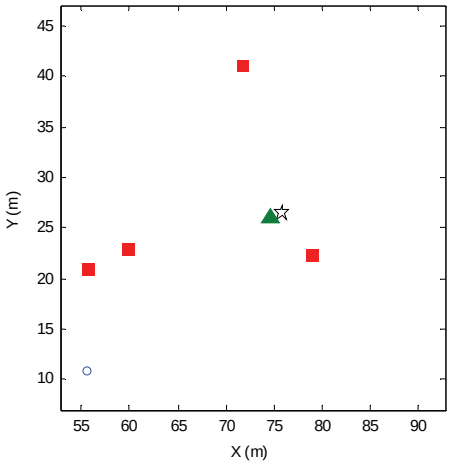



**Figure 12. Localization result of estimation time $N$=1 (localization error is 1.2547$m$).**

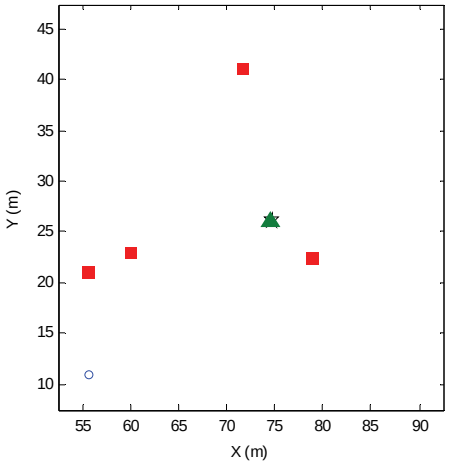



**Figure 13. Localization result of estimation time $N$=20 (localization error is 0.2464$m$).**

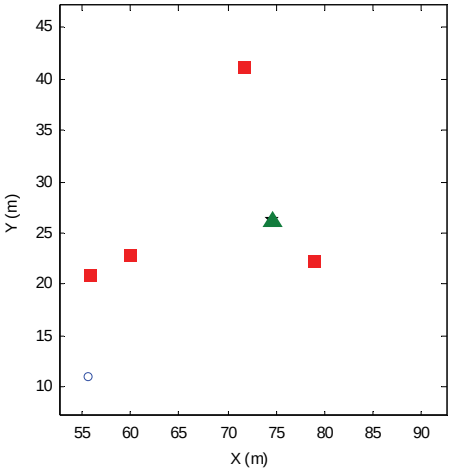



**Figure 14. Localization result of estimation time $N$= 100 (localization error is 0.0885$m$).**

Copyright © 2009 SciRes. 

Figure 10 analyzes the CRLB, which reflects the MSE of unbiased estimation, in LOS (line of sight) and NLOS (non-line of sight) environments. When $N$ increases, the CRLB in LOS and NLOS decrease correspondingly. For the same distance $d$ situations, the CRLB in NLOS is even smaller than the CRLB in LOS through our channel model and ranging method. Therefore, precise ranging and localization estimation also could be achieved in NLOS environment. This is especially attractive in coal mine environments.

### 7.3. Evaluation of the Localization Algorithms

From the analysis of Cramer-Rao low bound in Equation (21), the variance of ranging error can be shown in Equation (23). Similarly, the localization error can be expressed by the estimated coordinates and the real coordinates as $\sqrt{(\hat{x}-x)^2+(\hat{y}-y)^2}$.

Figure 11 shows the relation between ranging error and localization error in Equation (19) and Equation (20). It is obvious that the ranging error and localization error decrease when $N$ increases. The localization error when $N = 2$ is about 2/3 of that when $N = 1$. When $N = 20$, we could get the least localization error, which is $0.3009m$.

Secondly, we choose some nodes in Figure 4 to examine the impact of the number of iterations to the accuracy of localization computations as shown in Figure 12, Figure 13 and Figure 14. The triangle is the target node, the squares are reference nodes in the communication radius of the target node, the star is the estimated location of target node and the hollow circles are other nodes. We set the communication radius in Figure 4 to be $20m$, and the average number of reference nodes in this range to be $K = 5$. Therefore, allocating about 40 reference nodes in the $100m \times 100m$ mining area should be enough to monitor target node. As the location calculation algorithms described in Section 4-B, when $Tw = 1s$, $T_{RR} = 10ms$ and $\Delta_T = 10ms$, we can ensure 20 to 100 rangings between target node and every reference node. Figure 12, 13, 14 display the special deviation between estimated location and real location when $N = 1$, $N = 20$ and $N = 100$. Table 3 shows the localization error between the target node and a reference node that is $d = 14.9430m$ away, which shows that the impact of $N$ increments decreases when $N$ is greater than a few dozen.

**Table 3. The impact of the number of interations N to the localization errors when distance d = 14.9430m.**

| Iteration Number $N$ | Localization Error |
|---|---|
| $N = 1$ | $1.2547m$ |
| $N = 20$ | $0.2464m$ |
| $N = 50$ | $0.1253m$ |
| $N = 100$ | $0.0885m$ |

## 8. Conclusions

We proposed a group of communication protocols and localization algorithms for wireless sensor networks in coal mine environments, namely a new UWB coding method, called U-BOTH (UWB based on Orthogonal Variable Spreading Factor and Time Hopping), an ALOHA-type channel access method and a message exchange protocol to collect location information. Then we derived the corresponding UWB path loss model in order to apply the maximum likelihood estimation (MLE) method to compute the distances to the reference sensors using the RSSI information, and provided least squares (LS) method to estimate the coordinate of the moving target. The performance of U-BOTH communication system and the localization algorithms are analyzed using communication theories and simulations. Results show that UBOTH transmission technique can effectively reduce the bit error rate under the path loss model, and the corresponding ranging and localization algorithms can accurately compute moving object locations in coal mine environments.

## 9. Acknowledgment

We would like to express our sincere appreciation to Prof. Fanzi Zeng, and Prof. Juan Luo for their insightful feedbacks during the preparation of this manuscript, and of the anonymous reviewers for their helpful comments. This work has been generously sponsored in parts by the National Natural Science Foundation of China under Grant No. 60673061 and the Raytheon Company under Grant No. RC-42621.

## 10. References

[1] T. A. Alhmiedat and S. H. Yang, "A survey: Localization and tracking mobile targets through wireless sensors network," In The Eighth Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNET), 2007.

[2] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," IEEE Transactions on Signal Processing, Vol. 51, pp. 2137, 2003.

[3] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," In INFOCOM, 2000.

[4] K. Whitehouse and D. Culler, "Macro-calibration in sensor/actuator networks," In Mobile Networks and Applications (MONET), Vol. 8, pp. 463–472, 2003.

Copyright © 2009 SciRes. 

[5] T. He, J. A. Stankovic, C. Huang, T. Abdelzaher, and B. M. Blum, "Range-free localization schemes for large scale sensor networks," In Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM), pp. 81–95, 2003.

[6] IEEE Std 802.15.4a. part 15.4a, "Low rate alternative PHY task group (TG4a) for wireless personal area networks (WPANs)," Technical Report, IEEE, June 2007.

[7] A. F. Molisch, D. Cassioli, and C. -C. Chong, "A comprehensive standardized model for ultrawideband propagation channels," IEEE Transactions on Antennas and Propagation, Vol. 54, No. 11, pp. 3151–3166, 2006.

[8] I. Bucaille and A. Tonnerre, "MAC layer design for UWB LDR systems: PULSERS proposal," In 4th Workshop on Positioning, Navigation and Communication (WPNC), pp. 277–283, 2007.

[9] A. Fujii and H. Sekiguchi, "Impulse radio UWB positioning system," In IEEE Radio and Wireless Symposium, pp. 55–58, 2007.

[10] L. D. Nardis and M.-G. D. Benedetto, "Positioning accuracy in ultra wide band low data rate networks of uncoordinated terminals," In IEEE International Conference on UWB (ICUWB), pp. 611–616, 2006.

[11] S. Venkatesh and R. M. Buehrer, "Multiple-access design for Ad Hoc UWB position-location networks," In Proceedings IEEE Wireless Communications and Networking Conference (WCNC), Vol. 4, pp. 1866–1873, 2006.

[12] Y. Wang, Z. Wang, and H. Yu, "Simulation study and probe on UWB wireless communication in underground coal mine," Journal of China University Of Mining and Technology (English Edition), Vol. 16, No. 3, pp. 296–300, 2006.

[13] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communication," IEEE Transaction on Communication, Vol. 48, No. 4, pp. 679–691, 2000.

[14] M. G. D. Benedetto, and G. Giancola, "Understanding ultra wide band radio fundamentals," Prentice Hall, New Jersey, 2004.

[15] T. Gigl and G. J. M. Janssen, "Analysis of a UWB indoor positioning system based on received signal strength," In 4th Workshop on Positioning, Navigation and Communication (WPNC), pp. 97–101, 2007.

[16] S. S. Ghassemzadeh, R. Jana, C. W. Rice, W. Turin, and V. Tarokh, "Measurement and modeling of an ultra-wide bandwidth indoor channel," IEEE Transaction on Communication,Vol. 52, No. 10, pp. 1786–1796, 2004.

[17] F. Li, P. Han, X. Wu, and W. Xu, "Research of UWB signal propagation attenuation model in coal mine," Lecture Notes in Computer Science, Vol. 4611, pp. 819–828, 2007.

[18] C. Savarese, J.M. Rabaey, and J. Beutel, "Locationing in distributed Ad-Hoc wireless sensor networks," In Proceedings IEEE International Conference on Acoustics, Speech, and Signal, Vol. 4, pp. 2037–2040, 2001.

[19] F.Adachi, M. Sawahashi, and K. Okawa, "Tree-structured generation of orthogonal spreading codes with dierent lengths for the forward link of DS-CDMA mobile radio," IEE Electronics Letters, Vol. 1, No. 1, pp. 27–28, January 1997.

[20] H. Cam, "Nonblocking OVSF codes and enhancing network capacity for 3G wireless and beyond systems," Special Issue of Computer Communications on 3G Wireless and Beyond for Computer Communications, Vol. 26, No. 17, pp. 1907–1917, 2003.

[21] M. Joang and I. T. Lu, "Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network," In Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 776–83, New York, NY, USA, March 21−25, 1999.

[22] T. Makansi, "Trasmitter-oriented code assignment for multihop radio net-works," IEEE Transactions on Communications, Vol. 35, No. 12, pp. 1379–82, December 1987.

[23] J. G. proakis, Digital communications: Fourth Edition, McGraw- Hill, Columbus, 2001

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 422-432
doi:10.4236/ijcns.2009.25047 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Load Control for Overloaded MPLS/DiffServ Networks during SLA Negotiation

**Srećko KRILE[1], Dragan PERAKOVIĆ[2]**
[1]*University of Dubrovnik, Department of Electrical Engineering and Computing, Dubrovnik, Croatia*
[2]*University of Zagreb, Faculty of Transport and Traffic Engineering, Zagreb, Croatia*
*Email*: *srecko.krile@unidu.hr, dragan.perakovic@fpz.hr*
*Received April* 16, 2009*; revised May* 20, 2009; *accepted July* 2, 2009

## ABSTRACT

In end-to-end QoS provisioning some bandwidth portions on the link may be reserved for certain traffic classes (and for particular set of users) so the congestion problem of concurrent flows (traversing the network simultaneously) can appear. It means that in overloaded and poorly connected MPLS/DS networks the CR (Constraint-based Routing) becomes insufficient technique. If traffic engineering is supported with appropriate traffic load control the congestion possibility can be predicted before the utilization of guaranteed service. In that sense the initial (pro-active) routing can be pre-computed much earlier, possible during SLA (Service Level Agreement) negotiation. In the paper a load simulation technique for load balancing control purpose is proposed. It could be a very good solution for congestion avoidance and for better load-balancing purpose where links are running close to capacity. To be acceptable for real application such complicated load control technique needs very effective algorithm. Proposed algorithm was tested on the network with maximum $M$ core routers on the path and detail results are given for $N$=3 service classes. Further improvement through heuristic approach is made and results are discussed. Some heuristic options show significant complexity savings that is appropriate for load control in huge networks.

**Keywords:** End-To-End Qos Provisioning, Traffic Engineering In MPLS/Diffserv Networks, Constraint-Based Routing, Load Control during SLA Creation

## 1. Introduction

The core network for NGN (New Generation Network) is evolving to MPLS/DiffServ based network. With capability in service differentiation techniques the network operator can ensure the traffic priorization, specialy to quality voice (VoIP) and video calls (premium traffic), as same as for truly differentiated data services. It means that DiffServ classifies individual flows in a small number of service classes (at network edges). Also it enables ''soft'' reservation (allocation) of resources and special handling of packets in the core. Together, MPLS (Multi Protokol Label Switching) and DiffServ provide a scalable QoS solution for the core of NGN; see [1] and [2].

MPLS uses extensions to Resource Reservation Protocol (TE-RSVP) and the MPLS forwarding paradigm to provide explicit routing; see [3–5]. But end-to-end provisioning of coexisted and aggregated traffic in networks is still demanding problem, specially if over- provisioning is not possible. All traffic flows in domain are distributed among LSPs (Label Switching Path) related to $N$ service classes, but we know the IGP (Interior Getaway Protocol) uses simple on-line routing protocols (e.g. OSPF, IS-IS) based on shortest path methodology. With OSPF (*Open Shortest Path First*) some paths may become congested while others are underutilized. Constraint-based routing (CR) as an extension of explicit routing ensures traffic engineering (TE) capabilities. CSPF (Constrained Shortest Path First) allows an originating (ingress) router to compute a path (LSP) to egress router (sequence of intermediate LSRs), taking care of constraints such as bandwidth, delay and admi- nistrative policy; see [12]. It

Copyright © 2009 SciRes. 

can find out a longer but lightly loaded path, better than the heavily loaded shortest path. With constraint-based label distribution protocol (CR-LDP) the bandwidth provisioning dire- ctives and other information can be ensured (list of router's neighbors, attached networks, actual resource availability and other relevant information). It can be distributed for each service class and for each link along the path (LSP); see [6]. CR process can be incorporated into each ingress router and co-exists with the conventional routing technique.

MPLS/DiffServ aware TE (DS-TE) allows constraint-based routing of IP traffic with final task to adjust class load to actual class capacity. But the routing approach explained above can be effective in under loaded networks or in fully connected networks only. For them the RED or WRED (Weighted Random Early Detection) are effective congestion avoidance techniques. But in some networks dropping packets can lead to customer dissatisfaction and SLA violation. In the context of simultaneous flows (former contracted SLAs + new SLA creation) bandwidth overbooking is possible and congestion problem during service utilization can appear. For overloaded and poorly connected networks we need prediction of congestion probability much earlier. Load balancing has to be done much before the moment of service utilization, possibly during SLA negotiation process. To obtain quantitative end-to-end guarantees the QoS provisioning has to be in firm correlation with bandwidth management; see [7,8]. Also, such load control as a part of DS-TE can help in optimal bandwidth reservation, to predict sufficient resources and to ensure better end-to-end QoS provisioning, see [14]. TE involves the management of existing bandwidth resources to suit trafic demands and to meet the growing demands of customers in the network.

New approach in load control during SLA creation is given in Section 2. Load control can be seen as the capacity expansion problem (CEP) of $N$ capacity types. Expansions are possible in given limits for each bandwidth portion (sub-pool), influencing on each other. The mathematical model explanation for CEP is given in the Section 3. In the Section 4, we have CEP algorithm development and heuristic approach. The comparison of results for different algorithm options we can see in the Section 5.

## 2. LSP Creation during SLA Negotiation

The service provider for domain (e.g. ISP) wants to accept a new SLA that results with priority traffic flow between edge routers. A traffic trunk is defined as a logical pipeline within an LSP, with reservation of certain amount of capacity to serve the traffic associated with a certain SLA. So it is clear that LSP between an ingress/egress pair may carry multiple traffic trunks associated with different SLAs; see [10]. In Figure 2 we have situation on the path for the example of simultaneous SLA flows from Figure 1. All traffic flows on the path are participating possibly in the same time (the worst case). In that sense the network operator (e.g ISP) has to find the optimal LSPs for aggregated flows without any possible congestion in the core network; see [9]. Each traffic demand can be satisfied on appropriate or higher QoS level, using appropriate bandwidth portion from a sub-pool. The main condition is: the sufficient network resources must be available for the priority traffic at any moment.

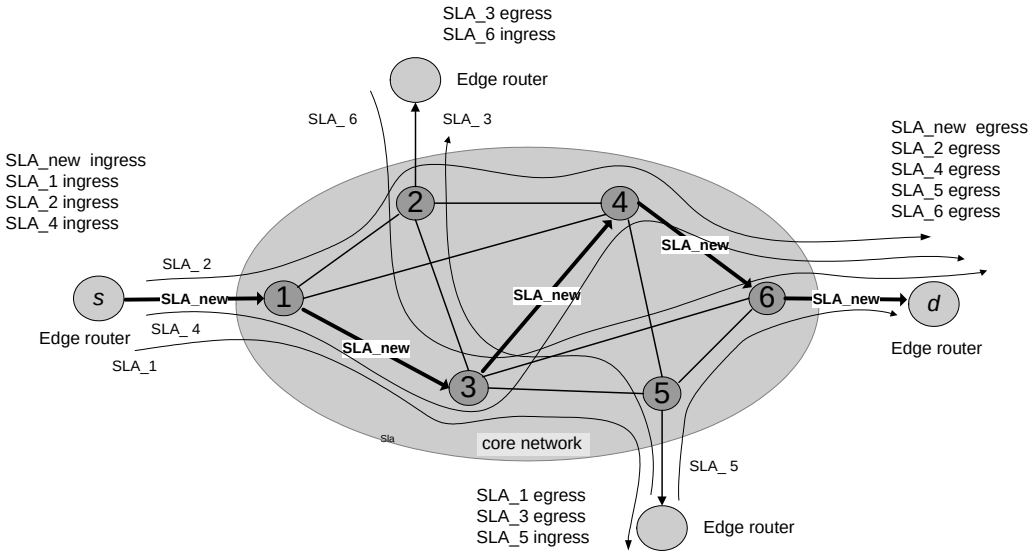



**Figure 1. An example of number of flows (former SLAs) in the context of new SLA creation.**

Copyright © 2009 SciRes. 

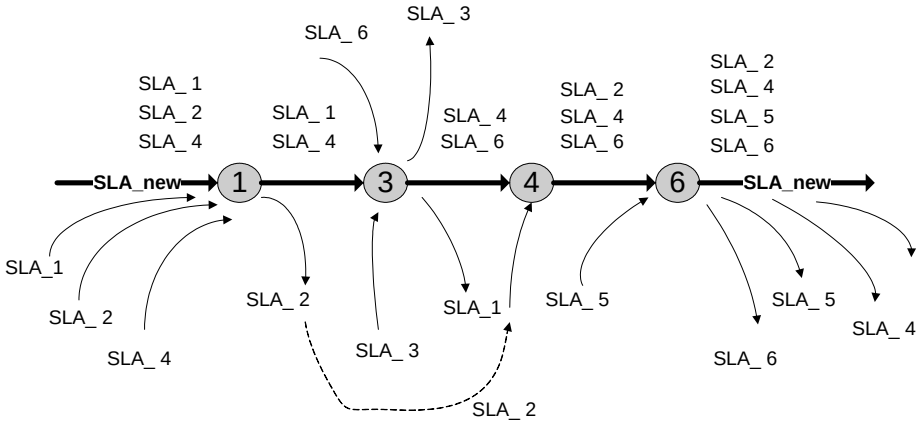



**Figure 2. Simultaneous flows with possibly congestion on the path.**

So we propose the next scenario: during SLA negotiation process the RM (Resource Manager) module has to determine the main parameters that characterize the required flow (i.e., bandwidth, QoS class, ingress and egress IP router addresses, time of service utilization); see [13]. At first RM can apply off-line routing to get initial LSP, taking care of all other simultaneous flows. It can be done with any shortest path-based routing algorithm (e.g. OSPF). From RM we can get statistical details for each SLA flow traversing the network simultaneously, e.g. ingress node, egress node, service class, bandwidth etc.). We need very effective algorithm to simulate such traffic load and detect congestion possibility on the path. Such optimization is a multi-constrained problem (MCP).

