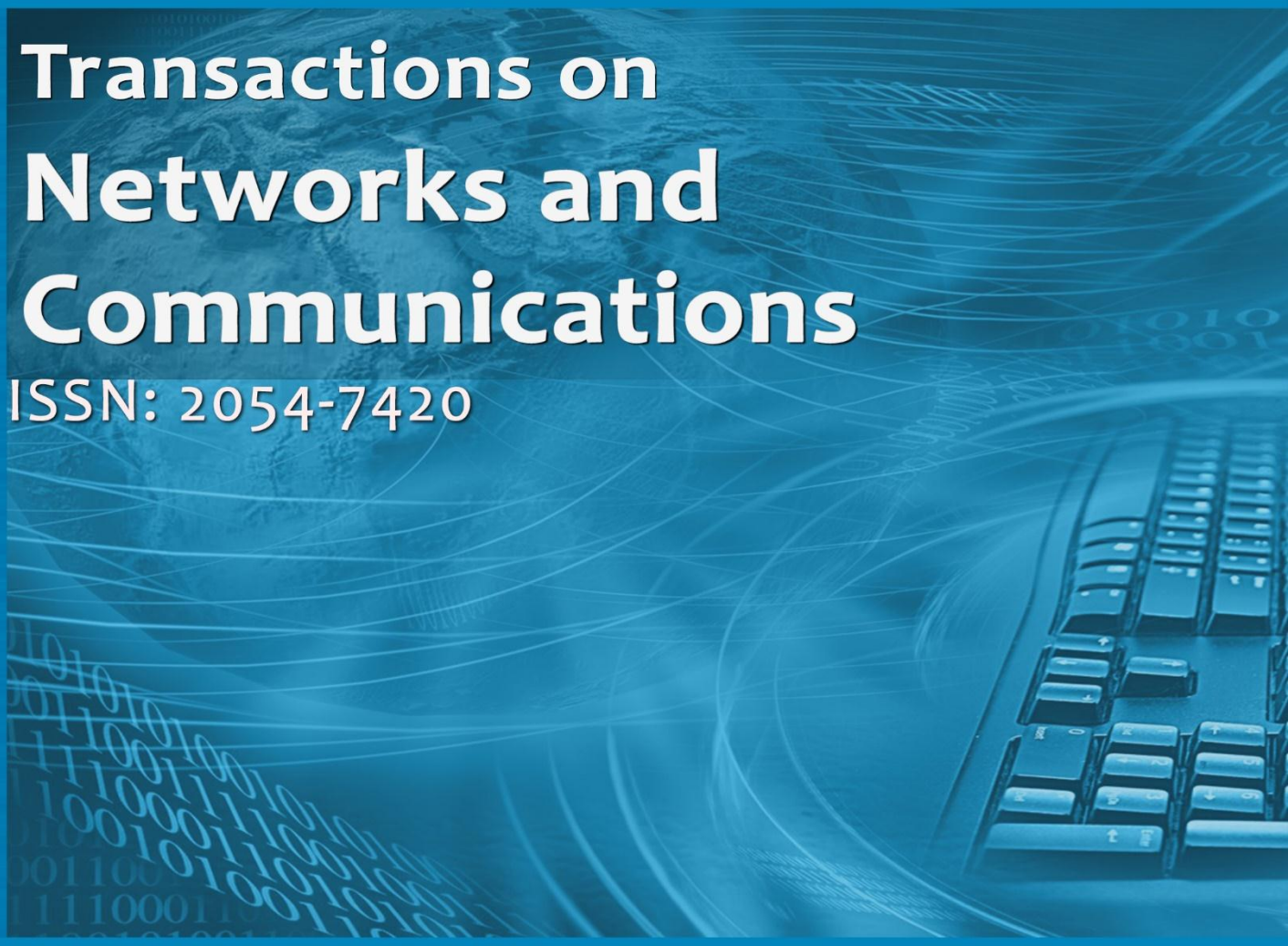
Volume 3. Issue 4 - August 2015

Transactions on Networks and Communications

ISSN: 2054-7420



Society for Sciences and Education - United Kingdom

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

A Survey on LDPC Codes for Cooperative Communications

Phuong T. Tran
Department of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam;
tranthanhphuong@tdt.edu.vn

ABSTRACT

LDPC codes are being considered as the codes that can approach the theoretical capacity limit while are not so complicated to implement. In this survey paper, the application of LDPC codes in decode-and-forward cooperative communications is investigated. Most of current researches focus on how to implement the LDPC-coded cooperation effectively. Here we consider two approaches: the first one uses factor graph decoupling and the other uses bilayer LDPC codes. The performance analysis of these schemes is carried out by density evolution and by EXIT chart analysis. In addition, the methods to design the optimal LDPC codes for these systems are also introduced. The numerical results show that we can reach very close to the capacity limit of the relay channels. Finally, simulation results of a new iterative decoding technique for LDPC codes are also presented.

Keywords: LDPC; Cooperative communication; factor graph; density evolution; Decode-and-Forward.

1 Introduction

Idea about cooperative communications started from the work of Cover and El Gamal in 1979 [2], and then it is described more rigorously in some papers starting from 2003 ([3]– [6]). A concise tutorial about cooperative communications can be found in [1]. Briefly speaking, in cooperative communication systems, each wireless user is assumed to transmit data as well as act as a cooperative agent for another user. The data from each user can reach the base station (BTS) by at least two ways: direct transmission to BTS and relayed transmission via another user [1].

The capacities of cooperative networks has been studied rigorously in some works such as [2], [7]. However, how to design the coding schemes that can approach these capacities and are not complicated to implement is still a challenging problem and is attracting the interest of many scientists. LDPC codes are promising candidates for this application. Invented by Gallagher in 1963 [8], they were almost forgotten for nearly 30 years before being rediscovered by Mac Kay in mid 90s and enhanced to irregular LDPC codes by Richardson et. al. in 2003 [9]. Since then, there has been a lot of studies about designing effective LDPC decoding methods, design optimal LDPC codes, as well as using LDPC codes in different communication systems.