The BB (Bandwidth Broker) will check if there are enough resources on the calculated path to satisfy the requested service class. If such calculated path has any link that exceeds allowed capacity limits (maximal bandwidth for appropriate service class) possible congestion exists; see [15]. It means that link capacity on the path cannot be sufficient for new traffic load. Alternatively, adding capacity arrangement (dynamic bandwidth reservation) for congested link has to be done but it may produce significant extra cost. Abut dynamic bandwidth reservation mechanism we can see in [20] and [21].

If calculation finds that proposed path has no any congestion the new SLA can be accepted and related LSP is assigned to that traffic flow (SLA) and stored in database of RM. In opposite the new SLA cannot be accepted or must be re-negotiated. In the moment of service invocation such calculated (and stored) LSP can be easily distributed from RM to the MPLS network to support explicit routing, leveraging bandwidth reservation and prioritization; see [16].

In that way the LSP creation should be in co-relation with SLA, to enable better load-balancing and to ensure congestion avoidance in domain. Also, such load balancing technique is appropriate for inter-domain end-to-end path provisioning in the part of optimal bandwidth reservation from neighbor ASes (Autonomous System). Capacity reservations have to be done in the most effective way in order to provide bandwidth guarantees for predicted traffic; see [11]. Having purchased access to sufficient bandwidth from downstream ASes, home AS can utilize both: purchased bandwidth and its own network capacity.

## 3. Mathematic Model of CEP for Congestion Control and Load Balancing Purpose

The congestion control technique explained above can be seen as the capacity expansion problem (CEP) on the path with or without shortages. If the fulfillment of traffic demand is the main condition we talk about CEP without shortages. Transmission link on the path are capable to serve traffic demands for $N$ different QoS levels (service class) for $i = 1, 2, ..., N$. For each traffic load we need appropriate bandwidth amount, so it looks like bandwidth expansion. Bandwidth portions on the link can be assigned to traffic flow of appropriate service class up to the given limit of bandwidth sub-pool (maximal capacity for defined service class). Used capacity can be increased in two forms: by expansion or by conversion. Expansions can be done separately for each service class or through conversion (redirected amount) to lower quality class. It means that capacity can be reused to serve the traffic of lover quality level under special conditions. For example, if capacity predefined for priority traffic is unused it can be redirected to support best effort services. Bandwidth usage for each service class (sub-pool) can be a part of resource reservation strategy but sum of all sub-pools has to be equal/less than link capacity. Figure 3 gives an example of network flow representation for network flow for multiple QoS levels ($N$) and $M$ core routers (LSR) and M+1 links on the path.

Copyright © 2009 SciRes. 

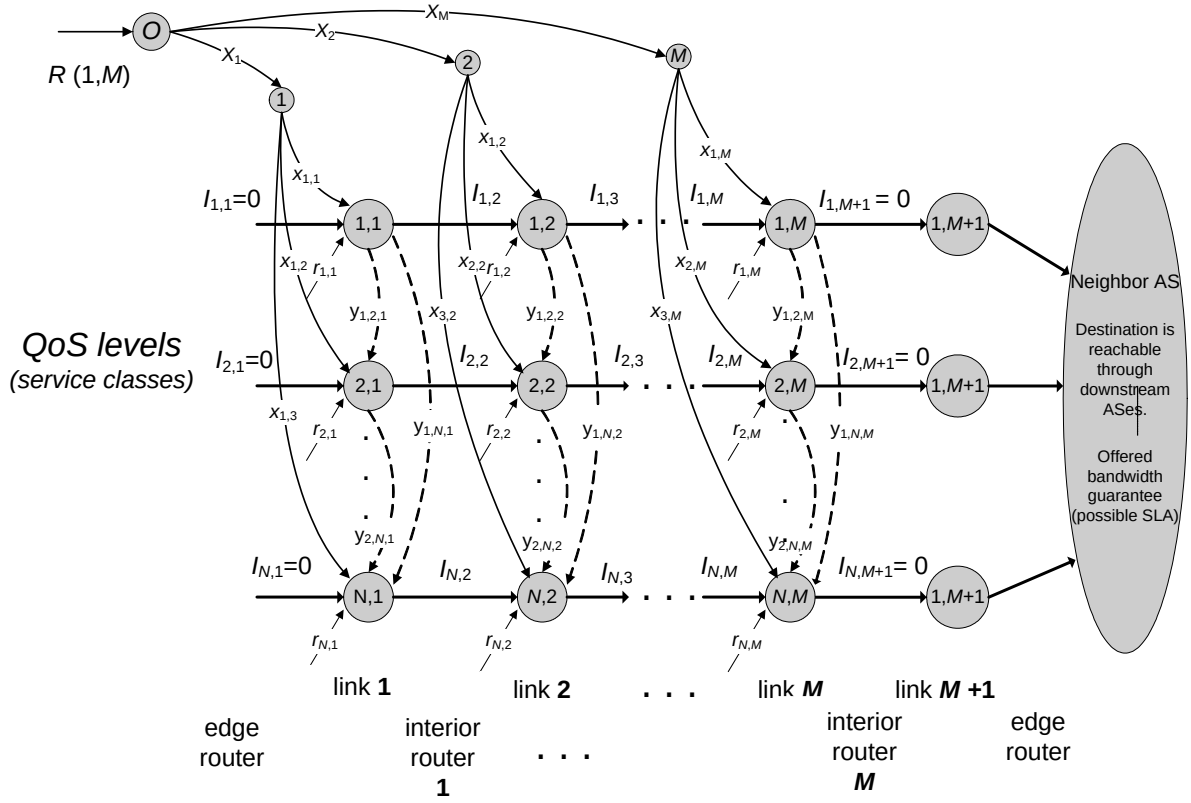



**Figure 3. The network flow representation of the CEP for Russian Dolls bandwidth allocation model.**

Network has $V$ core routers, $M \le V$, and $A$ links connecting all of them, including edge routers.

In the CEP model the following notation is used:

$i$, $j$ and $k$ = QoS level. We differentiate $n$ service classes (QoS levels).The $N$ levels are ranked from $i$ = 1, 2,..., $N$, from higher to lower.

$m$ = on the path, connecting two successive routers $m_1$ and $m_2$, $m$ = 1, …., $M$+1.

$r_{i,m}$ = traffic demand increment for additional capacity on the link $m$ from appropriate sub-pool $i$. For convenience, the $r_{i,m}$ is assumed to be integer. For the flow going out from the path $r_{i,m}$ is negative. The sum of traffic demands for capacity type $i$ between two routers on the path:

$$R_i(m_1, m_2) = \sum_{m=m_1}^{m_2} r_{i,m} \qquad \textbf{(1)}$$

$x_{i,m}$ = the amount of adding capacity for appropriate service class $i$ on the link $m$. Possible negative values – decrease. It means that we have idle (sufficient) capacity. If dynamic bandwidth management is possible such reduction can be realized.

$$X_m = \sum_{i=1}^{N} x_{i,m} \qquad \textbf{(2)}$$

The sum of demands for whole path and for all capacity types has to be positive or zero (including new SLA):

$$\sum_{i=1}^{N} R_i(1, M) = \sum_{m=1}^{M} X_m \ge 0 \qquad \textbf{(3)}$$

From this formulation it is obvious that sum of traffic demands on the path has to be equal to capacity amount used to satisfy them. It means that we don't expect reduction of total capacity on the path toward egress router, in other words we presume the increase of capacity.

$y_{i,j,m}$ = the amount of capacity for quality level $i$ on the link $m$, redirected to satisfy the traffic of lower quality level $j$.

Any traffic demand can also be satisfied by converted capacity from any capacity type $k < i$ with higher quality level. In Figure 3, such flows are marked with doted lines.

$I_{i,m}$ = relative amount for the capacity type $i$ on the link $m$, connecting two neighbor routers. Idle capacity is represented with positive value. If shortages are not allowed negative value cannot exist. $I_{i1} = 0$, $I_{i,M+1} = 0$ that means: no adding capacity is necessary on the link toward edge routers. It means that the capacity for that link is always sufficient.

Copyright © 2009 SciRes. 

$L_{i,m}$ = bandwidth constraints for link capacity values on the link $m$ and for appropriate service class $i$ ($L_{1,m}$, $L_{2,m}$, … $L_{N,m}$).

$w_{i,m}$= weight for the link $m$ and appropriate service class $i$ (QoS level).

$del_{i,m}$= delay on the link $m$ for appropriate service class $i$. Maximal delay on the path is denoted with $DEL_i$.

As we have nonlinear expansion functions (showing the economy of scale) the CEP can be solved by any nonlinear optimization technique. Instead of polynomial optimization (e.g. nonlinear convex programming), that can be very complicated (NP-complete), the network optimization methodology is efficiently applied. The main reason on such approach is the possibility of discrete capacity values for limited number of QoS classes, so the optimization can be significantly improved. The problem can be formulated as Minimum Cost Multi-Commodity Flow Problem (MCMCF). Such problem can be easily represented by multi-commodity the single (common) source multiple destination network; see Figure 3.

Let $G$ ($V$, $A$) denote a network topology, where $V$ is the set of vertices(nodes), representing link capacity states and $A$, the set of arcs, representing traffic flows between routers. Each link on the path is characterized by $z$-dimensional link weight vector, consisting of $z$-nonnegative QoS weights. The number of QoS measures (e.g. bandwidth, delay) is denoted by $z$. In general we have multi-constrained problem (MCP) but in this paper we talk about one-dimensional link weight vectors for $M$+1 links on the path $\{w_{i,m}, m \in A, i = 1, \ldots, N\}$. E.g. the capacity constraint for each link on the path is denoted with $L_{i,m}$ ($L_{1,m}$ $L_{2,m}$, … $L_{N,m}$). For non-additive measures (e.g. for bandwidth where the cost-function is concave, and we are looking for minimum) definition of the single-constrained problem is to find a path from ingress to egress node with minimal link weight along the path.

In the context of MCP we can introduce easily the adding constraint of max. Delay on the path (end-to-end). As it is an additive measure (more links on the path cause higher delay) it can be used as criteria to eliminate any unacceptable capacity expansion solution from calculation.

The flow situation on the link depends of expansion and conversion values ($x_{i,m}$, $y_{i,j,m}$). It means that the link weight (cost) is the function of used capacity: lower amount of used capacity (capacity utilization) gives lower weight. If the link expansion cost corresponds to the amount of used capacity, the objective is to find the optimal expansion policy that minimizes the total cost on the path. Definition of the single-constrained problem is to find a path $P$ from ingress to egress node such that:

$$w(P) = \min \sum_{m=1}^{M+1} \sum_{i=1}^{N} w_{i,m}(I_{i,m}, x_{i,m}, y_{i,j,m}) \quad \textbf{(4)}$$

where: $I_{i,m} \leq L_{i,m}$ **(5)**

satisfying condition of max. Delay for $P$:

$$\sum_{m_1}^{m_2} del_{i,m} \leq DEL_i \quad \textbf{(6)}$$

for $i = 1, \ldots, N$ ; $m = 1, \ldots, M$

A path obeying the above conditions is said to be feasible. Note that there may be multiple feasible paths between ingress and egress node. Generalizing the concept of the capacity states for each quality level of transmission link $m$ between LSRs in which the capacity states for each service class (QoS level) are known within defined limits we define *a capacity point* - $\alpha_m$.

$$\alpha_m = (I_{1,m}, I_{2,m}, \ldots, I_{N,m}) \quad \textbf{(7)}$$

$$\alpha_1 = \alpha_{M+1} = (0, 0, \ldots, 0) \quad \textbf{(8)}$$

In formulation (3.7) $\alpha_m$ denotes the vector of capacities $I_{i,m}$ for each service class on link $m$, and we call it capacity point. On the flow diagrams (Figure 2) each column represents a capacity point of the node, consisting of $N$ capacity state values (for $i$-th QoS level). Link capacity is capable to serve different service classes. Capacity amount labeled with $i$ is primarily used to serve traffic demands of that service class but it can be used to satisfy traffic of lower QoS Level $j$ ($j > i$).

Formulation (3.8) implies that idle capacities or capacity shortages are not allowed on the beginning and on the end of the path. It means that process is starting with new SLA flow that must be fully satisfied through the network (from ingress to egress node).

The objective function for CEP problem can be formulated as follows:

$$\min\left(\sum_{m=1}^{M+1}\left\{\sum_{i=1}^{N} c_{i,m}(x_{i,m}) + h_{i,m}(I_{i,m+1}) + g_{i,j,m}(y_{i,j,m})\right\}\right) \quad \textbf{(9)}$$

so that we have:

$$I_{i,m+1} = I_{i,m} + x_{i,m} - \sum_{j=i+1}^{N} y_{i,j,m} - r_{i,m} \quad \textbf{(10)}$$

$$I_{i,1} = I_{i,M+1} = 0 \quad \textbf{(11)}$$

for $m = 1, 2, \ldots, M+1$; $i = 1, 2, \ldots, N$; $j = i + 1, \ldots, N$.

In the objective function (3.9) the total cost (weight) includes some different costs. Expansion cost (adding capacity) is denoted with $c_{i,m}$ ($x_{i,m}$). For the link expansion in allowed limits we can set the expansion cost to zero. We can differentiate expansion cost for each service class. We can take in account the idle capacity cost $h_{i,m}$ ($I_{i,m+1}$), but only as a penalty cost to force the usage of the minimum link capacity (prevention of unused/idle capacity). Also we can introduce facility conversion cost $g_{i,j,m}$ ($y_{i,j,m}$) that can control non-effective usage of link capacity (e.g. usage of higher service class capacity instead). Costs are often represented by the fix-charge cost or with constant value. We assume that all cost functions

Copyright © 2009 SciRes. 

are concave and non-decreasing (reflecting economies of scale) and they differ from link to link. The objective function is necessarily non-linear cost. With different cost parameters we can influence on the optimization process, looking for benefits of the most appropriate expansion solution.

## 4. Algorithm Development

The network optimization can be divided in two steps. At first step weare calculating the minimal expansion weight $d_{u,v}$ for capacity expansion between two capacity points of neighbor links. It has to be done for all capacity points and for all pairs of neighbor routers (interconnected).

$u,v$ = the order number of capacity points in the sub-problem for appropriate link, $1 \le u, ..., v \le M+1$.

The calculation of weight value between two capacity points we call: capacity expansion sub-problem (CES); see (4.1). The expansion sub-problem for $N$ facilities $i$ = 1, 2, … , $N$ on the path between routers $u$ and $v$ is as:

$$d_{u,v} = \min\left\{\sum_{m=u}^{v}\left(\sum_{i=1}^{N} c_{i,m}(x_{i,m}) + h_{i,m}(I_{i,m+1}) + \sum_{j=m+1}^{N} g_{i,j,m}(y_{i,j,m})\right)\right\} \quad \textbf{(12)}$$

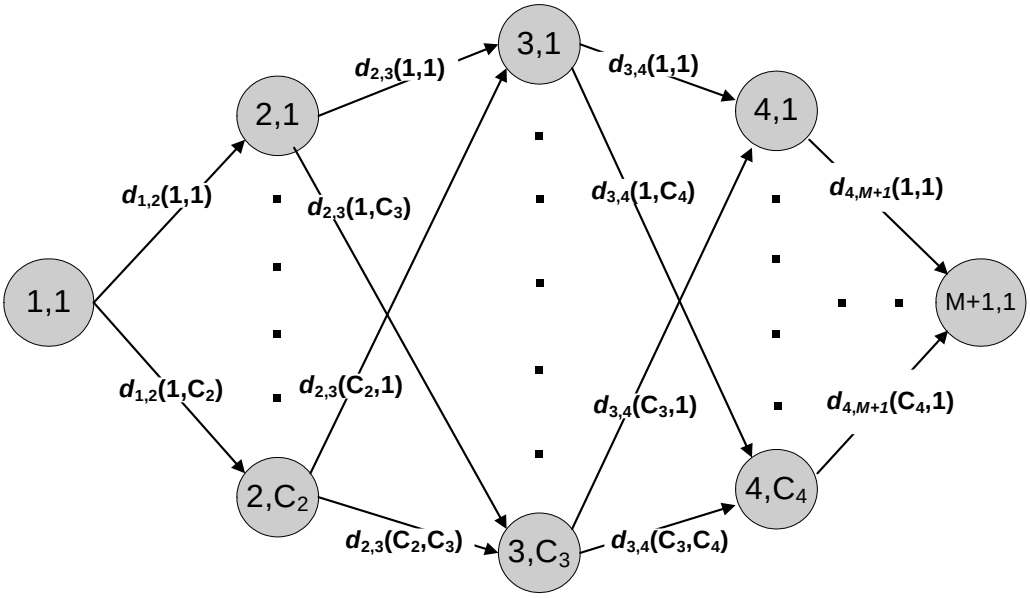



**Figure 4. The CEP problem can be seen as the shortest path problem for an acyclic network in which the nodes represent all possible values of capacity points and the links represent CES values.**

where:

$$I_{i,v} = I_{i,u} + D_i(u,v) - R_i(u,v) \quad \textbf{(13)}$$

$$R_i(u,v) = \sum_{m=u}^{v} r_{i,m} \quad \textbf{(14)}$$

$$D_i(u,v) = \sum_{m=u}^{v}\left(x_{i,m} - \sum_{j=1}^{N} y_{i,j,m}\right); \quad i \neq j \quad \textbf{(15)}$$

for $m$ = 1, 2, ... , $M$+1; $i$ = 1, 2, ... , $N$; $j$ = i + 1, ... , $N$.

Let $C_m$ be the number of the capacity point values for link $m$ between two neighbor core routers. Only one capacity point for the link that connects the edge router: $C_1 = C_{M+1} = 1$.

The total number of capacity points is:

$$C_p = \sum_{m=1}^{M+1} C_m \quad \textbf{(16)}$$

In the CEP we have to find many cost values $d_{u,v}(\alpha_u, \alpha_v)$ that emanate two capacity points, from each node ($u$, $\alpha_u$) to node ($v$, $\alpha_v$) for $v \ge u$. The total number of CES can be pretty large:

$$N_d = \sum_{m=1}^{M} C_m \cdot C_{m+1} \quad \textbf{(17)}$$

For every CES the calculation of many different expansion solutions can be derived from $D_i$ value. Many combinations exist from expansion and conversion amount.

The most of the computational effort is spent on computing of the sub-problem values. The number of all pos-

Copyright © 2009 SciRes. 

sible $d_{u,v}$ values depends on the total number of capacity points; see Figure 4.

Suppose that all links (sub-problems) are calculated, the optimal solution for CEP can be found by searching for the optimal sequence of capacity points and their associated link state values. On that level the CEP problem can be seen as a shortest path problem for an acyclic network in which the nodes represent all capacity point values, and branches represent CES values; see Figure 4, Then Dijkstra's algorithm or any similar algorithm can be applied.

The number of all possible $d_{u,v}(\alpha_u, \alpha_v)$ values depends on the total number of capacity points. It is very important to reduce that number ($C_p$) and that can be done through imposing of appropriate capacity bounds or by introduction of adding constraints (e.g. max. delay). Through numerical test-examples we'll see that many expansion solutions cannot be a part of the optimal expansion sequence.

### 4.1. Single Location Expansion Problem

Approach described in chapter above requires solving repeatedly a certain single location expansion problem (SLEP) in all possible modifications, looking for the best result. Let $\mathrm{SLEP}_{i,j}(m, D_i, \ldots D_j)$ be a *Single Location Expansion Problem* associated with link $m$ for facility (capacity) type $i, i+1, \ldots, j$ and corresponding values of *capacity change intention* $D_i, D_{i+1}, \ldots, D_j$.

For example, in solving $\mathrm{SLEP}_{1,3}$ for three different capacity types (bandwidth sub-pools) we have many expansion solutions divided into three different scenarios (expansion strategies):

a). capacity changes of one capacity type are not correlated with changes of others;

b). capacity changes of two capacity types depend on each other, but change of the third is independent;

c). capacity changes for all of three capacity types depend on each other.