Factor graph is a visualization technique used to model coding schemes [10]. In [11], an efficient implementation of LDPC codes for single-relay channel is proposed. In this scheme, the transmission of information from the source occurs in B blocks of equal length N. The factor graph of B-block transmission is then decoupled into the partial factor graphs, each of which is corresponding to a 2-block transmission. The LDPC codes for each partial factor graphs are then designed by methods proposed in [19]. By formulating the cooperative operations as equivalent SISO or MIMO systems, and using Gaussian approximation for AWGN channels, the author derive a joint relay and destination

DOI: 10.14738/tnc.34.1246
Publication Date: 15th July 2015
URL: http://dx.doi.org/10.14738/tnc.34.1246

optimization framework and develop the algorithms to solve these problems. The analysis and simulation of the performance of this LDPC-coded cooperative system will be demonstrated in this paper.

Another approach was developed from the Cover and Gamal's paper [2] is the concept of parity forwarding [18]. Based on the observation that the LDPC code designed for the relay systems is working at two different channel SNRs: a higher SNR at the relay and a lower SNR at the destination, a novel embedded LDPC code construction, namely, the bilayer LDPC code, is proposed [20]. More specifically, two new ensembles of LDPC codes, bilayer-expurgated codes and bilayer-lengthened LDPC codes, are proposed to simultaneously approach the capacities of two Gaussian channels at two different SNRs.

In this paper, the method for analyzing the performance of the design methodologies for the bilayer LDPC code, which is basically the generalization of the density evolution for standard LDPC codes, will be presented. The simulation results will show that these LDPC codes can approach in less-than-1dB distance to the theoretical capacity of the relay channels, for both single-relay case and multiple-relay case. Furthermore, this concept can be generalized to applied for more general networks.

The remaining of this paper is structured as follows: in Section II a short overview of the cooperative communication systems and LDPC codes will be given, followed by the performance analysis of the LDPC-coded cooperative systems which use the factor graph decoupling method together with the EXIT chart analysis [19]. The design methods for bilayer LDPC codes as well as the performance analysis of these codes is presented in Section III. Section VI is the conclusion of the paper.

2 LDPC-coded Cooperative System: Performance Analysis

2.1 Relay systems

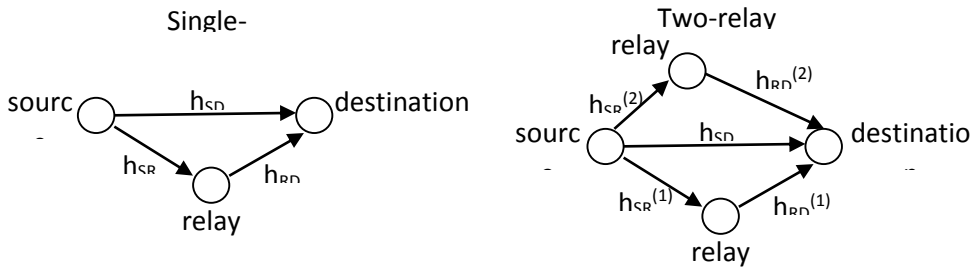


Figure 1. Diagram of single-relay system and multiple-relay system.

Basic relay systems are described by the diagrams in Figure 1. The left part is the diagram of a single-relay system and the right part is an example of multiple-relay system which two relays. These systems are modeled by the following equations:

$$y_R^{(i)} = h_{SR}^{(i)} x_S + \sum_{\substack{j=1 \\ j \neq i}}^{K} h_{j,i} x_R^{(j)} + n_R^{(i)}, 1 \leq i \leq K \tag{1}$$

$$y_D = h_{SD} x_S + \sum_{i=1}^{r} h_{RD}^{(i)} x_R^{(i)} + n_D \tag{2}$$

where x_S and $x_R^{(i)}$ are the signals transmitted from the source and from the i-th relay, respectively; $y_R^{(i)}$ and y_D are the received signal at the i-th relay and at the destination, respectively; $h_{SD}, h_{SR}^{(i)}, h_{RD}^{(i)}$, and $h_{j,i}$ are the channel gains between S and D, S and R_i, R_i and D, R_j and R_i, respectively; K is the number of active relays; $n_R^{(i)}$ and n_D are the AWGN noise at the i-th relay and at the destination, respectively. We assume the channel conditions is Rayleigh fading, i.e. the channel gains are zero-mean Gaussian distributed. Furthermore, we assume that the relays operate in full-duplex mode, and the channel distribution information (CDI) are known by the receivers, that is, the i-th relay has the knowledge of $h_{SR}^{(i)}$ and the destination has the knowledge of h_{SD} and $h_{RD}^{(i)}, i = 1, 2, ..., K$.

The cooperation protocols are described by Figure 2. In [19], two protocols are considered, both of which transmit data in blocks. There are B-1 information blocks are transmitted during B time slots.

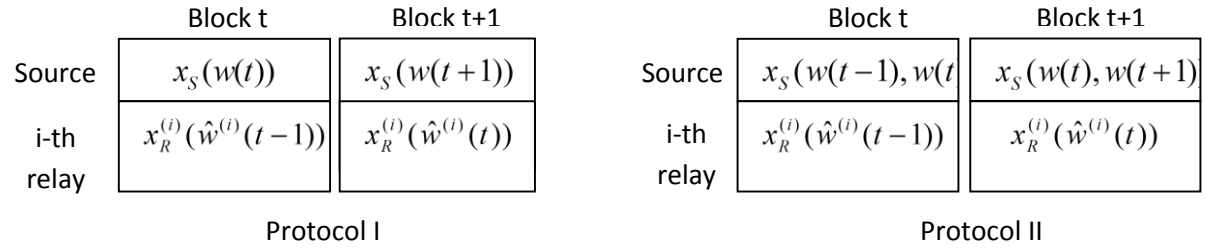


Figure 2. Cooperation protocols.

where w(t) denotes the codeword transmitted by source node during time slot t, and $\hat{w}^{(i)}(t)$ denotes the decoded codeword at the i-th relay node. For Protocol I, $x_S(w(t)) = \sqrt{P_S} w(t)$ and $x_R^{(i)}(\hat{w}^{(i)}(t)) = \sqrt{P_{R_i}} \hat{w}^{(i)}(t)$; P_S and P_{R_i} are the average transmit power per symbol at the source node and the i-th relay, respectively. For Protocol II, $x_S(w(t-1), w(t)) = \sqrt{P_{S,1}} w(t) + \sqrt{P_{S,2}} w(t-1)$ and $x_R^{(i)}(\hat{w}^{(i)}(t)) = \sqrt{P_{R_i}} \hat{w}^{(i)}(t)$; $P_{S,1}$ and $P_{S,2}$ are the average transmit power per symbol for w(t) and w(t-1) at the source node, respectively.

2.2 LPDC codes for relay systems

The performance analysis framework for the LDPC-coded cooperative systems in [19] are developed from the paper of S. ten Brink [15], in which the authors proposed a LDPC code design model for MIMO channels. In fact, the relay operations described above can be considered as a SISO system (for single-relay case) or MISO system (for multiple-relay case). Using the iterative decoding structure for MIMO systems in [15], we can set up the framework for analyzing the performance of two cooperation protocols mentioned in Section II-A. Figure 3 shows the diagram of the iterative receiver for the LDPC codes in the relay systems. The decoder consists of n/(M.Mc) individual detector nodes, each of which is connected to M.Mc variable nodes, where M is the number of inputs, Mc is the number of bits per symbol, n is the number of bits per codeword.

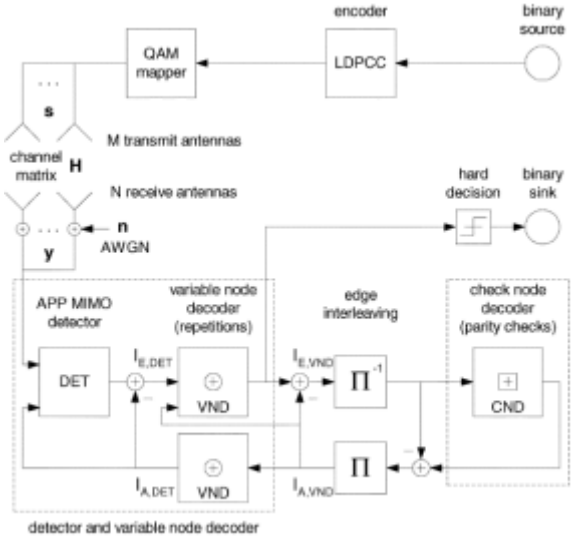


Figure 3. Iterative decoder for LDPC-coded MIMO systems

The EXIT curve formulas for this MIMO system are derived in [15]:

a. For check nodes:

$$I_{E,CND}(I_A, d_c) \approx 1 - J\left(\sqrt{d_c - 1}.J^{-1}(1 - I_A)\right) \tag{3}$$

or

$$I_{A,CND}(I_E, d_c) \approx 1 - J\left(\frac{J^{-1}(1 - I_E)}{\sqrt{d_c - 1}}\right) \tag{4}$$

where J(.) is a function that is defined in [15]:

$$J(\sigma) = \begin{cases} -0.0421\sigma^3 + 0.2093\sigma^2 - 0.0064\sigma & if\ 0 \le \sigma \le \sigma^* = 1.6363 \\ 1 - e^{0.0018\sigma^3 - 0.1427\sigma^2 - 0.0822\sigma + 0.0550} & if\ \sigma^* < \sigma < 10 \\ 1 & if\ \sigma \ge 10 \end{cases} \tag{5}$$

b. For variable nodes:

$$I_{E,VND}(I_{A,VND}, I_{E,DET}, d_v) = J\left(\sqrt{(d_v - 1).\left[J^{-1}(I_{A,VND})\right]^2 + \left[J^{-1}(I_{E,DET})\right]^2}\right) \tag{6}$$

where: $I_{E,DET}(I_{A,DET}, \frac{E_b}{N_0}, R)$ can be found by Monte-Carlo simulation and

$$I_{A,DET}(I_{A,VND}, d_v) = J\left(\sqrt{d_v}.J^{-1}(I_{A,VND})\right) \tag{7}$$

2.3 Factor graph decoupling

Any LDPC code is represented by its parity check matrix H, and also by its factor graph, which consists of the variable nodes (denoted by circles) and check nodes (denoted by squares). There is a connection between a check node i and a variable node j if and only if [H]i,j = 1. Figure 4a shows the factor graph of a rate ½ regular (3,6) LDPC code. Because we're considering the LDPC code for relay channel, which

consists of B blocks, it's more convenient to use the shorthand representation for each factor graph structure similar to the one in Figure 4a. Figure 4b shows the equivalent shorthand representation of the LDPC code described by Figure 4a.

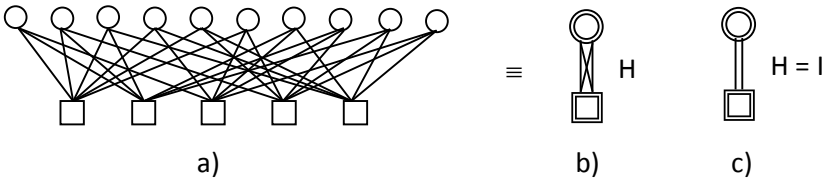


Figure 4. Factor graph of LDPC codes and its shorthand notation.