From three expansion scenarios (expansion strategy) many different expansion solutions can be derived, depending on $D_i$ polarity. A lot of them are not acceptable and are not part of optimal sequence. For this problem an acceptable expansion solution has to satisfy some basic properties:

$$x_{i,m} \cdot D_{i,m} \geq 0 \quad \textbf{(18)}$$

$$y_{i,j,m} \cdot D_{i,m} \leq 0 \quad \textbf{(19)}$$

$$y_{i,j,m} \cdot D_{j,m} \geq 0 \quad \textbf{(20)}$$

Property (4.1.1) implies that the expansion (increase) of capacity type $i$ cannot be acceptable if that facility has intention to be reduced on location (link) $m$ ($D_{i,m} < 0$). Similar stays for negative values.

Expansion (capacity increase) is also possible through conversion, so (18) and (19) imply the similar restriction as (20). Zero value of any capacity type means that any change of capacity is allowed. In scenario A. we have only one possible expansion solution. In scenario B. we can combine all three capacity types in couples. In scenario C. we can see that only one expansion solution exists. Totally, we have five different expansion solutions with many variations.

In scenarios B. and C. we have expansion solutions with conversions of capacity from one type to another. It can be done as stand-alone expansion or together with expansion. That means that the conversion is just complementary with the expansion in satisfying of traffic demands.

Conversions can be applied only when idle capacities are noticed or negative demand increments are present. Special case is occurred when conversion $y_{i,j,m}$ eliminates both: eliminating idle capacity of type $i$ plus satisfying traffic demands of capacity type $j$. Also we can make distinction between two options: the partial expansion and the excessive expansion. The partial expansion $x_{j,m}$ means that demands are satisfied by expansion of appropriate capacity type $j$ plus by conversion $y_{i,j,m}$ of capacity type $i$ with higher quality level, but only if shortage of facility $i$ is not occurred.

The excessive expansion means that the expansion amount $x_{i,m}$ is used to partially expand facility $i$ and to satisfy demands for lower capacity type $j$, with conversion amount $y_{i,j,m}$.

### 4.2. Adding Properties (the Improvement of CEP Algorithm)

The most of the computational effort is spent on computing of the sub-problem values. But a lot of expansion solutions are not acceptable and they cannot be a part of the optimal expansion sequence. The key for this very effective approach is in fact that extreme flow theory enables separation of these extreme flows which can be included in optimal expansion solution from those which cannot be. Any of $d_{u,v}$ value, if it cannot be a part of the optimal sequence, is set to infinity. It can be shown that a feasible flow in the network given in Figure 3 corresponds to an extreme point solution of CEP if and only if it is not the part of any cycle (loop) with positive flows, in which all flows satisfy given properties; see [18]. One may observe that the absence of cycles with positive flows implies that each node has at most one incoming flow from the source node (positive or negative). This result holds for all single source networks. That means that optimal solution of $d_{u,v}$ has at most one expansion (or reduction) for each facility.

Copyright © 2009 SciRes. 

Using a network flow theory approach, adding properties of extreme point solution are identified. These properties are used to develop an efficient search for the link costs $d_{u,v}$. Absence of such cycles with positive flows implies that extreme point solutions for CEP satisfy the following properties:

$$I_{i,m} \cdot x_{i,m} \le 0 \quad \textbf{(21)}$$

$$I_{i,m} \cdot y_{i,j,m} \ge 0 \quad \textbf{(22)}$$

$$I_{j,m} \cdot y_{i,j,m} \le 0 \quad \textbf{(23)}$$

$$I_{j,m} \cdot x_{i,m} \cdot y_{i,j,m} = 0 \quad \text{if } x_{i,m} \cdot y_{i,j,m} \ne 0 \quad \textbf{(24)}$$

$$I_{i,m} \cdot I_{j,m} \cdot y_{i,k,m} \cdot y_{j,k,m} = 0 \quad \text{if} \quad y_{i,k,m} \cdot y_{j,k,m} \ne 0 \quad \textbf{(25)}$$

for: $i, j, k = 1, 2, 3\ i \ne k \ne j$; $m = 1, \ldots, M+1$

Properties (21) to (25) imply that the capacity of any capacity type is changed through an expansion, reduction or by conversion only if it doesn't make cycles with positive flows.

(21) and (22) imply that the capacity of any capacity type can be increased by an expansion or by a conversion only if there is no idle capacity. Similar rule exists for reduction of idle capacity.

(23) implies that capacity can be reduced only if there is no capacity shortage.

(24) implies that incoming flow of facility, going to be converted (reduced) in partially or excessive expansion solution, has to be zero. If not, cycles with positive flows can be occurred; see Figure 5, on that diagram we have idle capacity from previous link (for first and second class). The third class is satisfied with capacity conversions

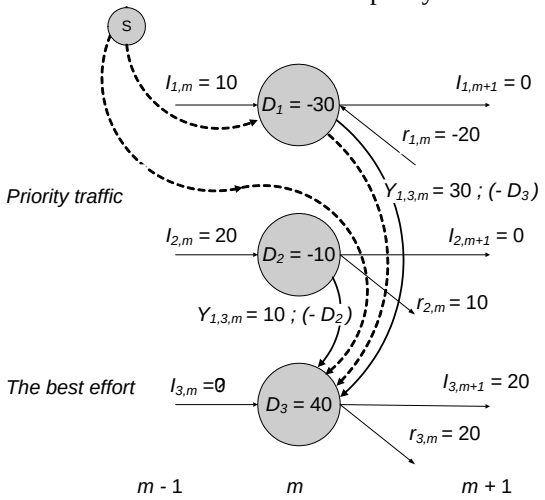



**Figure 5. An example of single location expansion solution that cannot be a part of the extreme solution.**

of higher classes. On that diagram dotted lines mark a cycle with positive flows from the common source. It means that such solution cannot be a part of extreme solution and has to be avoided from further calculation. One of the capacity values ($I_{2,m}$ or $I_{3,m}$) must be zero.

Property (25) is used for simultaneous multiconversion solution from scenario C. Only one incoming flow of converted (reduced) facility can exist. It means that two incoming flows are not allowed in the same time. In the case of simultaneous conversions, incoming flows have to be zero.

We can say that any acceptable $SLEP_{1,3}$ expansion solution for any CES have to satisfy properties (18)–(20) and (21)–(25). So many expansion solutions are not a part of optimal sequence and could be eliminated from further computation; see [19]. It means that any of subproblem value if it cannot be a part of the optimal sequence is set to infinity.

## 5. Testing Results and Comparison of Different Algorithm Options

The proposed algorithm is tested on many numerical test-examples, looking for optimal expansion sequence on the path. Between edge routers there are maximum $M$ core routers (LSR) and the path consists of maximum $M$+1links. Traffic demands (former contracted SLAs) are given in relative amount for each interior router on the path. Demands are overlapping in time and are defined for each capacity type (service class). Results obtained by improved algorithm (reduction of unacceptable expansion solutions) are compared with results obtained by referent algorithm that is calculating all possible expansion solutions for each CES.

For each test-example we know the total number of

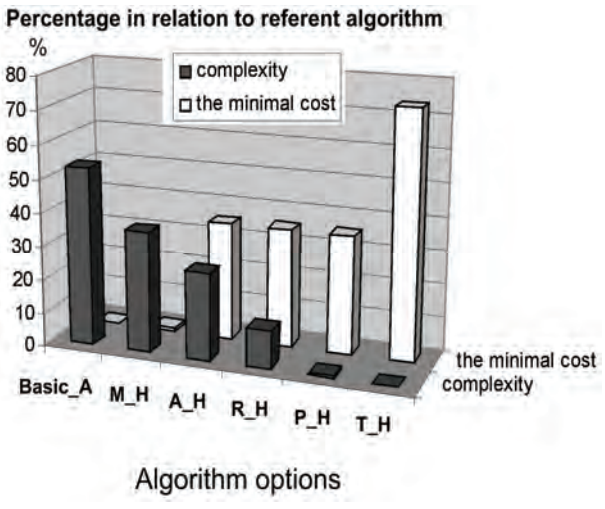



**Figure 6. Trends of algorithm complexity and comparison of results (minimal cost).**

Copyright © 2009 SciRes. 

**Table 1. Results of numerical test-example.**

| Traffic demands (increment) | | | | Algorithm option | The best result (minimal cost) | Number of capacity points | Number of sub-problems satisfying properties | Computational savings in perc. (%) |
|---|---|---|---|---|---|---|---|---|
| Routers on the path | $r_{1,m}$ | $r_{2,m}$ | $r_{3,m}$ | | | | | |
| 1 | -10 | 0 | 10 | Full approach | 9 487,58 | 839 | 30 133 | - |
| 2 | 10 | 10 | 0 | Basic_A | 9 487,58 | 839 | 17 869 | 40,70 (59,30) |
| 3 | 0 | 0 | 0 | M_H | 9 487,58 | 590 | 12 249 | 59,35 (40,65) |
| 4 | 10 | -10 | 10 | A_H | 9 487,58 | 424 | 11 146 | 63,01 (36,99) |
| 5 | -10 | 0 | 0 | R_H | 9 487,58 | 394 | 5 324 | 82,31 (17,69) |
| 6 | 10 | 10 | 0 | P_H | 9 487,58 | 91 | 963 | 96,80 (3,20) |
| | | | | T_H | 9 487,58 | 7 | 6 | 99,99 (0,01) |

$c_{i,m}(x_{i,m}) = f_i^{\,m-1}(A_i + B_i x_{i,m}^{\,ai})$, $A_1$=3000, $B_1$=25, $a_1$ = 0.9, $A_2$=1000, $B_2$=20, $a_2$=0.85, $A_3$=2000, $B_3$=30, $a_3$=0.95
For negative expansions ($x_{i,m} < 0$) $c_{i,m}(x_{i,m}) = - f_i^{\,m-1}(B_i\, abs(x_{i,m})^{ai})$.
$h_{i,m}(I_{i,m+1}) = f_i^{\,m-1} H_i\, I_{i,m+1}$, $H_i$ = 400 ($I_{i,m} > 0$) for $i$ = 1, 2, 3. Shortages ($I_{i,m} < 0$) are not allowed;
$g_{i,t}(y_{i,j,m}) = f_i^{\,m-1} G_i\ \ y_{i,j,m}$, $G_i$ =100 for $y_{i,j,m} > 0$, $i < j$. Conversions in opposite direction ($y_{i\,j,,m} < 0$) are not allowed;
For all positions and for all quality levels it is the same value $f_i$ = 0.9. All cost values are the same no matter of the link position on the path. In this test-example expansion limits $L_{i,m}$ are satisfied.

Optimal usage of the capacity (expansion sequence):

$y_{1,3,1} = 10$, $x_{1,2} = 10$, $x_{2,2} = 10$, $x_{1,4} = 20$, $y_{1,3,4} = 10$, $x_{2,4} = -10$, $x_{1,5} = -10$, $x_{1,6} = 10$, $x_{2,6} = 10$

Minimal cost: 9 487,58

capacity points. The number of possible CES is wellknown, so it is the measure of the complexity for the CEP-problem. Also, for each test-example we can see the number of acceptable sub-problems, satisfying basic and additional properties of optimal flow; see example from table 1. For all numerical test-examples the best possible result (near-optimal expansion sequence) can be obtained with improved algorithm (denoted with Basic_A), same as with referent algorithm (without reduction of unacceptable expansion solutions). For $N$=3 and $M$=6 algorithm complexity savings in percents are on average more that 40 % that is proportionally reflected on computation time savings; see Figure 6.

The number of all possible CES values depends on the total number of capacity points. So CEP requires the computation effort of $O(NMN_d)$ with linear influence of $N$. In real application we normally apply definite granularity of capacity values through discrete values (integer) of traffic demands $R_i$. It reduces the number of the capacity points significantly. Because of that the minimal step of capacity change ($step\_I_i$) has strong influence on the algorithm complexity.

In real situation we can introduce some limitations on the capacity state value, talking about heuristic algorithm options:

a) Only one negative capacity value in the capacity point. Such option is denoted with M_H *(Minimal-shortage Heuristic option*);

b) Total sum of the link capacity values (for all quality levels) is positive A_H *(Acceptable Heuristic option*);

c) Total sum is positive but only one value can be negative. Such option is denoted with R_H (*Real Heuristic option*);

d) Algorithm option that allows only non-negative capacity state values is denoted with P_H (*Positive Heuristic option*);

e) Only null capacity values are allowed. A trivial heuristic option (denoted with T_H) allows only zero values in capacity point (only one capacity point).

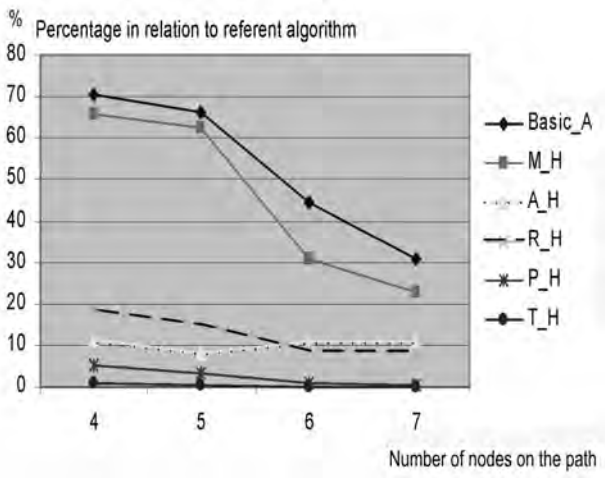



**Figure 7. The complexity savings increase with value *M*.**

Copyright © 2009 SciRes. 

We compared the efficiency of algorithm in above mentioned options. In Figure 6 we can see the average values of results for $N$=3 and $M$=6. Only for few test-examples (see table 1.) we can find the best expansion sequence, providing the minimal cost, no matter of algorithm option we use. For the most examples algorithm option M_H can obtain the best result with average saving more than 60 %. For other algorithm options the significant reduction of complexity is obvious but deterioration of result appears. In the most cases the trivial algorithm option (T_H) shows the significant deterioration of final result. A good fact for all algorithm options is that efficiency rises with increase of value $M$; see Figure 7.

## 6. Conclusions

Inappropriate bandwidth reservation or wrong traffic load could result in congestion possibilities. In this paper we propose an efficient algorithm for congestion control and load balancing purpose during SLA negotiation process.

We can check congestion probabilities on the path with algorithm of very low complexity first (e.g. P_H algorithm option). It means that only if congestion appears we need optimization with more complex algorithm (e.g. A_H). With the most complex algorithm option (Basic_A) we can get the best possible result, so we can be sure if congestion on the path could appear or not. In the case of congestion appearance new SLA cannot be accepted or adding capacity arrangement should be done. It means that SLA re-negotiation has to be done and customer has to change the service parameters: e.g. bandwidth (data speed), period of service utilization etc.

The proposed algorithm for load control (with different options) can be efficiently incorporated in SLA negotiation process. It may improve end-to-end provisioning, especially for overloaded and poorly connected networks where over-provisioning is not acceptable.

## 7. References


[1] F. L. Faucheur, *et al.*, "Multi-protocol label switching (MPLS) support of differentiated services," Technical Report RFC 3270, IETF, 2002.

[2] V. Sarangan and C. Jyh-Cheng, "Comparative study of protocols for dynamic service negotiation in the next-generation Internet," IEEE Communication Magazine, Vol. 44, No. 3, pp. 151−156, 2006.

[3] N. Degrande, G. V. Hoey, P. L. V. Poussin, and S. Busch, "Inter-area traffic engineering in a differentiated services network," Journal of Network and Systems Management (JNSM), Vol. 11, No. 4, 2003.

[4] M. D'Arienzo, A. Pescape, and G. Ventre, "Dynamic service management in heterogeneous networks," Journal of Network and Systems Management (JNSM), Vol. 12, No. 3, pp. 349−370, 2004.

[5] K. Haddadou, S. G. Doudane, *et al*., "Designing scalable on-demand policy-based resource allocation in IP networks," IEEE Communications Magazine, Vol. 44, No. 3, pp. 142−149, 2006

[6] R. Boutaba, W. Szeto, and Y. Iraqi, "DORA: Efficient routing for MPLS traffic engineering," Journal of Network and Systems Management (JNSM), Vol. 10, No. 3, pp. 309−325, 2002.

[7] Y. Cheng, R. Farha, A. Tizghadam, *et al*., "Virtual network approach to scalable IP service deployment and efficient resource management," IEEE Communication Magazines, Vol. 43, No. 10, pp. 76−84, 2005.

[8] S. Lima, P. Carvalho, and V. Freitas, "Distributed admission control for QoS and SLS management," Journal of Network and Systems Management (JNSM), Vol. 12, No. 3, pp. 397−426, 2004

[9] J. Guichard, F. L. Faucheur, and J. P. Vasseur, "Definitive MPLS Designs," Cisco Press, pp. 253−264, 2005.

[10] O. Younis and S. Fahmy, "Constraint-based routing in the internet: Basic principles and recent research," IEEE Communications Surveys & Tutorials, Vol. 5, No. 1, pp. 2−13, 3rd quarter 2003.

[11] K. H. Ho, P. H. Michael, N. Wang, G. Pavlou, and S. Georgoulas, "Inter-autonomous system provisioning for end-to-end bandwidth guarantees," Computer Communications, Vol. 30, No. 18, pp. 3757−3777, 2007.

[12] S. Bhatnagar, S. Ganguly, and B. Nath, "Creating multipoint-to-point LSPs for traffic engineering," IEEE Communications Magazines, Vol. 43, No. 1, pp. 95−100, 2005.

[13] M. Morrow and A. Sayeed, "MPLS and next-generation networks: Foundations for NGN and enterprise virtualization, " Cisco Press, 2006.

[14] S. Bakiras and L. Victor, "A scalable architecture for end-to-end QoS provisioning," Computer Communications, Vol. 27, No. 13, pp. 1330–1340, 2004.

[15] S. Giordano, S. Salsano, and G. Ventre, "Advanced QoS provisioning in IP networks: The European premium IP projects," IEEE Communication Magazines, Vol. 41, No. 1, pp. 30−36, 2003.

[16] D. Kagklis, C. Tsakiris, and N. Liampotis, "Quality of service: A mechanism for explicit activation of IP services based on RSVP," Journal of Electrical Engineering, Vol. 54, No. 9−10, Bratislava, pp. 250−254, 2003.

[17] H. Luss, "A heuristic for capacity expansion planning with multiple facility types," Naval Res. Log. Quart., Vol. 33, No. 4, pp. 685−701, 1986.

[18] W. I. Zangwill, "Minimum concavecost flows in certain networks," Mgmt. Sci., Vol. 14, pp. 429−450, 1968


Copyright © 2009 SciRes. 

[19] S. Krile and D. Kuzumilovic, "The application of bandwidth optimization technique in SLA negotiation process," Proceedings of 11th CAMAD'06, International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Net Network, Trento, pp. 115−121, 2006.

[20] S. Dasgupta, J. C. D. Oliveira, and J. P. Vasseur, "A new distributed dynamic bandwidth reservation mechanism to improve resource utilization: Simulation and analysis on real network and traffic scenarios," Proceedings of 25th IEEE International Conference on Computer Communications INFOCOM, Barcelona, pp. 1−12, 2006.

[21] S. Dasgupta, J. C. D. Oliveira, and J. P. Vasseur, "Dynamic traffic engineering for mixed traffic on international networks," Computer Networks, Vol. 11, pp. 2237 −2258, 2008.

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 433-452
doi:10.4236/ijcns.2009.25048 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Incremental Network Programming for Wireless Sensors

**Jaein JEONG , David CULLER**
*Computer Science Division, University of California, Berkeley, California, USA*
*Email*: {*jaein,culler*}*@eecs.berkeley.edu*
*Received April* 4, 2009; *revised May* 12, 2009; *accepted July* 5, 2009

## ABSTRACT

We present an incremental network programming mechanism which reprograms wireless sensors quickly by transmitting the incremental changes using the Rsync algorithm; we generate the difference of the two program images allowing us to distribute only the key changes. Unlike previous approaches, our design does not assume any prior knowledge of the program code structure and can be applied to any hardware platform. To meet the resource constraints of wireless sensors, we tuned the Rsync algorithm which was originally made for updating binary files among powerful host machines. The sensor node processes the delivery and the decoding of the difference script separately making it easy to extend for multi-hop network programming. We are able to get a speed-up of 9.1 for changing a constant and 2.1 to 2.5 for changing a few lines in the source code.

**Keywords:** Network Programming, Incremental, Wireless Sensor Networks, Difference Generation, Rsync Algorithm

## 1. Introduction

Typically, wireless sensors are designed for low power consumption and small size and don't have enough computing power and storage to support a rich programming development environment. Thus, the program code is developed on a more powerful host machine and is loaded onto a sensor node afterwards. The program code is usually loaded onto a sensor node through the parallel or serial port of the host machine; this is called in-system programming. In-system programming (ISP) is the most common way of programming sensor nodes because most microcontrollers support program loading through the parallel or serial port. However, ISP can only load the program code to one sensor node at a time. The programming time increases proportional to the number of wireless sensors to be deployed. During the development cycle of wireless sensor software, the source code can be modified for bug fixes or to add additional functionalities. With ISP, the cost of a software update is high; it involves all the efforts of collecting the sensor nodes placed at different locations and possibly disassembling and reassembling the enclosures. Network programming reduces these efforts by delivering the program code to each of the sensor nodes through the wireless links.

Network programming has been used since the introduction of TinyOS 1.1 release [1,2]. This implementation, XNP (Crossbow Network Programming), provides the basic capability of network programming; it delivers the program code to the sensor nodes remotely. However, it has some limitations: First, XNP does not scale to a large sensor network. XNP disseminates the program code only to the nodes that can be reached directly by the host machine. Therefore, the nodes outside the single hop boundary cannot be programmed. Second, XNP has a lower bandwidth compared than ISP. An experiment in [1] shows the programming time of XNP and ISP. In the experiment, we used a simple test application 'XnpCount' which has basic functionalities: network programming, counting numbers using LEDs and transmitting the number in radio packets. The version of 'XnpCount' we used was 37,000 bytes in size and required 841 XNP packets to transfer the entire program. The programming time of XNP was more than 4 times longer than that of ISP (Figure 1). When XNP updates the program code with another version, it sends the entire program code rather than the difference. This incurs the same program-

Copyright © 2009 SciRes. 

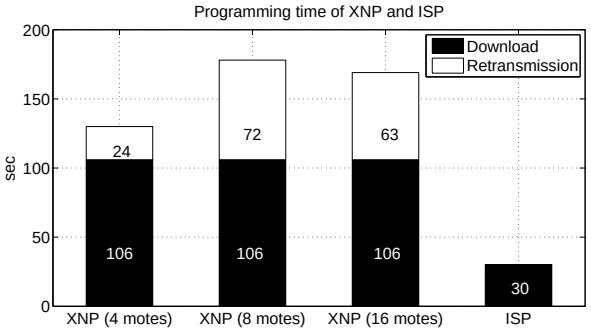



**Figure 1. Programming time of crossbow network programming (XNP) and in-system programming (ISP).**

ming time even when the difference is small. If the sensor nodes could build the program code image incrementally using the previous code image, the overall programming time can be reduced.

We present an incremental network programming mechanism which sends the new version of the program by transmitting the difference of the two program images. Unlike previous approaches, we generate the program code difference by comparing the program code in block level without any prior knowledge of the program code structure. This gives a general solution that can be applied to any hardware platform. We used the Rsync algorithm [3] to generate the difference. The Rsync algorithm finds the shared code blocks between the two program images and allows us to distribute only the key changes of the program. Originally, the Rsync algorithm was made for computationally powerful machines exchanging the update of binary files over a low-bandwidth communication link. We tuned the Rsync algorithm for wireless sensor network programming. First, we made the host program process expensive operations like building the hash table in favor of the sensor node. In order to rebuild the program image the sensor node simply reads or writes code blocks to flash memory.

Second, we structured the difference to avoid unnecessary flash memory accesses. In rebuilding the program image, the sensor node processes the script dissemination and the decoding in separate steps. This makes it easy to use dissemination protocols and to extend for multi-hop network programming. We are able to get a speed-up of 9.1 for changing a constant and 2.1 to 2.5 for changing a few lines in the source code over the non-incremental delivery.

The rest of the paper is organized as follows. Section 2 describes the in-system programming and the network programming as a background. Section 3 discusses the related work on wireless sensor network programming. Section 4 outlines incremental network programming and explains our first implementation. In Section 5, we use the Rsync algorithm to generate the program and show how this implementation improves performance. In Section 6, we discuss the extension to the script delivery which makes program delivery more reliable and faster. Finally, we conclude this thesis with Section 7.

## 2. Background

### 2.1. In-System Programming

The program development for wireless sensors starts with writing the source code. For the Berkeley sensor platform, the source code is written in the nesC programming language. Once the source code is successfully compiled, the binary code is generated (main.exe). The binary code is further converted to the Motorola SREC format (main.srec) and is then available for loading. The Motorola SREC format is an ASCII representation of binary code and each line of an SREC file contains the data bytes of the binary code with additional house keeping information (Figure 2).

With ISP, the binary code (SREC format) is loaded onto a sensor node through the direct connection (e.g. parallel port) from the host machine. The host programming tool (uisp) sends a special sequence of bytes that places the microcontroller of the sensor node in programming mode. While the microcontroller is in programming mode, the data bytes sent by the host programming tool are written directly to the program memory of the microcontroller (Figure 3(a)).

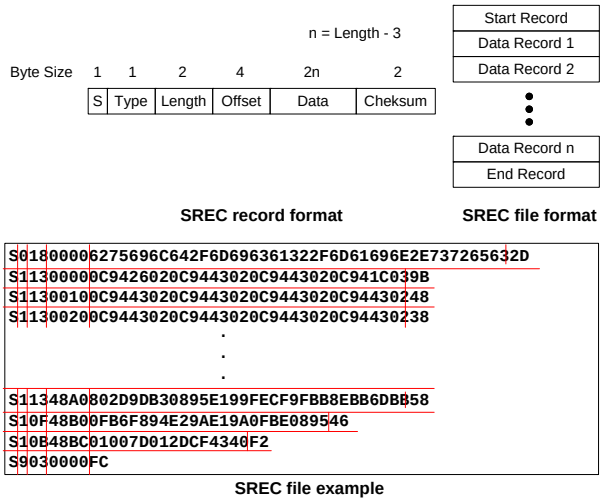
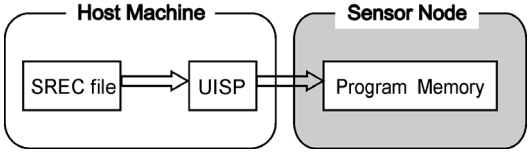
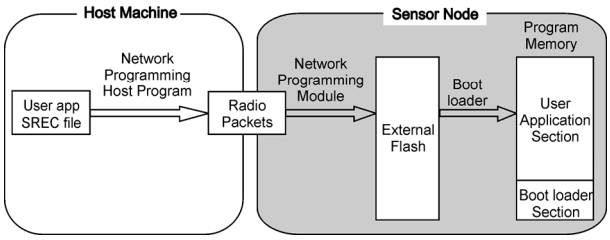



**Figure 2. Format of SREC file and its records with an example.**




**(a) Process of in-system programming**

Copyright © 2009 SciRes. 




**(b) Process of network programming**

**Figure 3. Steps for in-system programming and network programming.**

## 2.2. Network Programming

Network programming takes a different approach to loading the program code. Rather than writing the program code directly to program memory, network programming loads the program code in two steps. First, it delivers the program code to the sensor nodes. Second, it makes the sensor nodes move the downloaded code to program memory (Figure 3(b)).

In the first step, the network programming module stores the program code in external storage. Since the network programming module runs in user level as a part of the main application code, it does not have the privilege to write the program code into program memory. In the case of XNP, the network programming module writes the program code to the external flash memory outside program memory. The external flash memory of a MICA2/MICA2DOT mote is 512KB in size and is big enough for any application code (the maximum size of 128KB). During program delivery, part of the code may be missing due to the packet loss. The network programming module requests for any missing records of the program code to make sure that there are no missing records.

In the second step, the boot loader copies the program code from external flash memory to program memory. The boot loader is a program that resides in the high memory area (which we call the boot loader section) of an ATmega128 microcontroller and has the privileges to write data bytes to the user application section of program memory [4]. The boot loader starts execution when it is called by the network programming module. After it copies the program code from the external flash memory to program memory, it restarts the system.

In the paragraphs above, we assumed that the sensor nodes can update the current program image through network programming. However, a sensor node cannot be network programmed until it has the network programming module and the boot loader. Thus, we need to load the initial program code and the boot loader with ISP.

# 3. Related Work

## 3.1. Wireless Sensor Network Programming

XNP [1,2] is the network programming implementation for TinyOS that was introduced with 1.1 releases version. XNP supports basic network programming broadcasting the program code to multiple nodes in a single hop. However, it doesn't consider a large sensor network and incremental update.

MOAP [5] is a multihop network programming mechanism and their main contributions are its code dissemination and buffer management. For code dissemination, they used the Ripple dissemination protocol which disseminates the program code packets to a selective number of nodes without flooding the network with packets. For buffer management, they used a sliding window scheme which maintains a window of program code and allows lost packets within the window to be retransmitted. The sliding window uses a small footprint so that packets can be processed efficiently in on-chip RAM. MOAP was tested on the EmStar simulator and MICA2 motes.

Deluge [6] is a multihop network programming protocol that disseminates program code in an epidemic fashion to propagate program code while regulating the excess traffic. In order to increase the transmission rate, Deluge used optimization techniques like adjusting the packet transmit rate and spatial multiplexing. Unlike MOAP, Deluge uses a fixed sized page as a unit of buffer management and retransmission. Deluge was tested with the TOSSIM simulator [7] and MICA2 motes.

Reijers, *et al.* [8] developed an algorithm that updates binary images incrementally. With the algorithm, the host program generates an *edit script* to describe the difference between the two program code images. The sensor nodes build the program image after interpreting the edit script. The edit script consists of not only simple operations like copy and insert but also more complex operations (address repair and address patch) that modify the program code at the instruction level. This helps minimizing the edit script size. As an evaluation, this paper considers only the reduced script size on the host side. Since operations like address repair and address patch incur memory intensive EEPROM scanning, the experiments should have demonstrated the overall programming time in a sensor simulator or in a real sensor node.

Kapur, *et al.* [9,10] implemented a version of incremental network programming based on the algorithm of Reijers, *et al* [8]. Their implementation is composed of two parts: the diff encoder on the host side and the diff decoder on the sensor node side. The diff encoder generates the difference for the two versions of code at the

Copyright © 2009 SciRes. 

instruction level using copy, insert and repair operations. The difference script is delivered to the sensor node using MOAP [5] which was developed for reliable code dissemination. Then, the sensor node rebuilds the program code after decoding the downloaded script.

These two works on incremental network programming minimized the script transmission at the cost of program modification at the instruction level. In contrast, the implementation in this paper put less computational complexity on the sensor nodes. The difference generation, which is costly, is handled by the host program. The sensor nodes simply write the data blocks based on the script commands and this can be applied to less powerful sensor nodes.

While the examples above disseminated the program code in native binary code, Maté [11] distributes the program code in virtual machine instructions which are packed in radio packets. While XNP transmits the binary code that contains both the network programming module and the application, Maté only transmits the application code. This allows Maté to distribute the code quickly. One drawback of Maté is that it runs the program code only in virtual machine instructions and a regular sensor application needs to be converted to virtual machine instructions before execution.

Trickle [12] is an improvement over Maté. In Maté, each sensor node floods the network with packets to distribute the code and this can lead to network congestion but the algorithm can be used for a large sensor network. Trickle addresses this problem by using a "polite gossip" policy. Each sensor node periodically broadcasts a code summary to its local neighbors and stays quiet if it has recently heard a summary identical to its own summary. The sensor node broadcasts an update only when it hears from an older summary than its own.

### 3.2. Remote Code Update outside Wireless Sensor Community

Outside the sensor network community, there have been efforts to update program code incrementally. Emmerich *et al.* [13] demonstrated updating XML code in an incremental fashion. Specifying the update in XML is easier than a binary image because XML is a structured markup language and it allows specifying the update without changing the structure of the rest of the code. In contrast, inserting or replacing code blocks in binary code affects the rest of the code.

The cases of synchronizing general form of unstructured files can be found with Rsync and LBFS. Rsync [3] is a mechanism to efficiently synchronize two files connected over a low-bandwidth, bidirectional link. To find matching blocks between the two files, we can divide the first file into fixed sized blocks of B bytes and calculate the hash for each block. Then, we scan the second file and form a B byte window at each byte. After that we compare the hash for the window with hash values of all the blocks in the first file. This does not work that well. If the hash is expensive to calculate, finding the matching blocks will take long time. If the hash can be computed cheaply but with possible false matches, we may not find the correct block. The key idea of Rsync is to use two levels of hashes, rolling checksum (fast hash) and hash (strong hash) to make the computation overhead manageable while finding the matching blocks with high probability. Rsync calculates the rolling checksum of the B byte window of the second file at each byte and computes the hash only when the rolling checksums of the two blocks match. Since the hash is computed only for the possible matches, the cost of calculating the hash is manageable and we can filter out the false match.

LBFS [14], another mechanism to synchronize two files in a low-bandwidth, bidirectional link, takes a slightly different approach. Rather than divides a file into fixed blocks, LBFS divides each file into a number of variable sized blocks and computes the hash over each block. To find matching blocks between the two files, LBFS just compares these hashes (SHA-1 hash). The key idea of LBFS is in dividing a file into variable sized blocks. LBFS scans a file and forms a 48-byte window at each byte and calculates a 13-bit fingerprint. If the fingerprint matches a specific pattern, that position becomes the breakpoint of the block. This scheme has a property that modifying a block in a file does not change the hash values of the other blocks. When we are going to send a new version, we can just compare the hash values of each variable block and send only the non-matching blocks.

The mechanism patented by Metricom Inc. [15] disseminates the program code over multihop networks in an efficient way using an epidemic protocol. When a node *V* has a new version of code, it tells its neighbors that a new version of code is available. On hearing the advertisement from *V*, one of *V*'s neighbor, *P*, checks whether it has the newly advertised version. If it doesn't have the version, *P* requests *V* transmit the version of code. After that, *V* starts sending program code and finishes when it doesn't hear any requests. With this scheme, a sensor node can distribute the program code without causing much network traffic.

## 4. Design and Implementation

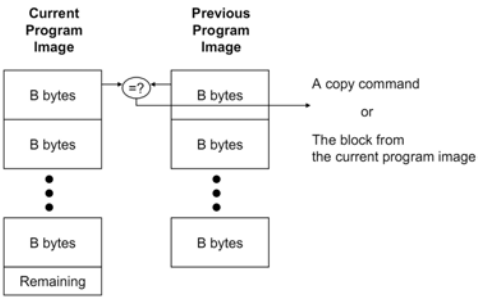
To design an incremental network programming mechanism, we need to consider some factors that affect performance. Compared to other sensor applications, network programming keeps a large amount of data in the sensor nodes contributing to long programming time.

Copyright © 2009 SciRes. 

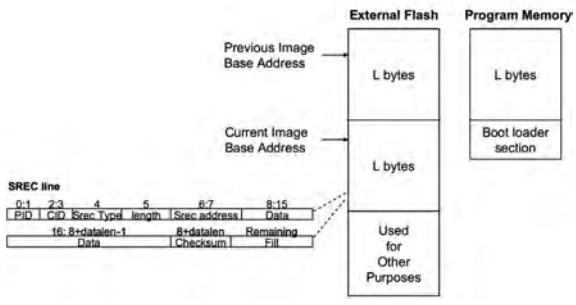
Since programming time is proportional to data size, reducing the amount of transmission data will improve programming time. External flash memory which is used for program storage also limits performance. The downloaded code is stored in the external flash memory because there is not enough space in on-chip memory. However, this external flash memory is much slower than the on-chip SRAM. For better performance, access to external memory should be made only when it is necessary. Caching frequently accessed data can help reducing flash memory accesses.

Another consideration is how much functionality is to be processed in the sensor nodes. More sophisticated algorithms could reduce overall programming time by reducing network traffic, but at the cost of higher complexity computation and memory accesses.

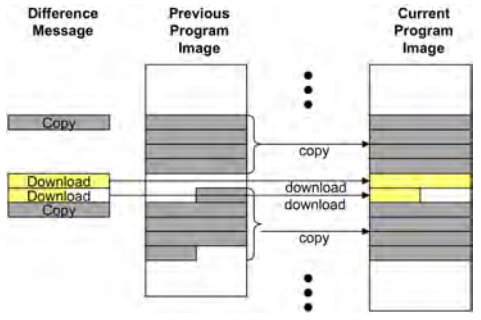
Finally, the design should be simple so that it can be understood and diagnosed without difficulty.




**(a) Generating difference**




**(b) Memory allocation**




**(c) Program image rebuild**

**Figure 4. Steps for incremental network programming.**

## 4.1. Design: Fixed Block Comparison

As a starting point, we can design a version of incremental network programming by extending XNP. This consists of two main parts: 1) difference generation and code delivery, 2) storage organization and image rebuild.

### 4.1.1. Difference Generation

To generate the program difference, the host program compares each fixed sized block of the new program image with the corresponding block of the previous image. We set the block size as the page size of the external flash memory (256 bytes). The host program sends the difference as messages while it compares the two program versions. If the two corresponding blocks match, the host program sends a CMD_COPY_BLOCK message. The message makes the network programming module in the sensor node copy the block of the previous image to the current image. When the two blocks don't match, the host program falls back to the normal download; it sends a number of CMD_DOWNLOADING messages for the SREC records of the block (Figure 4(a)).

The idea is that we can reduce the number of message transmissions by sending a CMD_COPY_BLOCK message instead of multiple CMD_DOWNLOADING messages when most of the blocks are the same between the two program images.

### 4.1.2. Operations

Table 1 shows the message types used for incremental network programming. Based on XNP messages, we made the following extensions for incremental network programming as in Figure 5.

- Start Download: CMD_START_DOWNLOAD_INCR message notifies the beginning of network programming in incremental mode. This message specifies not just the program ID of the current program but also the program ID of the previous program to ensure that the sensor node has the same program image as the host program.
- Download: Two operations CMD_DOWNLOADING and CMD_COPY_BLOCK are used to transmit the program image difference.
- Query and Reboot: The formats of query, reply and reboot messages are the same as XNP messages.
- Debugging Messages: CMD_GET_CURRENT_LINE and CMD_GET_PREV_LINE messages request the SREC record at the specified line. In response, the sensor node sends CMD_REPLY_LINE message.

Copyright © 2009 SciRes. 

**Table 1. Message types for incremental network programming.**

| Message ID | Description |
|---|---|
| CMD_START_DOWNLOAD | Start network programming in normal mode |
| CMD_DOWNLOADING | Deposit an SREC record |
| CMD_QUERY_COMPLETE | Signals that it received all the capsules |
| CMD_DOWNLOAD_STATUS | Request/response with download status |
| CMD_DOWNLOAD_COMPLETE | End of SREC record download |
| CMD_ISP_EXEC | Execute the boot loader |
| CMD_GET_PIDSTATUS | Get Program ID |
| CMD_GET_CIDMISSING | Retransmission message from the host |
| CMD_REQ_CIDMISSING | Request retransmission for a missing cap |
| CMD_START_DOWNLOAD_INCR | Start network programming incrementally |
| CMD_COPY_BLOCK | Copy SREC records from previous to current |
| CMD_GET_CURRENT_LINE | Read the current SREC record |
| CMD_GET_PREV_LINE | Read the previous SREC record |
| CMD_REPLY_LINE | Reply to SREC record request |

#### 4.1.3. Storage Organization

XNP stores the program image in a contiguous memory chunk in the external flash memory. Fixed Block Comparison scheme extends this by allocating two memory chunks, one for the previous program image and the other for the scratch space where the current image will be built (Figure 4(b)).

The two memory chunks have the same structure and they are swapped once the newly built program is loaded onto program memory. The current program image is now considered the previous image and the space for the previous image is available for the next version of program image. For the two memory chunks, two base address variables are maintained in the flash memory. By changing the address values in these variables, the two memory chunks can be swapped.

This memory organization has an advantage that it provides the same view of the memory as XNP and minimizes the effort of rewriting the boot loader code. The boot loader code of XNP reads the program code assuming that it is located at a fixed location in external flash memory. We modified the boot loader so that it reads the program code from the base address passed by an inter-process call argument. Thus, the boot loader can read the program code from any memory chunk depending on the base address value passed by the network program module.