2.3.1 Relay factor graph

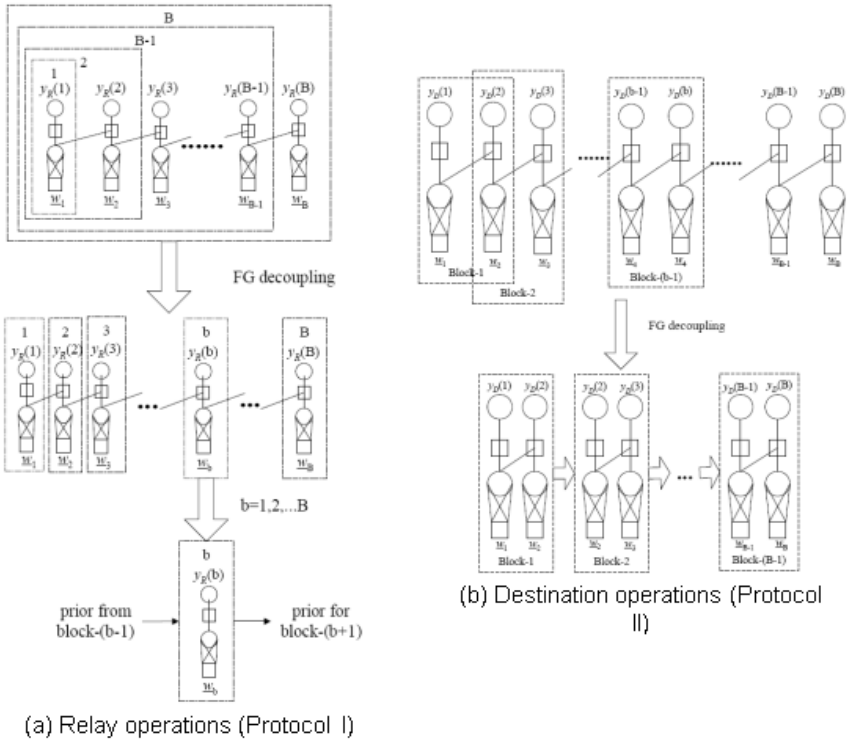


Figure 5. Factor graph for relay and destination

For Protocol I, the decoded message $\hat{w}(t)$ depends only on w(t), so the factor graph of the relay decoder consists of B independent, separated individual factor graphs, and we can consider each individual factor graph as a SISO model.

For Protocol II, there is a connection between the variable node w(t) with the variable node w(t+1), and so the decoded codeword $\hat{w}(t)$ depends on $y_R(t')$ for t' = 1, 2, …, B. The factor graph for Protocol II is shown on the top part of Figure 5a. Decoding such a factor graph is impractical, therefore, a factor graph decoupling is proposed in [19] to overcome this problem. By decoupling the original factor graph as shown in the lower parts of Figure 5, we are now going to deal with B partial factor graphs. In particular, to decode the b-th factor graph, we need only the observation $y_R(b)$ and the prior distribution of w(b-1). Denote $x_{R,j}^{DF}(t) \square [w_{t,j}, w_{t-1,j}]^T$ as the codeword transmitted by the source node during timeslot t, where the subscripts j indicate the j-th elements of the corresponding vector (for example, $w_{t,j}$ indicates the j-th element of the vector w(t)). Then:

$$y_{R,j}(t) = \underbrace{h_{SR,j}(t)\left[\sqrt{P_{S,1}}, \sqrt{P_{S,2}}\right]}_{\boldsymbol{h}_{R,j}^{DF}(t)} x_{R,j}^{DF}(t) + n_{R,j}(t) \quad (8)$$

This equation represents a virtual MISO model.

2.3.2 Destination factor graph

Figure 5b shows the factor graph for the destination decoding operations. Again, it's decoupled into B-1 partial factor graphs as shown in the lower part of Figure 6. The successive decoding method, proposed in [11], can be applied here. For forward-decoding method, the decoding starts from the first partial factor graph, gets the decoded codeword $\hat{w}(1)$ and $\hat{w}(2)$. After solving the first partial graph, the estimation $\hat{w}(1)$ will be much better than $\hat{w}(2)$ [11]. Then $\hat{w}(2)$ is used as input of the second partial factor graph. After solving the 2nd partial factor graph, the estimation $\hat{w}(2)$ has better performance. Then the process continues until we reach the final partial factor graph. For backward-decoding, the process start with the (B-1)-th factor graph, and then, the (B-2)-th factor graph, and so on.

For forward-decoding, denote $x_{f,j}(t) \square \left[w_{t,j}, w_{t+1,j}\right]^T$ and $y_{f,j}(t) \square \left[\tilde{y}_{D,j}(t), \tilde{y}_{D,j}(t+1)\right]^T$ as the transmitted codewords from the source and the received signals at the destination, respectively. For backward-decoding, denote $x_{b,j}(t) \square \left[w_{t,j}, w_{t-1,j}\right]^T$ and $y_{b,j}(t) \square \left[\tilde{y}_{D,j}(t), \tilde{y}_{D,j}(t-1)\right]^T$ as the transmitted codewords from the source and the received signals at the destination, respectively. Then we can express $y_{b,j}(t)$ and $y_{f,j}(t)$ as virtual MIMO models [19].

2.4 LDPC-coded cooperative system performance analysis

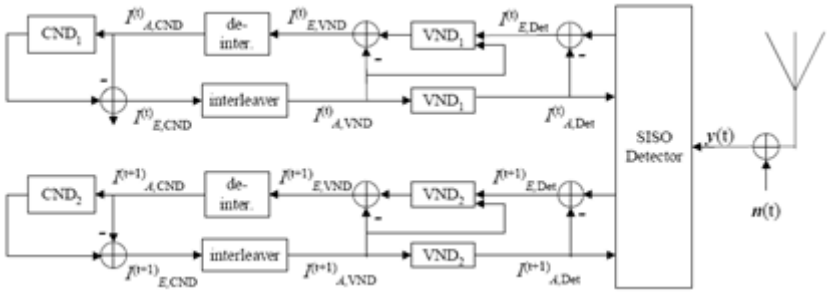


Figure 6. Iterative receiver for LDPC-coded cooperative system (virtual MISO model).

Based on the framework proposed by S. ten Brink et. al. [15], the iterative receivers for LDPC-coded relay systems are introduced in [19] (e.g, see Figure 7). Here we successively consider the SISO model and MISO model in this section. Denote $\{\lambda_j\}_{j=2}^{D_V}$ and $\{\rho_j\}_{j=2}^{D_C}$ as the edge distributions of LDPC code ensemble, where D_V and D_C are the maximum degree of the variable nodes and the check nodes, respectively. The performance analysis algorithms are summarized as follows [19].

2.4.1 Relay nodes

Algorithm 1: Performance analysis for single-relay operations under Protocol I (SISO model)

Inputs of the algorithm1 are: code ensemble edge distributions $\left\{\left\{\lambda_j\right\}_{j=2}^{D_V},\left\{\rho_j\right\}_{j=2}^{D_C}\right\}$, $\frac{E_b}{N_0}$, coding rate R and power allocation $\alpha=\frac{P_R}{P_R+P_S}$.Initialization: set $I_{E,CND}=0$.

a. Calculate the average mutual information of detector output: $I_{\det}(h_{SR},R,\frac{E_b}{N_0},\alpha)$. For fast fading channels, $I_{det}(.)$ is computed numerically using Monte-Carlo simulation. For AWGN channels, use

$$I_{\det}=J\left(2\left|h_{SR}\right|\sqrt{2(1-\alpha)R\frac{E_b}{N_0}}\right) \tag{9}$$

where $J(\sigma)\ \square\ 1-\int_{-\infty}^{+\infty}\frac{e^{-\left(x-\sigma^2/2\right)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}\log_2(1+e^{-x})dx$.