However, this scheme does not use the external flash memory space economically. It allocates 256 KB of space regardless of the program code size (128 KB of space both for the current and the previous image). This accounts for 50% of the flash memory space of a MICA2 mote and leaves less space for data logging.

#### 4.1.4. Image Rebuild

The program image is built in a straightforward way. The

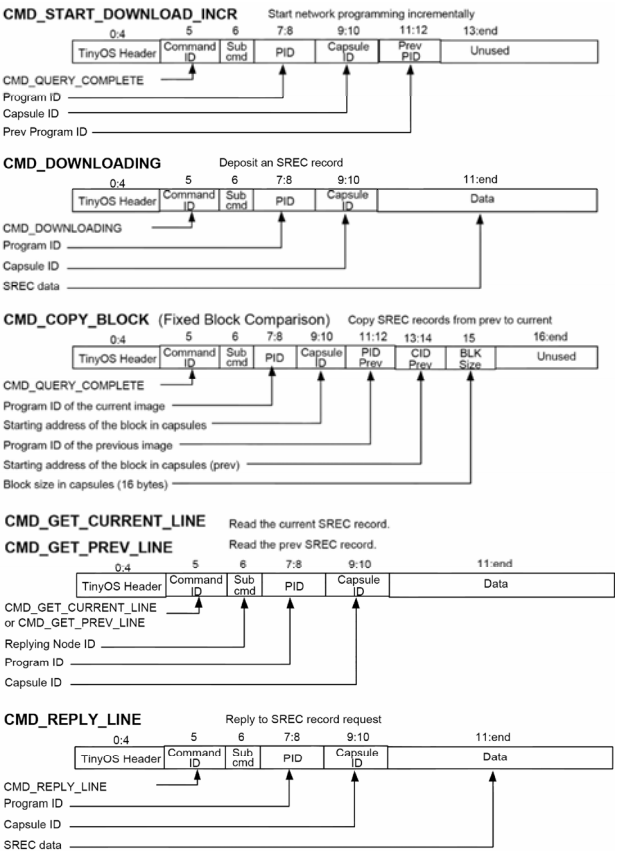



**Figure 5. Message format for incremental network prgramming.**

Copyright © 2009 SciRes. 

network programming module of the sensor node builds the program image by writing the SREC records based on a list of download and copy messages (Figure 4(c)).

The download message makes the sensor node deposit the data bytes from the message into the program image. The format of a download message is the same as an XNP download message. The capsule ID field specifies the location (line number) in the current program image and the data represents the data bytes to be written.

The copy message is for incremental network programming making the sensor node copy the SREC lines of a block in the previous program image to the current program image. The capsule ID field specifies the location of the first SREC record to be copied and the block size field specifies the number of SREC records to be copied.

## 4.2. Implementation

### 4.2.1. Difference Generation and Code Delivery

The host program, which is in charge of program image loading, difference generation and code delivery, is composed of the following classes:

- xnp: GUI, main module
- xnpUtil: loads the program image, generates the difference and provides utility functions
- xnpQry: processes queries and retransmissions
- xnpXmitCode: processes code delivery
- xnpMsg: defines the message structure
- MoteMsgIF: abstracts the interface to the serial forwarder

If the user selects the download command after loading the current and the previous program images, the xnp class spawns the xnpXmitCode class. xnpXmitCode compares each pair of blocks in the current and previous images by calling xnpUtil.CompareBlocks. Depending on the result, it either sends a copy message (CMD_COPY_BLOCK) or sends a download message (CMD_DOWNLOADING) for each line of the current block. Figure 6 illustrates this process.

### 4.2.2. Handling the Message

The network programming module for a sensor node is composed of the following components: XnpM.nc (implementation), XnpC.nc (configuration), Xnp.nc (interface), Xnp.h, XnpConst.h (constant definition). The implementation module has an event driven structure (Figure 7). When a XNP message arrives, ReceiveMsg. receive() sets the next state variable (cNextState) as the appropriate value and posts the NPX_STATEMACHINE() task. This message loop structure readily processes

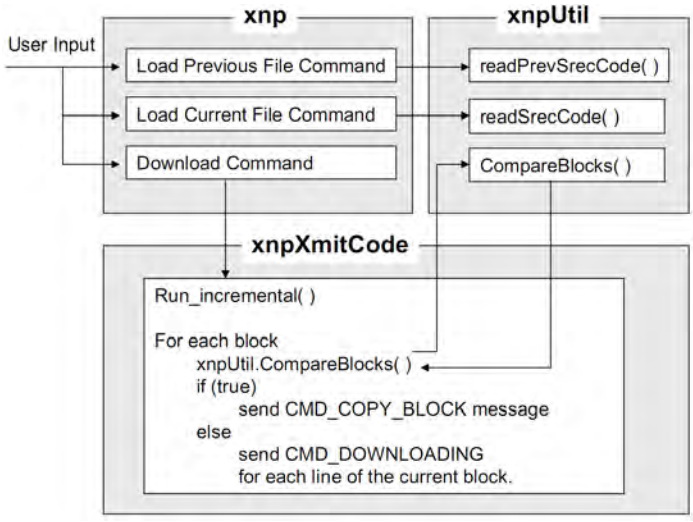



**Figure 6. Host program for incremental network programming.**

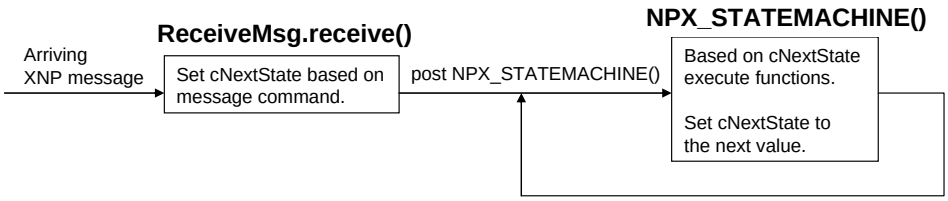



**Figure 7. Network programming module message handling.**

Copyright © 2009 SciRes. 

an incoming message without interrupting the message currently being processed.

One of the difficult parts was handling split phase operations like external flash reads and writes. To read an SREC record from external flash, EEPROMRead.read() is called. But this function returns before actually reading the record. The event handler EEPROMRead.readDone() is called when the record is actually read. And we specify the next state in the event handler. This makes us use multiple intermediate states to process an incoming message. Table 11 and 12 in the Appendix show which states were used to handle each message type.

To estimate the cost of message handling, we counted the source code lines for the two most important messages, CMD_DOWNLOADING, and CMD_COPY_BLOCK. The number of lines are 136 and 153 respectively. Table 13 shows the cost at each step of the message loop.

#### 4.2.3. Calling the Boot Loader

XnpM builds the new program image based on the previous version and the difference. In order to transfer the new image to program memory, we modified the XnpM module and the boot loader. The part of the XnpM code that executes the boot loader is shown in Figure 8. wEEProgStart is passed as the starting address of the new program image in the external flash memory. Here, 0x1F800 is the starting address of the boot loader in the Atmega128 microcontroller memory map. The boot loader uses the address passed as a parameter to access the new image.

### 4.3. Experiment Setup

To evaluate the performance of this design choice, we will count the number of block or packet transmissions of the test set. We considered the following five cases as a test scenario:

```
task void NPX_ISP() {
...
wPID = ~wProgramID; //inverted prog id
__asm__ __volatile__
("movw r20,%0" "\n\t"::"r" (wPID):"r20","r21");
wPID = wEEProgStart;
__asm__ __volatile__
("movw r22,%0" "\n\t"::"r" (wPID):"r22","r23");
wPID = wProgramID; //the prog id
__asm__ __volatile__
("movw r24,%0" "\n\t"::"r" (wPID):"r24","r25");
//call bootloader - it may never return...
__asm__ __volatile__
("call 0x1F800" "\n\t"::);//bootloader at 0xFC00
...
}
```

**Figure 8. Passing the starting address of the new program image to the boot loader.**

#### 4.3.1. Case 1 (Changing Constants)

This is the case with the minimum amount of change. We modified the constant in XnpBlink that represents the blinking rate of the LED. XnpBlink is an application written for demonstrating network programming. It accepts network programming and blinks the red LED. The following code segment shows the modification to this program.

#### 4.3.2. Case 2 (Modifying Implementation File)

This is a more general case of program modification. We added a few lines of code to the XnpCount program. XnpCount is a simple network programmable application. It counts a number, displays the number in its LEDs and broadcasts the number in radio packets. The following code segment shows the modification to this program.

#### 4.3.3. Case 3 (Major Change)

In this case, we used two programs, XnpCount and XnpBlink as input to generate the difference. The difference is larger than the first two cases, but these two applications still share a large portion of the source level code (Table 2).

**Table 2. Code size of test applications.**

| | XnpBlink | XnpCount |
|---|---|---|
| # of source code lines for network programming modules | 2049 | 2049 |
| # of source code lines for application specific modules | 157 | 198 |
| # SREC lines | 1139 | 1166 |

```
command result_t StdControl.start() {
// Start a repeating timer that fires every
1000ms.
// This period can be changed with different
value.
return call Timer.start(TIMER_REPEAT, 1000);
}
```

**(a) Case 1: Changing constants.**

```
event result_t Xnp.NPX_DOWNLOAD_DONE(
uint16_t wProgramID,
uint8_t bRet,uint16_t wEENofP){
if (bRet == TRUE)
  call CntControl.start();
else // can be deleted
  call CntControl.stop(); // can be deleted
return SUCCESS;
}
```

**(b) Case 2: Modifying implementation file.**

Copyright © 2009 SciRes. 

```
configuration XnpCount {
}
implementation {
  components Main, Counter, /* IntToLeds,*/
  IntToRfm, TimerC, XnpCountM, XnpC;
  ...
  // Main.StdControl -> IntToLeds.StdControl;
  // IntToLeds <- Counter.IntOutput;
  ...
}
```

**(c) Case 4: Modifying configuration file (commenting out IntToLeds).**

```
configuration XnpCount {
}
implementation {
  components Main, Counter, IntToLeds,
  /* IntToRfm,*/ TimerC, XnpCountM, XnpC;
  ...
  // Main.StdControl -> IntToRfm.StdControl;
  // Counter.IntOutput -> IntToRfm;
  ...
}
```

**(d) Case 5: Modifying configuration file (commenting out IntToRfm).**

**Figure 9. Test scenarios.**

#### 4.3.4. Case 4 (Modifying Configuration Filecommenting out IntToLeds)

We commented out a few lines in the XnpCount program so that we do not use the IntToLeds module. IntToLeds is a simple module that takes an integer input and displays it on the LEDs of the sensor node. The following code segment shows the modification to this program.

#### 4.3.5. Case 5 (Modifying Configuration Filecommenting out IntToRfm)

We commented out a few lines in XnpCount program so that we do not use the IntToRfm module. IntToRfm takes an integer input and transmits it over radio. Since commenting out IntToRfm forces the radio stack components not to be used, we expect a larger change in the program image than commenting out the IntToLeds module.

### 4.4. Results

To evaluate the performance of Fixed Block Comparison, we estimated the transmission time for each scenario. The host program calculates the estimated transmission time by counting how many download and copy messages it has sent. If it takes $t_{down}$ to send a download message and $t_{copy}$ to send a copy message, then the transmission time for Fixed Block Comparison, $T$, can be calculated as follows:

$$T = L_{down} \cdot t_{down} + N_{copy} \cdot t_{copy}$$

where $L_{down}$ is the number of SREC lines sent by download messages and $N_{copy}$ is the number of copy messages. As a baseline for comparison, we can also calculate the transmission time for non-incremental delivery as follows:

$$T_{xnp} = L_{down} \cdot t_{down} + L_{copy} \cdot t_{down}$$

where $L_{copy}$ is the number of SREC lines to be copied by a copy message. We found values for $t_{down}$ and $t_{copy}$ after a number of trials. We set them as 120 ms and 300 ms respectively. Table 3 shows the parameters used for estimating the performance.

Next, we measured the transmission time by reading the system clock values. Table 4 shows the estimation and measurement data.

**Table 3. Parameters for performance evaluation.**

| Parameter | Description |
|---|---|
| $t_{down}$ | Time to send a download message |
| $t_{copy}$ | Time to send a copy message |
| $L_{down}$ | Number of SREC lines sent by download message |
| $L_{copy}$ | Number of SREC lines transferred by copy message |
| $N_{copy}$ | Number of copy messages |
| $T$ | Transmission time of Fixed Block Comparison |
| $T_{xnp}$ | Transmission time of non-incremental delivery |

**Table 4. Transmission time for each case.**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.9KB | 50.1KB | 50.1KB | 49.7KB | 49.6KB |
| #-SRECs | 1139 | 1167 | 1167 | 1156 | 1155 |
| $L_{down}$ | 19 | 911 | 1135 | 1124 | 1123 |
| $L_{copy}$ | 1120 | 256 | 32 | 32 | 32 |
| $N_{copy}$ | 70 | 16 | 2 | 2 | 2 |
| **Estimation** | | | | | |
| $T$ | 23.3s | 114.1s | 136.8s | 135.5s | 135.4s |
| $T_{xnp}$ | 136.7s | 138.7s | 140.0s | 138.7s | 138.6s |
| Speed-up $T_{xnp} / T$ | 5.87 | 1.22 | 1.02 | 1.02 | 1.02 |
| **Measurement** | | | | | |
| $T$ | 25.1s | 124.4s | 149.0s | 147.1s | 146.8s |
| $T_{xnp}$ | 149.9s | 153.0s | 153.0s | 150.5s | 150.5s |
| Speed-up $T_{xnp} / T$ | 5.97 | 1.23 | 1.03 | 1.02 | 1.02 |

Copyright © 2009 SciRes. 

**Table 5. Level of code sharing in blocks, lines and bytes.**

| | Case 1 | Case 2 | Case 3 |
|---|---|---|---|
| Blocks | 97.2% | 21.9% | 2.7% |
| SREC lines | 98.3% | 40.8% | 12.0% |
| Bytes | 100.0% | 98.3% | 90.5% |

In Case 1, the difference between the two program images is small. Most SREC lines (1120 out of 1139) are transferred by copy messages and the speed-up ($T_{xnp} / T$) is about 5.9.

In Case 2, where we added a few lines to the source code, we find that less than a quarter of the SREC lines are transferred by copy messages (256 out of 1167) and the speed-up is 1.2.

In Case 3, only 32 out of 1167 lines are transferred by copy messages and the speed-up is about 1.03. Although XnpBlink and XnpCount share much at the source code level, they share little at the binary code level. The main reason is that XnpCount uses the radio stack components while XnpBlink does not. The radio stack is one of the most important modules in TinyOS, and it takes a large number of source code lines.

In Case 4 and 5, where we commented out the IntToLeds and the IntToRfm components in the configuration file XnpCount.nc, we find that only a small number of lines are transferred by copy messages and the speed-up is very small (1.02 for each case).

Fixed block comparison was not so effective for incremental network programming. It works well when the program structure doesn't change (Case 1). But, the level of sharing was low when we added a few lines of code (Case 2), which we think is a more general case of program modification.

We want to see why we have such a small level of binary code sharing. Does the program code completely change after the source modification, or does the program code still have much similarity at the byte level? To investigate further, we compared the program code at different levels: blocks (Fixed Block Comparison), SREC lines and bytes.

To compare the program code in SREC lines, we used the UNIX diff command. diff takes two ASCII files and describes how one file can be transformed to the other. To compare the program code at the byte level, we extracted the data bytes from an SREC file and stored each data byte in a line of the temporary file. We then used the UNIX diff to find the difference between the two byte list files.

Table 5 shows that Case 2 and Case 3 have a much higher level of sharing at the byte level than at the block level. For Case 2, most of the binary code was similar at the byte level (98.3%) while a small number of blocks were shared at the block level (21.9%). This implies that modifying the source code shifts the binary program code, but the program code bytes are still preserved. We can think of two ways to address this problem.

One approach is to place the shared code at a fixed location in the binary code with the help of the compiler. We can insert compiler directives and inline function calls. Then, the compiler recognizes the network programming module and determines its location in topological order.

Another approach is to utilize code sharing without modifying the code. As Table 5 suggests, much of the binary code is shared at byte level. By comparing the two binary images with a variable size boundary like Rsync [3] and LBFS [14], we can find more chances of code sharing.

## 5. Optimizing Difference Generation

Fixed Block Comparison, our first design choice for incremental network programming, was not effective in reducing data transmission traffic. It worked well only when the modified program image had the same structure as the previous program image. When additional lines are inserted into the source code, the program image is shifted and does not match the previous program image at the fixed sized block boundary.

In this section, we use the Rsync algorithm to generate the difference and rebuild the program image. The Rsync algorithm was originally made for efficient binary data update in a low bandwidth computer network. We expect the Rsync algorithm to find more matching blocks than the fixed block comparison because it compares the program image block at an arbitrary position.

### 5.1. Design

#### 5.1.1. Difference Generation

The host program generates the difference using the Rsync algorithm as in Figure 10(a).

1) The Rsync algorithm calculates a checksum pair (checksum, hash) for each fixed sized block (e.g. B bytes) of the previous program image. And the checksum pair is inserted into a lookup table.

2) Rsync reads the current program image and calculates the checksum for the B byte block at each byte. If it finds a matching checksum in the lookup table, Rsync calculates the hash for the block and compares it with the corresponding entry in the table. If the hash also matches, the block is considered a matching block.

3) Rsync moves to the next byte for comparison if the block doesn't have a matching checksum or hash. A region of bytes that doesn't have any matching blocks is tagged as a non-matching block and needs to be sent explicitly for rebuilding.

Copyright © 2009 SciRes. 

Figure 10(a) illustrates how the Rsync algorithm captures a matching block. Suppose there is a shift by a modification operation in the middle of the program image. Rsync forms a B byte window and calculates the hash for it. If the modified bytes are different from any blocks in the previous program image, there is a high probability that the hash of the modified bytes won't match any hash table entry. Rsync moves the window one byte at a time and calculates the checksum for any possible match. It doesn't match until Rsync starts to read unmodified blocks. At this moment, Rsync has found a matching block.

### 5.1.2. Program Code Storage and Rebuild

As with the case of fixed block comparison, we maintain two memory chunks in a sensor node to build the program image from the previous program image and the difference. The difference consists of a list of matching and non-matching blocks.

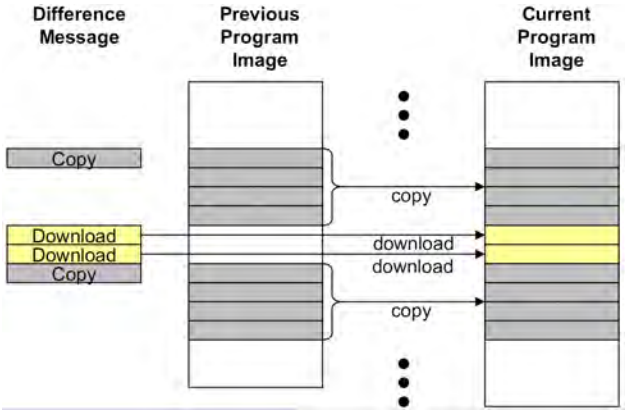
The host program sends a CMD_COPY_BLOCK message for each matching block in the difference. After hearing the message, the sensor node copies the block from the previous image to the current image. The block size of a copy message is a multiple of a SREC line and the sensor node copies each SREC line iteratively. Since the block from the previous image can be mapped to any location in the current image, the offset address field of the SREC record needs be modified (Figure 10(b)).

For each non-matching block in the difference, the host program sends one or more download (CMD_DOWNLOADING) messages. When a non-matching block is bigger than a single SREC record (16 bytes), the block is divided into multiple fragments and each fragment is sent in a download message. The data bytes of a download message can be shorter than a full SREC record if the non-matching block is not a multiple of 16 bytes. The host program does not fill the remaining bytes. This is to avoid extra flash memory accesses although the resulting program image can have a different layout from the original program image (Figure 10(c)).

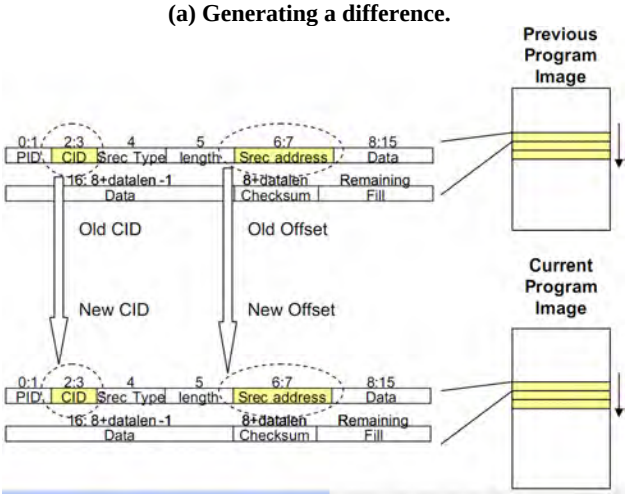
Unlike fixed block comparison, we use the base and current program version to generate the program code incrementally. If we rebuild the current program image by comparing the last version and the current version, the host program and the sensor node may have different code leading to an incorrect program build. Instead, we compare the base and the current program version. This ensures that the sensor node reads the same data bytes as the host program.

### 5.1.3. Operations

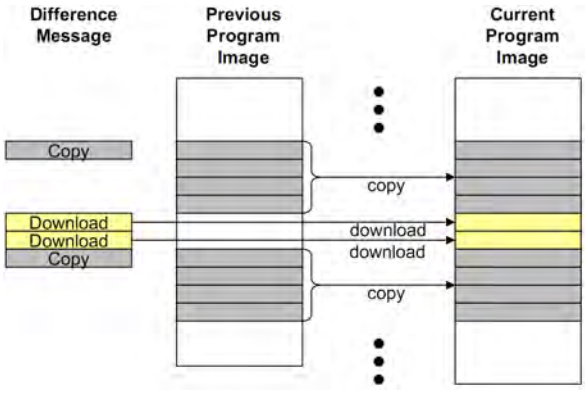
We modified the format of CMD_COPY_BLOCK to specify the starting byte address of each copy block (Figure 11). When the Rsync algorithm generates the difference, the starting byte address of each block may not be a multiple of the SREC record size. We need to specify the starting byte address as well as the CID to correctly copy SREC records.




**(a) Generating a difference.**




**(b) Copying a matching block**




**(c) Downloading a non-matching block.**

**Figure 10. Steps for incremental network programming with Rsync difference generation.**

Copyright © 2009 SciRes. 

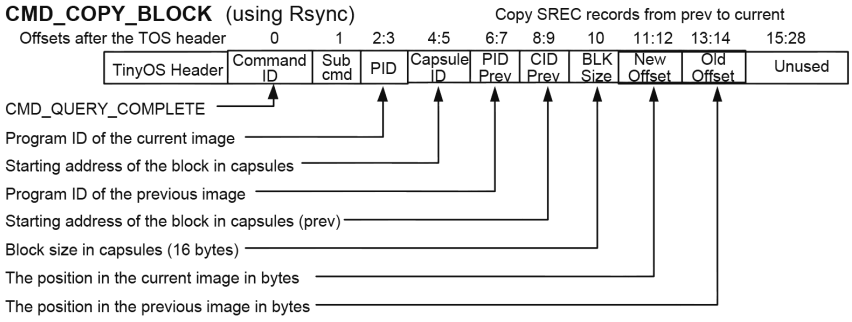



**Figure 11. Message format for incremental network programming with Rsync difference generation.**

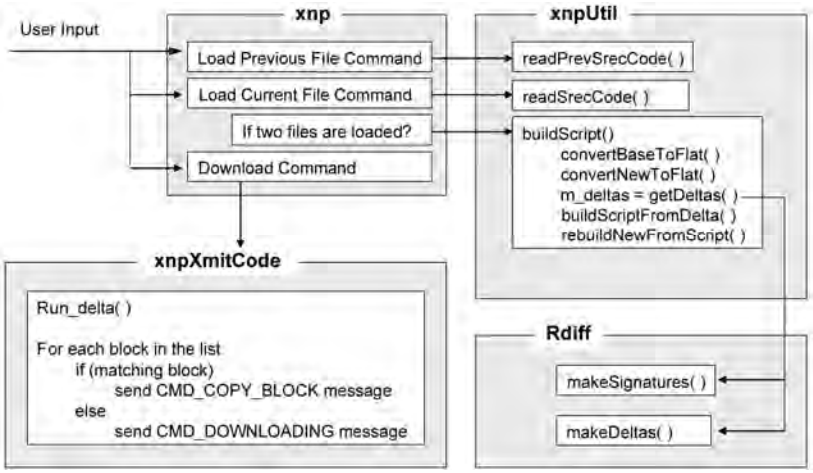



**Figure 12. Host program for Rsync difference generation.**

## 5.2. Implementation

### 5.2.1. Difference Generation

We used Jarsync [16] for the Rsync algorithm implementation. The host program calls the following methods to generate the difference: Rdiff.makeSignatures() and Rdiff.makeDeltas(). makeSignatures() calculates the checksum pair for each block in the image file and returns a list of checksum pairs. makeDeltas() compares the two image files and returns the difference as a list of matching blocks and unmatched blocks. Since these Jarsync methods assume a flat data file as input, the host program extracts only the data bytes from the SREC program image file and stores them in a temporary file before it calls the Jarsync module.

The difference returned by makeDeltas() needs postprocessing. The data bytes of an unmatched block can be an arbitrary size, but a download message can contain only up to 16 bytes. The host program divides an unmatched block into multiple blocks so that the data bytes of each block can fit in an SREC record. List entries for matching blocks are also postprocessed. Two matching blocks at consecutive locations are merged into a bigger block and this reduces the number of message transmissions.

### 5.2.2. Program Code Storage and Rebuild

The rebuilt program can be different from the original file due to the missing packets. If the host program sends a query for the missing record (CMD_GET_ CIDMISSING), the sensor node scans the current program section of external flash memory. Each record contains program ID (PID) and the capsule ID (CID, sequence number) fields. The PID should match the PID advertised at the start of incremental network programming (CMD_ START_DOWNLOAD_INCR). The CID field should match the line number where the record is written to. If either PID or CID does not match, the sensor node considers this a missing record and requests the retransmission of the SREC record. The host finds the missing record and sends it back. Then, the sensor node can fill the hole.

When the sensor node requests the retransmission of a missing SREC record, it specifies the missing record by CID field. Since the rebuilt program image can have a

Copyright © 2009 SciRes. 

**Table 6. Complexity of incremental network programming.**

| Incremental Network Programming | | Radio Stack MAC | | ADC Operation |
|---|---|---|---|---|
| Download | Copy (Rsync) | Send | Receive | Get and DataReady |
| 136 | 153 | 112 | 88 | 35 |

different layout from the original program file, just reading the specified record from the original program file does not return the correct data. To address this issue, the host program rebuilds the new program with the same layout as the program image to be built in a sensor node. The host program reads the SREC records of this image for retransmission requests.

#### 5.2.3. Code Complexity

To estimate the complexity of our implementation, we counted the source code lines in the the XnpM.nc file. A CMD_DOWNLOADING message costs 136 lines and a CMD_COPY_BLOCK message (for Rsync) costs 153 lines. The details are shown in Table 13. These numbers are comparable to those of other TinyOS modules. Sending and receiving radio packets are handled in several modules and CC1000RadioIntM.nc is a core module. A send operation takes 112 lines and a receive operation takes 88 lines in this module. As another example, we analyzed the ADCM.nc module which handles the reading of data from an ADC channel. It takes 35 lines to get a byte of data with ADCM.nc. Table 6 summarizes this.

### 5.3. Results

To evaluate the performance of incremental network programming with the Rsync algorithm, we estimated and measured the transmission time for three cases: 1) changing a constant in XnpBlink, 2) adding a few lines in XnpCount and 3) transforming XnpBlink to XnpCount. Table 7 shows the results.

In Case 1, most SREC records (1116 lines out of 1120) were transferred and the speed-up over non-incremental delivery was 6.25 (measurement). This is almost the same as the speed-up for Fixed Block Comparison (Case 1 in Figure 13).

In Case 2, 954 lines out of 1154 lines were transferred by copy messages and the speed-up over non-incremental delivery was 2.44 (measurement). Whereas Fixed Block Comparison has a speed-up of 1.2 (Case 2 in Figure 13). The improved speed-up was caused by the efficient difference generation of the Rsync algorithm.

In Case 3, the level of sharing was much smaller and the speed-up was 1.04 (measurement). We have some number of copy messages (85 messages), but they cover only a small number of blocks and are not so helpful in reducing programming time.

In Case 4, 814 lines out of 1140 lines were transferred by copy messages and the speed-up over non-incremental delivery was 1.92 (measurement). In contrast, the speed-up with Fixed Block Comparison was almost negligible (1.02).

In Case 5, 276 lines out of 1140 lines were transferred by copy messages and the speed-up over non-incremental delivery was quite small – 1.06 (measurement). Both Case 4 and Case 5 commented out a few lines in the configuration file. But, in Case 5, commenting out the IntToRfm component caused the radio stack to not be used and this changed the layout of the program image file a great deal.

**Table 7. Transmission time with the Rsync algorithm.**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.2KB | 49.4KB | 49.4KB | 48.9KB | 48.9KB |
| #-SRECs | 1120 | 1154 | 1156 | 1140 | 1147 |
| $L_{down}$ | 4 | 200 | 888 | 326 | 871 |
| $L_{copy}$ | 1116 | 954 | 278 | 814 | 276 |
| $N_{copy}$ | 72 | 104 | 85 | 107 | 83 |
| **Estimation** | | | | | |
| $T$ | 22.1s | 55.2s | 132.1s | 71.2s | 129.4s |
| $T_{xnp}$ | 134.4s | 138.5s | 139.9s | 136.8s | 137.6s |
| Speed-up $T_{xnp} / T$ | 6.09 | 2.51 | 1.06 | 1.92 | 1.06 |
| **Measurement** | | | | | |
| $T$ | 23.8s | 61.0s | 142.6s | 77.1s | 140.3s |
| $T_{xnp}$ | 148.8s | 148.9s | 148.9s | 148.2s | 148.0s |
| Speed-up $T_{xnp} / T$ | 6.25 | 2.44 | 1.04 | 1.92 | 1.05 |

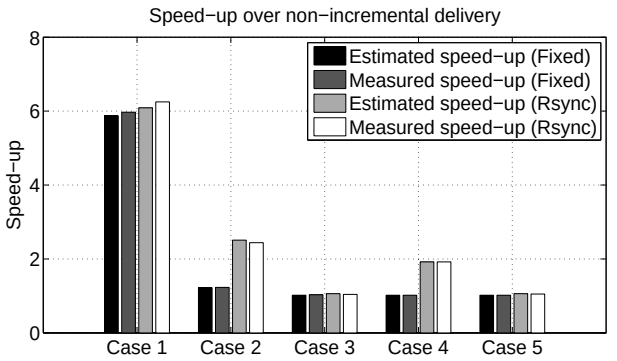



**Figure 13. Speed-up in programming time for incremental network programming with and without Rsync difference generation.**

Copyright © 2009 SciRes. 

In summary, using the Rsync algorithm achieves a speed-up of 6 for changing the constant and 2.4 for adding a few source code lines. These numbers are larger than those of Fixed Block Comparison, but using the Rsync algorithm is not still effective with a major code change.

As for the results in Table 7, we have some comments. First, we can ask why 4 SREC lines were transmitted as download messages in Case 1 when we changed only a constant in the source file. One of the reason is that the network programming module includes a timestamp value that is given at compile time. This ensures that each program image is different each time we compile the program. Another reason is that the previous SREC file was not aligned in the SREC record boundary at the end of the file. When we convert the SREC file to a flat file for Rsync, the layout changes.

Another question is why we sent 72 copy messages even though we could send fewer messages. In our design, the sensor node copies the program image blocks after hearing a copy message. To bound the execution time, we made each copy message handle up to 16 SREC lines (256 bytes).

## 6. Optimizing Difference Delivery

Compared to Fixed Block Comparison, the Rsync algorithm achieves shorter programming time by efficiently finding the shared blocks between the two binary code files. However, we can find some things to improve:

First, the network programming module transfers only a limited number of SREC records for each copy message. This is to bound the running time of a copy message so that the network programming module finishes processing a copy request before it receives another request.

Second, the network programming module interprets a copy request right after it receives the request without saving the request. In case there is a missing command, the network programming module has to check the rebuilt program image because it hasn't stored the script commands. Since the network programming module does not know whether a missing hole was caused by a missing copy message or a number of download messages, it sends a retransmission requests for each missing record from the current program image. This will take more time than retransmitting only the missing command.

Thus, we propose extending the implementation of Section 5 as follows:

1) The sensor node receives all the commands for the script.

2) The sensor node checks for any missing records in the script.

3) The sensor node starts to decode script records in response to the script decode message.

### 6.1. Design

#### 6.1.1. Operations

Since the script commands are stored in the storage space of the sensor node, we modified CMD_DOWNLOADING message to send script messages as in Figure 14. This has an advantage that we can reuse most of the code for handling normal data records to process the script commands.

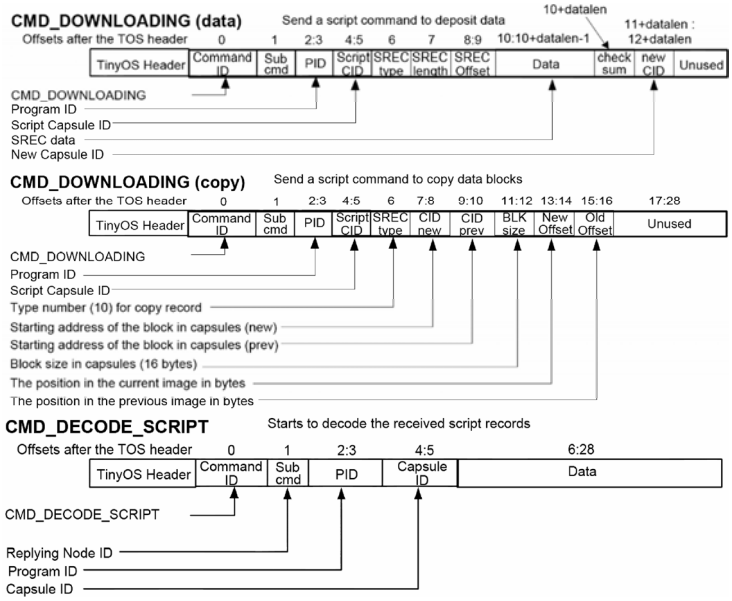
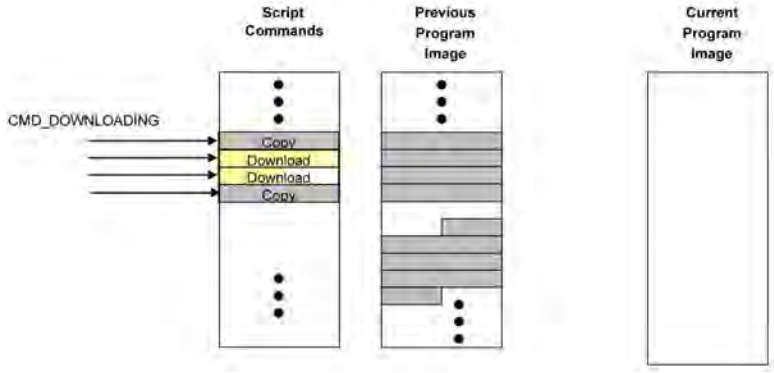
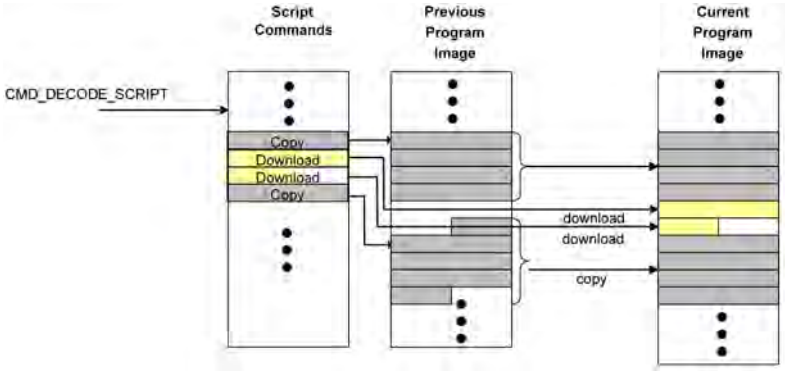



**Figure 14. Message format for incremental network programming with Rsync difference generation and decode script.**

Copyright © 2009 SciRes. 




**(a) Receiving script commands.**




**(b) Decoding script commands.**

**Figure 15. Steps for incremental network programming with Rsync difference generation and decode script.**

Message CMD_DOWNLOADING (data) has almost the same format as a normal data record download message except for the script CID and new CID fields. The script CID field is the sequence number of the command within the script and the new CID field is the location where the data record embedded in the command will be copied for building the program image.

Message CMD_DOWNLOADING (copy) is also stored in a similar way as a normal data record. A copy command has the SREC type field. This is for the Motorola SREC type and only several values are allowed by the specification (0,1,2,3,5,7,8 and 9). We extended the meaning of this field so that the value 10 represents a copy record. This allows us to store a copy command in the same manner as other data records, but can still interpret the copy command correctly. Finally, message CMD_DECODE_SCRIPT makes the network programming module start decoding the downloaded script commands.

### 6.1.2. Storage Organization and Program Rebuild

As for the storage space for the script commands, we need to choose among RAM, internal flash memory and external flash memory. RAM would be better than the others for its fast access time. However, the size of a script can be as large as a list of download messages in the worst case. Since the largest program size is 128 KB, it may not fit into RAM (4 KB) or the internal flash memory (4 KB) when the program size is large. Thus, the script should be stored in the external flash memory.

We divided the external flash memory into three sections: the previous program image, the current program image and the script sections.

At first, the host program sends the script as CMD_DOWNLOADING messages. The sensor node stores these messages in the script section if it is in the incremental network programming state. This is shown in Figure 15(a). When the host program queries any missing script commands, the sensor node scans the script section. When the difference between the two program versions is small, the traversal of the script section can finish quickly. If the sensor node finds any missing record, it requests the retransmission of the record. Then, the host program sends the record again.

After receiving the decode command from the host program, the sensor node starts rebuilding the program code. This is shown in Figure 15(b). A download command is copied from the script section to the current program image section after the CID field is modified to the new CID value. As for a copy command, the sensor node starts copying SREC records from the previous program image to the current program image. A SREC record

Copyright © 2009 SciRes. 

from the previous section is copied to the current program section after the CID and the byte offset fields are modified for the new values.

## 6.2. Results

Since a sensor node does not rebuild the program image until it receives all the script commands, we modified the metrics for the evaluation. We measured the transmission time and the decode time for the three cases. The host program saves the time stamp value when it sends a decode command and gets the next time stamp value when it receives the reply from the sensor node. The decode time is calculated as the difference of the two time stamp values. Table 8 shows the results.