b. Set $I_{A,VND}=I_{E,CND}$; then calculate the average extrinsic mutual information for the variable nodes:

$$I_{E,VND}=\sum_{i=2}^{D_V}\lambda_i J\left(\sqrt{\left[J^{-1}(I_{\det})\right]^2+(i-1)\left[J^{-1}(I_{A,VND})\right]^2}\right) \tag{10}$$

c. Set $I_{A,CND}=I_{E,VND}$; then calculate the average extrinsic mutual information for the check nodes:

$$I_{E,CND}=\sum_{j=2}^{D_V}\rho_j\left[1-J\left(\sqrt{(j-1)}J^{-1}\left(1-I_{A,CND}\right)\right)\right] \tag{11}$$

d. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (b).

$$I_R=\sum_{i=2}^{D_V}\lambda_i J\left(\sqrt{\left[J^{-1}(I_{\det})\right]^2+i\left[J^{-1}(I_{A,VND})\right]^2}\right) \tag{12}$$

Algorithm 2: Performance analysis for single-relay operations under Protocol II (Virtual MISO model)

During time slot t: the inputs are $I_{A,VND}^{(t-1)}$, $\left\{\left\{\lambda_j\right\}_{j=2}^{D_V},\left\{\rho_j\right\}_{j=2}^{D_C}\right\}$, $\frac{E_b}{N_0}$, R, α. Initialization: set $I_{A,VND}^{(t)}=0$

.

a. Calculate $I_{E,Det}^{(k)}(i)=f_{R,Det}\left(I_{A,Det}^{(t)}(i),I_{A,Det}^{(t-1)}(i),\boldsymbol{h}_R^{DF},\frac{E_b}{N_0},R\right)$ for k = t, t-1 and $2\le i\le D_V$ (13)

The function $f_{R,Det}$ (.) can be computed numerically by Monte-Carlo simulation.

b. Calculate

$$I_{E,VND}^{(k)}=\sum_{i=2}^{D_V}\lambda_i J\left(\sqrt{\left[J^{-1}(I_{E,Det}^{(k)}(i))\right]^2+(i-1)\left[J^{-1}(I_{A,VND}^{(k)})\right]^2}\right) \tag{14}$$

and set $I_{A,CND}^{(k)} = I_{E,VND}^{(k)}$ for k = t, t-1.

c. Calculate $I_{E,CND}^{(k)}$ using **Error! Reference source not found.** and set $I_{A,VND}^{(k)} = I_{E,CND}^{(k)}$ for k = t, t-1.

d. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (a).

$$I_R^{(k)} = \sum_{i=2}^{D_V} \lambda_i J\left(\sqrt{\left[J^{-1}(I_{E,Det}^{(k)}(i))\right]^2 + i\left[J^{-1}(I_{A,VND}^{(k)})\right]^2}\right), \text{ k = t, t-1} \quad (15)$$

2.4.2 Destination nodes

The decoding operations at the destination is equivalent to a virtual MIMO model. By the similar method which was applied for relay operations, we can derive the performance analysis algorithm for destination node. However, different from the previous MISO model, in this case, we have to deal with the imperfect relay decoding. To solve this problem, the author of [19] proposed a BSC modeling for the relay decoding as follows. Each bit $\hat{w}_{t,j}$ of the decoded codeword from a certain relay is considered as the output of a BSC channel whose input is $w_{t,j}$. By using Gaussian approximation for the LLR of this BSC channel, its crossover probability can be estimated by

$$p_0 = Q\left(\frac{J^{-1}(I_R)}{2}\right) \quad (16)$$

$Q(x) \square \int_x^{+\infty} \frac{e^{-t^2/2}}{\sqrt{2\pi}} dt$. Now the Algorithm 3 for the destination node can be summarized below:

Algorithm 3: Performance analysis for joint relay-destination operations (Virtual MIMO model)

During time slot t: the inputs are $\left\{\{\lambda_j\}_{j=2}^{D_V}, \{\rho_j\}_{j=2}^{D_C}\right\}$, $\frac{E_b}{N_0}$, R. Initialization: set $I_{A,VND}^{(t)} = I_{A,VND}^{(t-1)} = 0$

a. Compute the mutual information I_R of the relay output using Algorithm 1 (for Protocol I) or Algorithm 2 (for Protocol II).

b. Compute the crossover probability p_0 of the equivalent BSC channel for the relay output using **Error! Reference source not found.**.

c. Analyze the destination performance as follows:

c1. Randomly generate w(k) and $\hat{w}(k)$ using p_0 above. Then compute the mutual information of the destination detector output: $I_{E,Det}^{(k)}(i) = f_{D,Det}\left(I_{A,Det}^{(t)}(i), I_{A,Det}^{(t-1)}(i), \boldsymbol{H}, p_0, \frac{E_b}{N_0}, R\right)$. The function $f_{D,det}(.)$ can be computed numerically.

c2. Calculate $I_{E,VND}^{(k)}$ using **Error! Reference source not found.** and set $I_{A,CND}^{(k)} = I_{E,VND}^{(k)}$ for k = t, t-1.

c3. Calculate $I_{E,CND}^{(k)}$ using **Error! Reference source not found.** and set $I_{A,VND}^{(k)} = I_{E,CND}^{(k)}$ for k = t, t-1.

c4. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (c1).

$$I_D^{(k)} = \sum_{i=2}^{D_V} \lambda_i J\left(\sqrt{\left[J^{-1}(I_{E,Det}^{(k)}(i))\right]^2 + i\left[J^{-1}(I_{A,VND}^{(k)})\right]^2}\right), \text{ k = t, t-1} \quad (17)$$

2.4.3 LDPC code ensemble optimization

With the framework above for evaluating the performance of LDPC codes, the optimal code ensemble can be found by some searching strategies, for example, different evolution [12]. First, we generate a code ensemble, then evaluate its performance using the Algorithm 3. The criterion for assessment is the mutual information of the destination output. According to this value, we update the code ensemble to get better performance. Repeat these step until the mutual information converges to its optimum value, then we get the optimal code.