**Table 8. Transmission time for incremental network programming with Rsync difference generation and decode script.**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.9KB | 50.1KB | 50.1KB | 49.7KB | 49.7KB |
| #-SRECs | 1139 | 1167 | 1167 | 1156 | 1156 |
| #-cmds | 7 | 337 | 996 | 419 | 964 |
| **Estimation** | | | | | |
| $T$ | 0.9s | 45.8s | 130.7s | 54.5s | 125.6s |
| $T_{decode}$ | 16.0s | 16.7s | 16.9s | 16.8s | 16.8s |
| $T_{xnp}$ | 154.0s | 158.5s | 158.5s | 150.7s | 150.5s |
| Speed-up $T_{xnp} / T$ | 9.10 | 2.53 | 1.07 | 2.11 | 1.06 |

**Table 9. Speed-up in programming time for three versions of incremental network programming.**

**Fixed block comparison**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.9KB | 50.1KB | 50.1KB | 49.7KB | 49.6KB |
| #-SRECs | 1139 | 1167 | 1167 | 1156 | 1155 |
| $L_{down}$ | 19 | 911 | 1135 | 1124 | 1123 |
| $L_{copy}$ | 1120 | 256 | 32 | 32 | 32 |
| $N_{copy}$ | 70 | 16 | 2 | 2 | 2 |
| $T$ | 23.3s | 114.1s | 136.8s | 135.5s | 135.4s |
| $T_{xnp}$ | 136.7s | 138.7s | 140.0s | 138.7s | 138.6s |
| Speed-up $T_{xnp} / T$ | 5.87 | 1.22 | 1.02 | 1.02 | 1.02 |

**Rsync**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.2KB | 49.4KB | 49.4KB | 48.9KB | 48.9KB |
| #-SRECs | 1120 | 1154 | 1156 | 1140 | 1147 |
| $L_{down}$ | 4 | 200 | 888 | 326 | 871 |
| $L_{copy}$ | 1116 | 954 | 278 | 814 | 276 |
| $N_{copy}$ | 72 | 104 | 85 | 107 | 83 |
| $T$ | 22.1s | 55.2s | 132.1s | 71.2s | 129.4s |
| $T_{xnp}$ | 134.4s | 138.5s | 139.9s | 136.8s | 137.6s |
| Speed-up $T_{xnp} / T$ | 6.09 | 2.51 | 1.06 | 1.92 | 1.06 |

**Rsync with split decode**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| Bytes | 48.9KB | 50.1KB | 50.1KB | 49.7KB | 49.7KB |
| #-SRECs | 1139 | 1167 | 1167 | 1156 | 1156 |
| #-cmds | 7 | 337 | 996 | 419 | 964 |
| $T$ | 0.9s | 45.8s | 130.7s | 54.5s | 125.6s |
| $T_{decode}$ | 16.0s | 16.7s | 16.9s | 16.8s | 16.8s |
| $T_{xnp}$ | 154.0s | 158.5s | 158.5s | 150.7s | 150.5s |
| Speed-up $T_{xnp} / T$ | 9.10 | 2.53 | 1.07 | 2.11 | 1.06 |

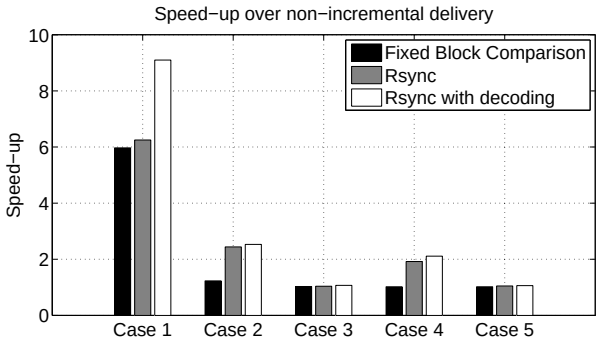



**Figure 16. Speed-up in programming time for three versions of incremental network programming.**

For Case 1, only 7 script messages were transmitted and this made the transmission time very small. The sum of transmission time and the decode time is 16.1s while non-incremental delivery took 154.0s. This gives a speed-up of 9.10. For Case 2, more script lines were transmitted (337 script messages for the 1167 line program code) and the speed-up over non-incremental delivery was 2.53. For Case 3, we sent an even larger number of script messages (996 messages for the 1167 line program code) and the speed-up was 1.07. When we

Copyright © 2009 SciRes. 

modified the configuration file, we had a similar result as Section 5. For Case 4, 419 script messages for the 1156 line program code had a speed-up of 2.11 over nonincremental delivery. For Case 5, most of the SREC records were transmitted as download script commands (964 out of 1156) and the speed-up was 1.06.

Figure 16 and Table 9 show the results of the three incremental network programming implementations: Fixed Block Comparison, Rsync and Rsync with split decode. We can find that splitting the script transmission and the program rebuild improves the overall programming time. When the source code is modified at minimum, the implementation with Rsync and split decode saved programming time by sending fewer script messages even though it has to decode the script messages. When a small number of source code lines were added, the programming time was a little better than the implementation that just uses the Rsync algorithm. For the major program change, it didn't achieve the speed-up, but it was still as good as non-incremental delivery.

We can comment on Case 3. Even though we used the Rsync algorithm and split decode, the speed-up over non-incremental delivery was negligible. This is because the difference between the two program images cannot be described with a small number of insert, copy and skip operations.

## 7. Conclusions

Network programming is a way of programming wireless sensor nodes by sending the program code over radio packets. By sending program code packets to multiple sensor nodes with a single transfer, network programming saves the programming efforts for a large sensor network. The network programming implementation in TinyOS releases 1.1 or later provides the basic capability of network programming – delivering the program code to the sensor nodes remotely. However, the network programming implementation is not optimized when part of the program code has changed. It transmits all the code bytes even though the new version of program code is only slightly different.

We extended the network programming implementation so that it reduces programming time by transmitting an incremental update rather than the entire program code. The host program generates the difference of the two program images using the Rsync algorithm and transmits the difference to the sensor nodes. Then, the sensor nodes decode the difference script and build the program image based on the previous program version and the difference script. We tested our incremental network programming implementation with some test applications. We have a speed-up of 9.1 for changing a constant and 2.1 to 2.5 for changing a few lines of code in the source code.

For future work, we plan to extend our incremental network programming for multihop delivery. One way is to use an existing multihop network programming mechanism such as Deluge [6] or MOAP [5]. In this case, we need to modify the underlying multihop delivery mechanism to be compatible with an incremental program image as well as non-incremental image. Another way is to use a generic multihop routing protocol. Since a generic routing protocol just delivers packets without storing the program image, our incremental network programmig mechanism can be easily extended for multihop delivery by replacing a single-hop send command with a multihop version.

## 8. Acknowledgements

Thanks to Crossbow Technology for providing the source code for the network programming module and the boot loader. This work is was supported by the Defense Advanced Research Projects Agency under a contract F33615-01-C1895 ("NEST"), the National Science Foundation under grants #0435454 ("NeTS-NR") and #0454432 ("CNS-CRI"), a grant from the Keck Foundation, and generous gifts from HP and Intel.

## 9. References

[1] J. Jeong, S. Kim, and A. Broad, "Network reprogramming," http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/Net-workReprogramming.pdf., 2003.

[2] Crossbow Technology. Mote in network programming user reference, http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/Xnp.pdf., 2003.

[3] A. Tridgell, "Efficient algorithms for sorting and synchronization. PhD thesis," Australian National University, Canberra, Australia, February 1999.

[4] A. Atmega, 128 microcontroller reference, http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.

[5] T. Stathopoulos, J. Heidemann, and D. Estrin, "A remote code update mechanism for wireless sensor networks, cens technical report #30," http://lecs.cs.ucla.edu/ thanos/moap-TR.pdf., 2003.

[6] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," pp. 81–94, November 2004.

[7] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," The First ACM Conference on Embedded Networked Sensor Systems (Sensys'03), 2003.

[8] N. Reijers and K. Langendoen, "Efficient code distribution in wireless sensor networks," in Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA'03), pp. 60–67, September 2003.

Copyright © 2009 SciRes. 

[9] R. Kapur, T. Yeh, and U. Lahoti, "Differential wireless reprogramming of sensor networks, ucla cs213 project report," 2003.

[10] T. Yeh, H. Yamamoto, and T. Stathopolous, "Over-theair reprogramming of wireless sensor nodes, ucla ee202a project report, http://www.cs.ucla.edu/˜tomyeh/ee202a/project/EE202a_final_writeup.doc., 2003.

[11] P. Levis and D. C. Maté, "A tiny virtual machine for sensor networks," Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'02), pp. 85–95, October 2002.

[12] P. Levis, N. Patel, S. Shenker, and D. Culler, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation (NSDI'04), pp. 2, March 2004.

[13] W. Emmerich, C. Mascolo, and A. Finkelstein, in Proceedings of the 22nd International Conference on Software Engineering, June 2000.

[14] A. Muthitacharoen, B. Chen, and D. Mazi´eres, "A low -bandwidth network file system," pp. 174–187, October 2001.

[15] G. H. Flammer, "Method for distributing program code to intelligent nodes in a wireless mesh data communication network," US Patent 5,903,566, May 1999.

[16] C. M. Jarsync, "A java implementation of the rsync algorithm," http://jarsync.sourceforge.net/.

## Appendix

**Table 10. Receiving the incoming message.**

| Message Command | Next State | Action |
|---|---|---|
| CMD_START_DOWNLOAD | SYS_DL_START | post NPX_STATEMACHINE() |
| CMD_DOWNLOADING | SYS_DL_SRECWRITE | post NPX_STATEMACHINE() |
| CMD_DOWNLOAD_COMPLETE | SYS_DL_END | post NPX_STATEMACHINE() |
| CMD_ISP_EXEC | SYS_ISP_REQ | post NPX_STATEMACHINE() |
| CMD_GET_CIDMISSING | SYS_REQ_CIDMISSING | post NPX_STATEMACHINE() |
| CMD_START_DOWNLOAD_INCR | SYS_DL_START_INCR | post NPX_STATEMACHINE() |
| CMD_COPY_BLOCK | SYS_COPY_BLOCK_PREP | post NPX_STATEMACHINE() |
| CMD_GET_CURRENT_LINE | SYS_GET_CURRENT_LINE_PREP | post NPX_STATEMACHINE() |
| CMD_GET_PREV_LINE | SYS_GET_PREV_LINE_PREP | post NPX_STATEMACHINE() |

**Table 11. NPX_STATEMACHINE() state transision.**

**Start Download**

| Current State | Next State | Action |
|---|---|---|
| SYS_DL_START | | fNPXStartDownload() signal Xnp.NPX_DOWNLOAD_REQ() |
| Call from main application Xnp.NPX_DOWNLOAD_ACK() | SYS_DL_START1 | Post NPX_STATE_MACHINE() |
| SYS_DL_START1 | SYS_DL_START2 | Call EEPROMWrite.endWrite() Post NPX_STATEMACHINE() |
| SYS_DL_START2 | SYS_ACK | Post NPX_STATEMACHINE() |

**Download End**

| Current State | Next State | Action |
|---|---|---|
| SYS_DL_END | SYS_DL_END_SIGNAL | Call EEPROMWrite.endWrite() Post NPX_STATEMACHINE() |
| SYS_DL_END_SIGNAL | SYS_ACK | Post NPX_STATEMACHINE() |

Copyright © 2009 SciRes. 

**Download**

| Current State | Next State | Action |
|---|---|---|
| SYS_DL_SRECWRITE | SYS_EEFLASH_WRITEPREP or SYS_ACK | post NPX_STATEMACHINE() |
| SYS_EEFLASH_WRITEPREP | SYS_EEFLASH_WRITE | post NPX_STATEMACHINE() |
| SYS_EEFLASH_WRITE | SYS_EEFLASH_WRITEDONE | post NPX_STATEMACHINE() |
| SYS_EEFLASH_WRITEDONE | SYS_ACK | Call EEPROMWrite.endWrite()<br>Post NPX_STATEMACHINE() |

**Idle**

| Current State | Next State | Action |
|---|---|---|
| SYS_ACK | SYS_IDLE | post NPX_STATEMACHINE() |
| SYS_IDLE | SYS_IDLE | post NPX_STATEMACHINE() |

**Retransmission**

| Current State | Next State | Action |
|---|---|---|
| SYS_REQ_CIDMISSING | SYS_GET_CIDMISSING | Call EEPROMWrite.endWrite()<br>Post NPX_STATEMACHINE() |
| SYS_GET_CIDMISSING | SYS_GETNEXTCID | post NPX_STATEMACHINE() |
| SYS_GETNEXTCID | SYS_GETNEXTCID or SYS_GETDONE | post NPX_STATEMACHINE() |
| SYS_GETDONE | SYS_IDLE | post NPX_STATEMACHINE() |

**Reprogram**

| Current State | Next State | Action |
|---|---|---|
| SYS_ISP_REQ | SYS_ISP_REQ1 | post NPX_STATEMACHINE() |
| SYS_ISP_REQ1 | SYS_ACK | post NPX_ISP() |
| SYS_DL_START_INCR | | fNPXStartDownloadIncr()<br>signal Xnp.NPX_DOWNLOAD_REQ() |

**Table 12. NPX_STATEMACHINE() state transition (added for incremental network programming).**

**Start Download**

| Current State | Next State | Action |
|---|---|---|
| SYS_DL_START_INCR | | fNPXStartDownloadIncr()<br>Signal Xnp.NPX_DOWNLOAD_REQ() |

**Copy Command**

| Current State | Next State | Action |
|---|---|---|
| SYS_COPY_BLOCK_PREP | SYS_COPY_BLOCK_READ | Call EEPROMWrite.endWrite()<br>post NPX_STATEMACHINE() |
| SYS_COPY_BLOCK_READ | SYS_EEFLASH_COPYWRITE | Call EEPROMRead.read()<br>fNPXCopyBlk()<br>post NPX_STATEMACHINE() |
| SYS_EEFLASH_COPYWRITE | SYS_EEFLASH_COPYWRITEDONE | Post NPX_wEE_LineWrite()<br>Post NPX_STATEMACHINE() |
| SYS_EEFLASH_COPYWRITEDONE | SYS_COPY_BLOCK_PREP or SYS_ACK | Post NPX_STATEMACHINE() |

**Debugging Commands**

| Current State | Next State | Action |
|---|---|---|
| SYS_GET_PREV_LINE_PREP | SYS_ACK | Call EEPROMRead.read()<br>fNPXGetLine()<br>post NPX_STATEMACHINE() |
| SYS_GET_CURRENT_LINE_PREP | SYS_ACK | Call EEPROMRead.read()<br>fNPXGetLine()<br>post NPX_STATEMACHINE() |

Copyright © 2009 SciRes. 

**Table 13. Cost of message handling.**

**CMD_DOWNLOADING**

| Step | Source Lines | Description |
|---|---|---|
| CMD_DOWNLOADING | 29 | |
| SYS_DL_SRECWRITE | 41 | |
| SYS_EEFLASH_WRITEPREP | 22 | |
| SYS_EEFLASH_WRITE | 31 | |
| SYS_EEFLASH_WRITEDONE | 13 | |
| Total | 136 | |

**CMD_COPY_BLOCK (Fixed Block Comparison)**

| Step | Source Lines | Description |
|---|---|---|
| CMD_COPY_BLOCK | 46 | |
| SYS_COPY_BLOCK_PREP | 16 | Repeated for each SREC line |
| SYS_COPY_BLOCK_READ | 40 | Repeated for each SREC line |
| SYS_EEFLASH_COPYWRITE | 29 | Repeated for each SREC line |
| SYS_EEFLASH_COPYWRITEDONE | 22 | Repeated for each SREC line |
| Total | 153 | |

**CMD_COPY_BLOCK (Rsync)**

| Step | Source Lines | Description |
|---|---|---|
| CMD_COPY_BLOCK | 46 | |
| SYS_COPY_BLOCK_PREP | 16 | Repeated for each SREC line |
| SYS_COPY_BLOCK_READ | 44 | Repeated for each SREC line |
| SYS_EEFLASH_COPYWRITE | 29 | Repeated for each SREC line |
| SYS_EEFLASH_COPYWRITEDONE | 22 | Repeated for each SREC line |
| Total | 157 | |

Copyright © 2009 SciRes. 

*Int. J. Communications, Network and System Sciences*, 2009, 5, 453-458
doi:10.4236/ijcns.2009.25049 Published Online August 2009 (http://www.SciRP.org/journal/ijcns/).

# Improved C-V Level Set Algorithm and its Application in Video Segmentation

**Jinsheng XIAO, Benshun YI, Xiaoxiao QIU**
*School of Electronic Information, Wuhan University, Wuhan, China*
*Email*: *js_xiao@tom.com*
*Received March* 23, 2009; *revised May* 15, 2009; *accepted June* 21, 2009

## ABSTRACT

Image segmentation method based on level set model has wide potential application for its excellent segmentation result. However its complex computing restricts its application in video segmentation. In order to improve the speed of image segmentation, this paper presents a new level set initialization method based on Chan-Vese level set model. After a simple iterative, we can separate out the outline of objects. Experiments show that the method is simple and efficient, with good separation effects. The improved Chan-Vese method can be applied in video segmentation.

**Keywords:** Image Segmentation, Level Set, C-V Model, Video Segmentation

## 1. Introduction

Image segmentation is intended to separate objects from the image, and the corresponding border is gained at the same time. In recent years, Researchers in the theory and technology of image segmentation has achieved fruitful research results, and active contours extraction is one of the important research results. The research of active contour can be divided into two groups: parameterized active contour based on Snake model which was proposed by Kass [1]; geometric active contour based on Level set methods which were first introduced by Osher and Sethian [2] for capturing moving fronts. Level set methods overcome the weaknesses of other algorithms. Its segmentation results are not sensitive to the initial position and the topology adaptability is strong. Level set method is a powerful tool of curve evolution, which can effectively deal with cusp and has a strong ability to separate complex structure of objects.

Chan-Vese level set model (C-V model) proposed by Chan and Vese [3] was integrated with the ideal of level set and Mumford-Shah model [4]. Being different from the traditional model based on the deformation parameters and geometry active contour model, this model does not rely on the gradient of image when extracting the boundary of objects, so the images with gradient edge meaningless and ambiguous verge can get a good segmentation. However, it has a weakness like general level set model, amount of computation. At this stage, works on C-V model mainly concentrate on revising its model, such as J. Li [5], etc, through improved C-V model, upgraded the capture of outline from local to the whole image; Y. Y. Gong [6], etc, by amending the C-V model, multi-objects can be extracted based on single level set, and so on. In this paper, we propose a improved C-V model, which can greatly improve the efficiency of segmentation, and can be applied to real-time video image segmentation.

## 2. Description of C-V Model

C-V model, which is integrated with the thinking of level set and Mumford-Shah model, does not take advantage of gradient information, but minimizes the energy function to evolve curve [3]. Assume that image $I(x, y)$ is formed by two regions: objects ($C_o$) and background ($C_b$), which is separated by the evolving curve $C$ in $\Omega$. The constants $c_o, c_b$, depending on $C$, are the averages of image $I$ inside $C$ ($C_o$) and outside $C$ ($C_b$) respectively. Chan and Vese introduced the energy function $E(C, c_o, c_b)$, defined by

Copyright © 2009 SciRes. 

$$E(C,c_o,c_b)=\mu L(C)+\nu S(C)+\lambda_o\int_{C_o}|I-c_o|^2dxdy+\lambda_b\int_{C_b}|I-c_b|^2dxdy \quad (1)$$

where $L(C)$ is the length of the cure $C$, and $S(C)$ is the area of $C_o$ , $\mu,\nu\geq 0$, $\lambda_o,\lambda_b>0$ are fixed parameters. Therefore the energy function is minimized if the curve is on the boundary of the object. Optimization (1), we can get the ultimate location of segmentation line $C$, as well as the unknown $c_o$ , $c_b$ .

$$\{C^o,c_o^o,c_b^o\}=\inf_{C,c_o,c_b}E(C,c_o,c_b) \quad (2)$$

Using the Heaviside function $H(z)$, and the one-dimensional Dirac measure $\delta(z)$, and defined, respectively, by

$$H(z)=\begin{cases}1, z\geq 0\\ 0, z<0\end{cases},\quad \delta(z)=\frac{d}{dz}H(z) \quad (3)$$

Partial differential equations, gotten by Chan and Vese using Euler-Lagrange method, are as follows: (4)–(7)

$$c_o(\phi)=\frac{\int_\Omega I(x,y)H(\phi(x,y))dxdy}{\int_\Omega H(\phi(x,y))dxdy} \quad (4)$$

$$c_b(\phi)=\frac{\int_\Omega I(x,y)(1-H(\phi(x,y)))dxdy}{\int_\Omega (1-H(\phi(x,y)))dxdy} \quad (5)$$

$$\frac{\partial\phi}{\partial t}=\delta(\phi)\left[\mu div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\nu-\lambda_o(I-c_o)^2+\lambda_b(I-c_b)^2\right] \quad (6)$$

$$\phi(0,x,y)=\phi_0(x,y),\quad in\,\Omega \quad (7)$$

$$\begin{cases}H_\varepsilon(z)=\frac{1}{2}\left[1+\frac{2}{\pi}\arctan\left(\frac{z}{\varepsilon}\right)\right]\\ \delta_\varepsilon(z)=\frac{1}{\pi}\cdot\frac{\varepsilon}{\varepsilon^2+z^2}\end{cases} \quad (8)$$

In the numerical calculations, regularizing Function (8) is used to replace $H(z),\delta(z)$ respectively. So that the gradient flow Equation (6) roles in all of the level set, and we can automatically monitor the empty goal with the internal region, and make the overall energy function to the minimum.

Let's disperse the equation in $\phi$, use a finite differences implicit scheme. Recall first the usual notations: let $h$ be the space step, $\Delta t$ be the time step, and $(x_i,y_j)=(ih,jh)$ be the grid points, where $1\leq i,j\leq M$ . Let $\phi_{i,j}^n=\phi(n\Delta t,x_i,y_j)$ be an approximation of $\phi(t,x,y)$. Knowing $\phi^n$ , we can get and $c_o(\phi^n),c_b(\phi^n)$ using (4) and (5). The finite differences are

$$\Delta_\pm^x\phi_{i,j}=\pm(\phi_{i\pm1,j}-\phi_{i,j}),\quad \Delta_\pm^y\phi_{i,j}=\pm(\phi_{i,j\pm1}-\phi_{i,j})$$

Chan and Vese compute $\phi^{n+1}$ through (11).

$$L=\frac{\mu}{h}\Delta_-^x\left(\frac{\Delta_+^x\phi_{i,j}^n}{\sqrt{(\Delta_+^x\phi_{i,j}^n)^2+(\Delta_+^y\phi_{i,j}^n)^2}}\right)+\frac{\mu}{h}\Delta_-^y\left(\frac{\Delta_+^y\phi_{i,j}^n}{\sqrt{(\Delta_+^x\phi_{i,j}^n)^2+(\Delta_+^y\phi_{i,j}^n)^2}}\right) \quad (9)$$

$$R=-\nu-\lambda_o(u_{0,i,j}-c_o(\phi_{i,j}^n))^2+\lambda_b(u_{0,i,j}-c_b(\phi_{i,j}^n))^2 \quad (10)$$

$$\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^n}{\Delta t}=\delta_h(\phi_{i,j}^n)[L+R] \quad (11)$$

From the Equation (6), can be seen, the definition of partial differential equations involving image function $I(x,y)$ is domain-wide map data, and the definition of other two unknown $c_o,c_b$ is also image definition of the region, with the overall characteristics. Hence, updating level set function is in the entire defined region, the computation is large [3].

## 3. Improved C-V Model

From the above analysis, we know that C-V method has a grate calculation. If we can effectively reduce the amount of computation, C-V method can be more widely applied. In traditional level set methods, it is necessary to initialize the level set function $\Phi$ as a signed distance

Copyright © 2009 SciRes. 

function $\Phi_0$. Curve C divides the plane into internal and external regions. $\Phi(x,y)=\pm d, d$ is the distance from point (x, y) to curve C. Generally, the distance of internal and external points are negative and positive respectively and signed distance function needs to be re-initialized. C-V method generally defines the symbol distance function (SDF) as a cone, with particularly complex calculation.

Lie [8] demonstrates that the presence of signed distance function is not inevitable. Lie imposed a binary level set model. We will introduce this idea into C-V method. Define radius of the closed curve C as infinite, C represents a straight line in plane $\Omega$, which will be divided into upper regional $\Omega_u$ and lower regional $\Omega_d$. Initialization function $\phi_0$ is defined as:

$$\phi_0(x,y)=\begin{cases}-\rho_u, & (x,y)\in\Omega_u\\ \rho_d, & (x,y)\in\Omega_d\end{cases}\quad \rho_u,\rho_d>0 \qquad \textbf{(12)}$$

So first level set evolution, the curve of the internal and external simplified curve of the upper and lower regions, the calculation of $c_o, c_b$ are very simple, The calculation of difference operator is very simple too, as only points on the boundary of upper and lower region are non-zero constant, and the remaining places are zero value; The initialization level set function is fixed constants, as well as $\delta(z)$. In a linear mesh C at the point, follow the Reference [7] approach, iterative Formula (9) can be transformed as follows:

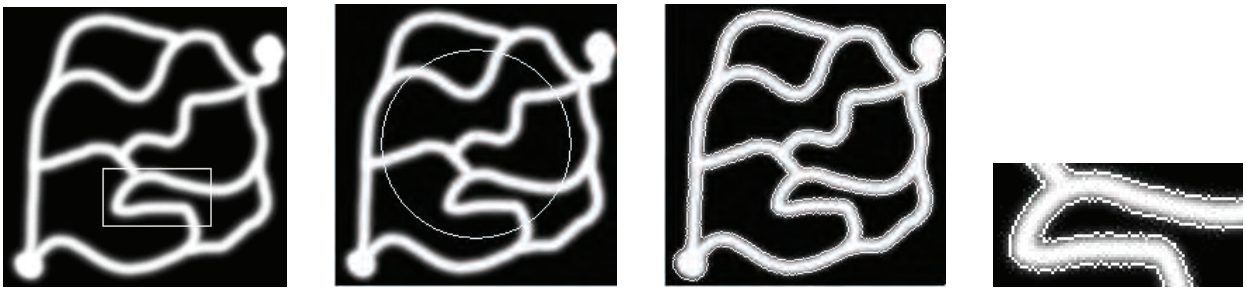
$$\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta t}=\begin{cases}\delta_h(\phi_{i,j}^n)R, & j<j_0-1, j>j_0\\ \delta_h(\phi_{i,j}^n)\left(-\dfrac{u}{h}sign(\rho_u-\rho_d)+R\right), & j=j_0\\ \delta_h(\phi_{i,j}^n)\left(\dfrac{u}{h}sign(\rho_u-\rho_d)+R\right), & j=j_0-1\end{cases} \qquad \textbf{(13)}$$

Obviously, compared with the traditional C-V method, the calculation of our method is much smaller in the first level set iterative.

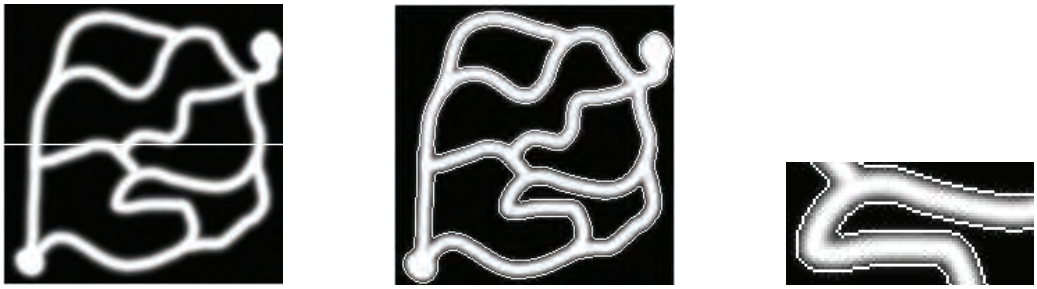
In order to ensure the level set method not departure from SDF, the time step must be very small, usually 0.1 s, which increasing the evolution of time. Since the existence of SDF is not inevitable, we appropriately increase the time step. Using larger time step can speed up the evolution, but may cause error in the boundary location if the time step is chosen too large. There is a tradeoff between choosing larger time step and accuracy in boundary location. Usually, we use $\Delta t \le 10.0$ for the most images.

We list out some of the experimental results in the following paper. In our numerical experiments, we generally choose the parameters as follows: $\lambda_o=\lambda_b=1$, $\nu=0$, $h=1$ (the space step), $\Delta t=2$ (the time step), $\varepsilon=1$.

In the first experiment, we chose the test image cavern. Jpg (Figure 1(a)), whose size is 200 × 200. The region of



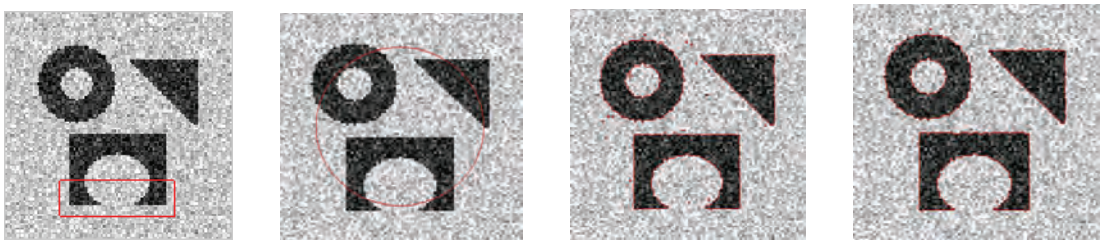
**(a). The original image.** **(b). Initialized as CV.** **(c). 15 iterations of CV.** **(d). Local enlarge of (c).**



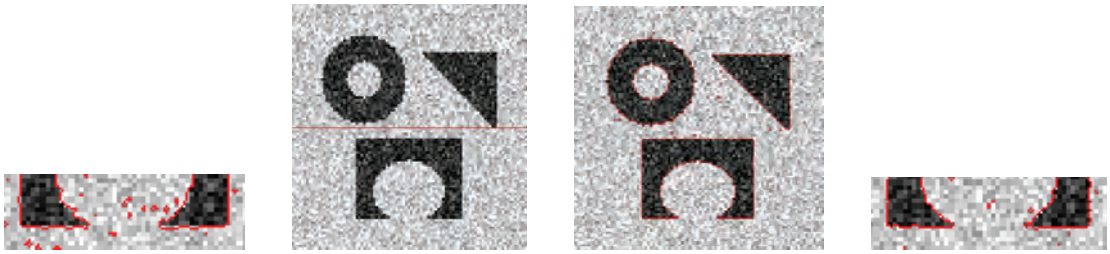
**(e). Initialized as ICV.** **(f). 1 iteration of ICV.** **(g). Local enlarge of (f).**

**Figure 1. The binary image.**

Copyright © 2009 SciRes. 



**(a). The original image.** **(b). Initialized as CV.** **(c). 20 iteration of CV.** **(d). 200 iteration of CV.**



**(e). Local enlarge of (c).** **(f). Initialized as ICV.** **(g). 20 iteration of ICV.** **(h). Local enlarge of (g).**

**Figure 2. The noisy image.**

interest is specified by the white box. It needs 15 iterative to achieve the ideal state of division when the level set function is initialized as $\phi_0(x,y)=67-\sqrt{(x-100)^2+(y-100)^2}$, and time-consuming is 1.578 s. Figures 1(b),1(c) show the traditional method of initialization(CV method) and the final result of division. With our improved method (ICV method), $\rho_u=-1,\rho_d=1$, we can get the desired effect of split after one iterative and the time-consuming is 0.016s (Figures 1(e),1(f)). To compare the segmentation results of the two methods in more detail, we show a zoomed version of the results in Figure (d) and Figure (g) for a region delineated by the white box in Figure (a). It's clear to see that we can get a smoother curve split with our approach.

In Figure 2, we show how our arithmetic and traditional C-V methods work on a noisy synthetic image. The region of interest is specified by the white box (Figure 2(a)). Bose of the two methods can automatically detect the objects. If we use traditional C-V model initialization method, $\phi_0(x,y)=33-\sqrt{(x-50)^2+(y-50)^2}$, the outline of objectives are separated after 20 times of iterative, but the right-angle region isn't well separated(Figures 2(c), 2(e)). After 200 iterative, the situation is improved. However, it spends 20 times iteration to achieve a perfect result with our method. The rectangular region is also divided perfect (Figures 2(g), 2(h)).

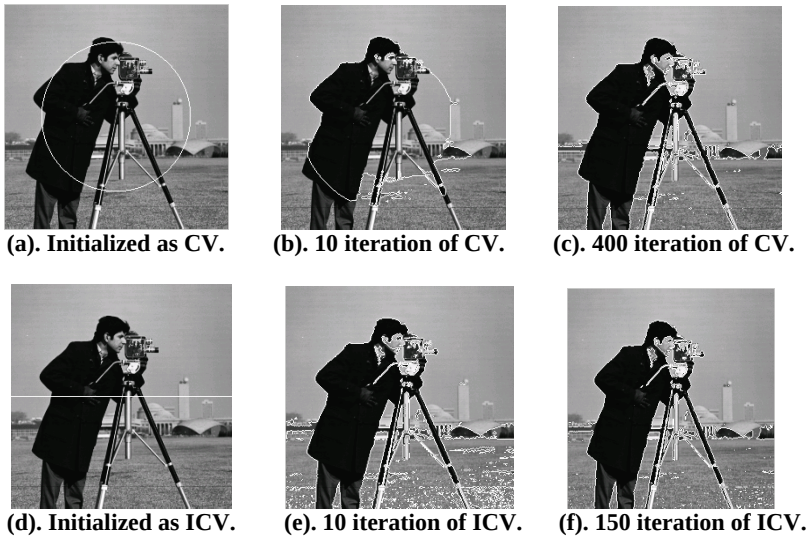
The test results of cameraman.tif are showed in Figure 3. Size of the picture is 256×256. In Figure 3(a), the level set function was initialized as signed distance function: $\phi_0(x,y)=-\sqrt{(x-128)^2+(y-128)^2}+85$. Figure 3(b) and Figure 3(c) are results of 10 iterations and 400 iterations respectively. In Figure 5(d) level set function was initialized as a linear curve. We can get the photographer's outline after only one time of iteration. But there is a lot of noise divisions on the lake, noises would gradually reduce after multiple iterations. Figure 3(e) is the result of 10 iterations. After 150 iterations, we get a stable state. We can see that the noise has been significantly reduced, but not completely filtered out.

## 4. Video Image Segmentation Based on ICV

C-V model is the key to resolve the two issues separate, and categories of backgrounds and objectives in the video images are unknown. C-V model can not be directly applied to the whole map. So we get the movement area by analyzing the characteristics of H.264 codec first of all, and then the improved C-V model is applied to the movement region which was detected. This article focuses on the moving target in the same scene, the video image processing and experimental data are based on a single static camera, so the background image is static. According the principles of H.264 video codec, motion vector of macro block as a background is usually zero value, and only motion vector of macro block in the movement region is non-zero, so we can get the moving region through the distribution of motion vector.

When inter coding being chosen, the coding frame need to use the frame before (reference frame) for movement searching, and then motion vector of each block is get. We illustrate in Figure 4 the above remarks. In Figure 4(a), the large box represents 16 × 16 macro block, and the small box represents 16 × 8 or 8 × 16 block, black line is motion vector. By the vector distribu-

Copyright © 2009 SciRes. 



**(a). Initialized as CV.** **(b). 10 iteration of CV.** **(c). 400 iteration of CV.**

**(d). Initialized as ICV.** **(e). 10 iteration of ICV.** **(f). 150 iteration of ICV.**

**Figure 3. The complex background image.**

**(a). Distribution of vector.** **(b). Regional campaign.** **(c). Separated results.**

**Figure 4. Moving target separated.**

tion map, we can frame the regional campaign, as shown in Figure 4(b).

Figure 4(c) is a real-time video segmentation results map, target detection of the system and tracking algorithm and real-time are tested respectively. We choose the parameters as follows: $\nu = 0$ , $\lambda_o = \lambda_b = 1$ , $\mu = 0.01 \times 255 \times 255$ , $\varepsilon = 1$ , $\Delta t = 2$ , $h = 1$ , $\rho_u = -1, \rho_d = 1$ . The regional campaign is extracted, and the moving target is tracked very well. In Figure 4(c), however, we can see that the outline of some of the goal in frame is not very ideal, sometimes the background is also included. It is because the delimitation of regional campaign is mainly based on motion estimation which processed by video coding. If light is changed, or other factors, the reference block selecting is inaccurate in motion estimation, resulting in the coding block is mistakenly believed as campaign block, so the error division is produced. The brightness weight of hands is very close to the brightness weight of the wall. As target tracking algorithm uses the brightness information of pixel, when the goal and background have similar brightness information, the algorithm will get the wrong track.

H.264 codec set the frame rate of 25 fps. The size of the regional movement we get by the motion vector information of decoder is an important factor of computing time, which affect the moving target detection and tracking module. Larger the regional campaign is, more time the target detection and tracking module need. We have real-time measured the running time of the module, which is in 15−19 ms range. When the ICV model is evolved in the whole frame, the time-consuming is not more than 20 ms, fully meet the real-time requirements.

## 5. Conclusions

In this paper, we have improved the image segmentation efficiency based on the C-V model from initialization and algorithms simplifying. The improved C-V model guarantees the division in effect while greatly improves the efficiency of the division. We apply it to the real-time H.264 video codec system. The experimental results show that, for the image of simple background, it can partition the outline of objects with dramatic speed and

Copyright © 2009 SciRes. 

high efficiency segmentation, using the methods proposed in this paper. However, for the complex background image, the outline of objects can be accurately divided, but there will be some regional background mistakenly separated. How to filter this noise fast is the next step that we need to improve. This article improves the C-V method to separate video images, which can be used to real-time detect and track the moving targets.

## 6. References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models [J]," International Journal of Computer Vision, Vol. 1, No. 4, pp. 321−331, 1987.

[2] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations [J]," Journal of Computational Physics, Vol. 79, pp. 12−49. 1988.

[3] F. T. Chan and L. Vese, "Active contours without edges [J]," IEEE Transaction Image Processing, Vol. 10, No. 2, pp. 266−277, 2001.

[4] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems [J]," Communication of Pure Applied Mathematics, Vol. 42, No. 5, pp. 577−685, 1989.

[5] J. Li, X. Yang, and P. F. Shi, "A fast level set approach to image segmentation based on mumford-shah model [J]," Chinese Journal of Computers, Vol. 25, No. 11, pp. 1175 −1183. 2002.

[6] Y. Y. Gong, X. N. Luo, H. Huang, G. J. Liao, and Y. Zhang, "Multi-objects extracted based on single level set [J]," Chinese Journal of Computers, Vol. 30, No. 1, pp. 120−128, 2007.

[7] J. S. Xiao, H. Feng, and B. S. Yi, "Finite difference method for semilinear parabolic differential inclusions [J]," Journal of Wuhan University, Natural Sciences Edition, Vol. 52, No. 3, pp. 262−266, 2006.

[8] J. Lie, M. Lysaker, and X. C. Tai, "A binary level set model and some applications to mumford-shah image segmentation," IEEE Transactions on Image Processing, Vol. 15, No. 5, pp. 1171−1181, 2006.

Copyright © 2009 SciRes. 

# 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.

### Wireless Communications

- B3G and 4G Technologies
- MIMO and OFDM
- UWB
- Cognitive Radio
- Coding, Detection and Modulation
- Signal Processing
- Channel Model and Characterization
- Antenna and Circuit

### IMPORTANT DATES

| | |
|---|---|
| Paper due: | March 10, 2010 |
| Acceptance Notification: | May 10, 2010 |
| Camera-ready due: | May 31, 2010 |

### Network Technologies

- Ad hoc and Mesh Networks
- 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

# Call for Papers

International Journal of

# Communications, Network and System Sciences (IJCNS)

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)

http://www.scirp.org/journal/ijcns/

IJCNS is an international refereed journal dedicated to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas.

## Editors-in-Chief

| | |
|---|---|
| Prof. Huaibei Zhou | Advanced Research Center for Sci. & Tech., Wuhan University, China |
| Prof. Tom Hou | Department of Electrical and Computer Engineering, Virginia Tech., USA |

## Subject Coverage

This journal invites original research and review papers that address the following issues in wireless communications and networks. Topics of interest include, but are not limited to:

| | |
|---|---|
| MIMO and OFDM technologies | Sensor networks |
| UWB technologies | Ad Hoc and mesh networks |
| Wave propagation and antenna design | Network protocol, QoS and congestion control |
| Signal processing and channel modeling | Efficient MAC and resource management protocols |
| Coding, detection and modulation | Simulation and optimization tools |
| 3G and 4G technologies | Network security |

We are also interested in:

- Short reports—Discussion corner of the journal:
  2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data.
- Book reviews—Comments and critiques.

## 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. For more details about the submissions, please access the website.

## Website and E-Mail

http://www.scirp.org/journal/ijcns ijcns@scirp.org

*Int. J. Communications, Network and System Sciences,* **2009, 5, 325-459**
Published Online August 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

# TABLE OF CONTENTS

**Volume 2 Number 5** **August 2009**



Copyright©2009 SciRes 

ISSN: 1913-3715
9771913371005 09