The remaining problem is the complexity of the optimization procedure. The bottle-neck of this procedure is computing $f_{R,Det}(.)$ and $f_{D,Det}(.)$ in Algorithm 3. In general, both these functions don't have closed-form expression, and must be computed numerically; therefore, it makes the optimization very time-consuming, or even infeasible. In [19], an efficient destination detector based on Gaussian approximation is introduced. The key idea is: based on the expression of the received signal at the destination, the signal component $y_D(t)$ which corresponds to time slot t, has little effect on detecting w(t+1). Therefore, the log-likelihood ratio $L_{E,Det}^{(t+1)}$ can be approximated using $y_D(t+1)$ only. Then, by using Gaussian approximation for $L_{E,Det}^{(t+1)}$, we can develop a sub-optimum detector $L_{E,Det}^{(t+1)}$, and can compute the average SNR of w(t+1), which is denoted as $\gamma_{E,Det}^{(t+1)}$. Using this, we compute the extrinsic mutual information: $I_{E,Det}^{(t+1)} = J\left(2\sqrt{\gamma_{E,Det}^{(t+1)}}\right)$. Now, the sub-optimum detector for w(t) can be performed by using $y_D(t)$ and $y_D(t+1)$ separately, which generate two LLR outputs. Then we sum up the results and get $L_{E,Det}^{(t)}$. This LLR is used to compute the average SNR ($\gamma_{E,Det}^{(t)}$) for w(t), and hence, the corresponding extrinsic mutual information $I_{E,Det}^{(t)} = I\left(w(t), L_{E,Det}^{(t)}\right) = \sum_{X=-1,1} I\left(X, L_{E,Det}^{(t)}\right)$. Hence, step (c1) in Algorithm 3 can be performed semi-analytically, and the complexity is reduced [19].

2.5 Simulation results

In the simulation, we use the code ensemble of rate R = ½. The distance between source and destination is normalized to 1, and the distance from source to relay is d < 1. Under Protocol I, we consider two cases: d = 0.25 and d = 0.25, and the corresponding power allocation are α = 0.79 and α = 0.44, respectively. Under Protocol II, we choose the optimum power allocation for AWGN channels, which is expressed by the following formulas:

$$P_{S,1} = \frac{\left(|h_{SD}|^2 + |h_{RD}|^2\right)P_{total}}{|h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2}, \; P_{S,2} = \frac{\left(|h_{SD}|^2 . |h_{SR}|^2\right)P_{total}}{\left(|h_{SD}|^2 + |h_{RD}|^2\right)\left(|h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2\right)}$$

$$P_R = \frac{\left(|h_{RD}|^2 . |h_{SR}|^2\right)P_{total}}{\left(|h_{SD}|^2 + |h_{RD}|^2\right)\left(|h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2\right)}$$

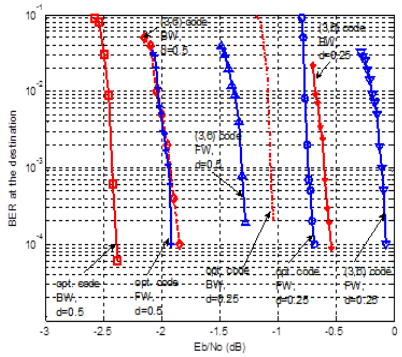


Figure 7. BER performance at the destination: AWGN channels, single-relay, Protocol I

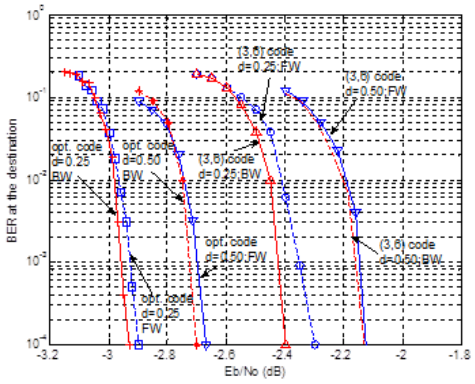


Figure 8. BER performance at the destination: AWGN channels, single-relay, Protocol II

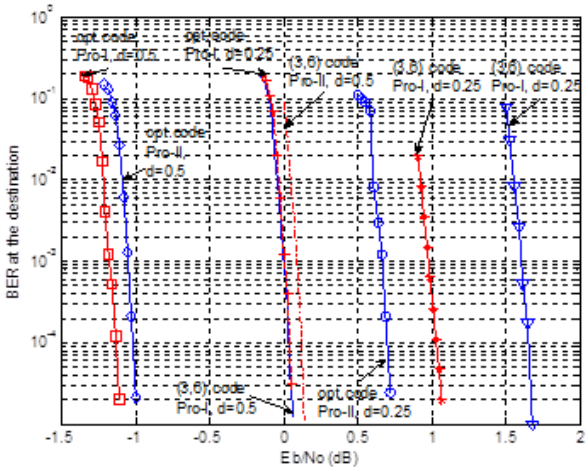


Figure 9. BER performance at the destination: fast fading channels, single-relay, backward-decoding

Figure8 and Figure 9 show the BER performance at the destination for single-relay systems on AWGN channels, under Protocol I and Protocol II, respectively. Figure 10 shows the BER performance for single-relay system on fast fading channel.

The following are some important observation which drawed from the simulation results:

The optimized codes outperform the (3,6) codes with the gain about 1dB.

Backward-decoding outperforms forward-decoding. But under Protocol II, the gain is unsignificant.

The optimized codes approach the capacity bound within 0.1dB gap for d = 0.25, and within 0.5dB gap for d = 0.5.

Protocol II is better than Protocol I in AWGN case, Protocol I is better than Protocol II in fast fading case. However, Protocol I is a special case of Protocol II when $P_{S,2} = 0$. We can conclude that the performance of Protocol II depends on how well the power is allocated.

3 Bilayer LDPC codes for Decode-and-Forward Cooperative Communication Systems

3.1 Background

3.1.1 Decode-and-Forward (DF) Strategy

In this section, we review the decode-and-forward cooperation strategy which is first introduced by Cover and Gamal in 1979 **[2]**. Recall that a Gaussian relay channel can be modeled as:

$$Y_1 = X + Z_1 \tag{18}$$

$$Y = X + X_1 + Z_2 \tag{19}$$

where Y_1 and Y are the received signal at the relay node and the destination, respectively; X_1 and X are the signals transmitted by the relay and the source, respectively. $Z_1 \sim N(0,N_1)$ and $Z_2 \sim N(0,N_2)$ are AWGN noises at the relay node and the destination, respectively. The decode-and-forward strategy is described as follows.

Let n be the number of bits in each codeword transmitted by the source. During block i, the source selects a message $w_i \in \{1,2,\ldots,2^{nR}\}$, where R is the rate of the code. The set of source messages are randomly partitioned into 2^{nR_1} bins, each of which has the size of $2^{n(R-R_1)}$ ($R_1 \le R$). Let s_i denote the bin index of the message w_{i-1}. In block i, the source transmit a linear combination of the encoded version of the message w_i and the bin index of message w_{i-1}, while the relay transmits the encoded version of the bin index s_i of the previous message:

$$\underbrace{\boldsymbol{X}(w_i \mid s_i)}_{\text{transmitted by src}} = \underbrace{\tilde{\boldsymbol{X}}(w_i)}_{\text{enc.of .}w_i} + \sqrt{\frac{(1-\alpha)P}{P_1}}. \underbrace{\boldsymbol{X}(s_i)}_{\substack{\text{enc.of .}s_i \\ \text{transmitted by relay}}} \tag{20}$$

where P and P_1 are the maximum transmit powers of source and relay, respectively; α is a fraction of power used for transmitting new message w_i.

The decoding process happens as follows: the relay node know $\mathbf{X}(s_i)$, so it can decode w_i based on $\tilde{\boldsymbol{X}}(w_i)$ (plus Gaussian noise Z_1). After decoding w_i, it can compute s_{i+1}, which is transmitted in the next block. At the destination,

$$\boldsymbol{Y} = \tilde{\boldsymbol{X}}(w_i) + \left(1 + \sqrt{\frac{(1-\alpha)P}{P_1}}\right)\boldsymbol{X}(s_i) + \boldsymbol{Z}_2 \tag{21}$$

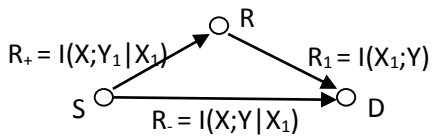


Figure 10. Code designing problem for DF relay systems

First, the destination considers $\tilde{\boldsymbol{X}}(w_i)$ as noise and decode s_i. After that, the $\mathbf{X}(s_i)$ is subtracted from **Y**, and the remaining is used to decode w_i. For successful decoding in each step, R, R_1 and $R - R_1$ must be upper-bounded by some constraints (Shannon's Theorem). Combining these constraints, the overall DF rate is upper-bounded by

$$R \le \max_{\alpha} \min \left\{ \frac{1}{2}\log\left(1+\frac{\alpha P}{N_1}\right), \frac{1}{2}\log\left(1+\frac{P+P_1+2\sqrt{(1-\alpha)PP_1}}{N_1+N_2}\right) \right\} \quad (22)$$

The goal of code designing for DF relay systems is summarized in Figure 11: want to construct a source codebook that simultaneously approach the rates R_+ and R_-, and a relay codebook to approach the rate $R_1 = R_+ - R_-$.

3.1.2 Parity forwarding

Naturally, binning can be implemented by generating extra parity bits (or syndromes) on the codewords in the source codebook $\tilde{X}$. Codewords in each bin must satisfy a set of parity equation. Therefore, we can implement binning on the DF relay systems using LDPC code as follows.

a. The source message is encoded using an (n, n-k_1) LDPC code and then transmitted during block i.

b. The relay node decodes the transmitted codeword $\tilde{\boldsymbol{X}}(w_i)$, generating k_2 extra parity bits, encodes these k_2 bits using its codebook $\tilde{X}_1$ and send the result codeword to the destination in block i + 1.

c. The destination first decodes the extra k_2 parity bits, then decodes the source message over a bilayer code construction.

This strategy is called parity-forwarding strategy.

3.2 Designing Bilayer-Expurgated LDPC Codes

This code is proposed by P. Razaghi and W. Yu [20]. Its structure is shown in Figure 12. The lower layer of Figure 11 represents an (n, n – k_1) LDPC code for source-relay channel. The whole graph represent a (n, n – k_1 – k_2) subcode of the lower layer code, which is called bilayer-expurgated code. This subcode satisfies two set of parity equations: k_1 equations by the source codebook and k_2 equations by the relay codebook.

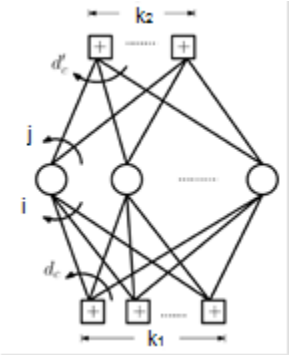


Figure 11. Bilayer-expurgated LDPC code structure

Our goal is to design the source and relay codes such that the lower code can approach the capacity R_+ and the bilayer code can approach the capacity R_-.

3.2.1 BE-LDPC code ensemble

Like a standard LDPC code ensemble, an ensemble of bilayer-expurgated LDPC code is defined based on edge distributions. However, there is some modification in the definition of edge distributions for bilayer-expurgated LDPC codes. First notice that we have two sets of check nodes: the lower check

nodes corresponding to the k_1 source parity checks and the upper check nodes corresponding to the k_2 relay parity checks. Now we define:

- *A lower edge* is an edge connecting a variable node to a lower check node.
- *An upper edge* is an edge connecting a variable node to an upper check node.
- *Lower degree of a variable node* is the number of lower edges connected to it.
- *Upper degree of a variable node* is the number of upper edges connected to it.
- *Check node degree of a check node* is the number of edges connected to it.
- *Lower degree of an edge* is the lower degree of the variable node it is connected to.
- *Upper degree of an edge* is the upper degree of the variable node it is connected to.
- *An edge (or a variable node) has degree (i,j)*iff it has lower degree i and upper degree j.
- *Variable degree distribution* $\lambda_{i,j}$ is the percentage of edges with degree (i,j) ($i \geq 2$, $j \geq 0$)
- η percentages of *lower* edges in the bilayer graph.

A bilayer-expurgated LDPC code ensemble is characterized by $\left\{\left\{\lambda_{i,j}\right\}_{\substack{i\geq 2\\ j\geq 0}}, \eta\right\}$.

3.2.2 Bilayer Density Evolution

Different from the density evolution for standard LDPC codes, the density evolution for bilayer LDPC codes employs two densities to be tracked: lower density corresponding to the messages in lower part and upper density corresponding to the upper part of the bilayer graph.

Denote $p^{(t)}$ and $q^{(t)}$ as the message pdf at the input of the lower and upper check nodes, respectively, at the beginning of the t-th iteration. Because each check node involves in only one kind of density (lower or upper), the update rule for check nodes of bilayer code is not different from the update rule for check nodes of standard LDPC codes. Denote $p'^{(t)}$ and $q'^{(t)}$ as the densities resulted from the lower and upper check update, respectively. Then:

$$p'^{(t)} = F_{L,CND}(p^{(t)}, d_c) \tag{23}$$

$$q'^{(t)} = F_{U,CND}(q^{(t)}, d_c^{'}) \tag{24}$$

where d_c and $d_c^{'}$ are the lower and upper check degree, respectively; $F_{L,CND}(.)$ and $F_{U,CND}(.)$ are the update functions for lower and upper check nodes, respectively.

Now we proceed with the message density at a degree (i, j) variable node at the beginning of the (t+1)-th iteration:

$$p_{i,j}^{(t+1)} = \left(\otimes^{i-1} p'^{(t)}\right) \otimes \left(\otimes^{j} q'^{(t)}\right) \otimes p_c, i \geq 2, j \geq 0 \tag{25}$$

$$q_{i,j}^{(t+1)} = \left(\otimes^{i} p'^{(t)}\right) \otimes \left(\otimes^{j-1} q'^{(t)}\right) \otimes p_c, i \geq 2, j \geq 1 \tag{26}$$

where p_c is the density of the LLR received over the channel, and $\otimes^k f = f \otimes f \otimes ... \otimes f$ (k terms) for $k \geq 2$ and $\otimes^1 f = f, \otimes^0 f = \delta$ (Dirac function).

Finally, the message densities at input of the lower and upper check nodes at the beginning of the (t+1)-th iteration is updated as

$$p^{(t+1)} = \sum_{i\geq 2, j\geq 0} \frac{i}{i+j} \lambda_{i,j} p_{i,j}^{(t+1)} \tag{27}$$

$$q^{(t+1)} = \sum_{i\geq 2, j\geq 0} \frac{i}{i+j} \lambda_{i,j} q_{i,j}^{(t+1)} \tag{28}$$

The overall message error probability at the beginning of the (t+1)-th iteration is

$$e\left(p^{(t+1)}, q^{(t+1)}\right) = \sum_{i\geq 2, j\geq 0} \lambda_{i,j} \left(\frac{i}{i+j} e_{i,j}^{1}(p^{(t)}, q^{(t)}) + \frac{j}{i+j} e_{i,j}^{2}(p^{(t)}, q^{(t)}) \right) \tag{29}$$

where $e_{i,j}^{1}(p^{(t)}, q^{(t)})$ and $e_{i,j}^{2}(p^{(t)}, q^{(t)})$ are the message error probability corresponding to the densities $p_{i,j}^{(t+1)}$ and $q_{i,j}^{(t+1)}$, respectively, after one evolution iteration with inputs $p^{(t)}$ and $q^{(t)}$, respectively.

3.2.3 BE-LDPC code optimization

There are many approaches to optimization the code ensemble to achieve the goal we mentioned as the beginning of this section. A simple but efficient approach is proposed in [20]. The key idea is first fix the check degrees $d_c, d_c^{'}$ and the lower graph, try to find the edge distributions such that the rate of the bilayer code approaches R-. Then change to another pair ($d_c, d_c^{'}$) and repeat the optimization process.

Step 1: Fix $d_c, d_c^{'}$ and the lower subgraph. The rate of the bilayer code is $1 - (k_1 + k_2)/n$, so to maximize it we need to minimize k_2, or equivalently, maximize η because $\eta = \frac{d_c k_1}{d_c k_1 + d_c^{'} k_2}$. By fixing the lower graph, we also fix its edge distribution λ_i, which is related to $\lambda_{i,j}$ by

$$\lambda_i = \frac{1}{\eta} \sum_{j\geq 0} \frac{i}{i+j} \lambda_{i,j} \Leftrightarrow \sum_{j\geq 0} \frac{i}{i+j} \lambda_{i,j} - \eta \lambda_i = 0 \tag{30}$$

Therefore, our problem at step 1 can be formulated as follows:

$$\max_{\{\lambda_{i,j}\}, \eta} \quad \eta \tag{31}$$

$$\text{subject to} \quad \sum_{j\geq 0} \frac{i}{i+j} \lambda_{i,j} - \eta \lambda_i = 0 \tag{32}$$

$$\sum_{i\geq 2, j\geq 0} \lambda_{i,j} \left(\frac{i}{i+j} e_{i,j}^{1}(p^{(t)}, q^{(t)}) + \frac{j}{i+j} e_{i,j}^{2}(p^{(t)}, q^{(t)}) \right) < \mu^h e\left(p^{(t)}, q^{(t)}\right) \tag{33}$$

$$\sum_{\substack{i\geq 2 \\ j\geq 0}} \lambda_{i,j} = 1 \tag{34}$$

where h is the optimization iteration number, $0 < \mu^h < 1$ is a convergence factor that is increased in each iteration towards 1. This problem is a linear programming problem and can be solved iteratively [20]

Step 2: Establish an appropriate range for $d_c, d_c^{'}$ and searching over this range to find the optimal value. To do this, repeat Step 1 for each feasible pair ($d_c, d_c^{'}$).

3.3 Designing Bilayer-Lengthened LDPC Code

In this section, we consider the second codes proposed in [20], namely, the bilayer-lengthened LDPC codes (BL-LDPC). Its representation graph is shown in Figure 13. Both the lower graph and the overall graph have the same number of check nodes.

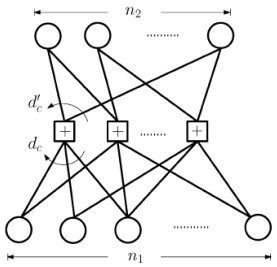


Figure 12. Bilayer-lengthened LDPC code

First, the source transmits the codewords from the (n_1 + n_2, k_1) LDPC codes (denoted as C_1) represented by the overall graph. The relay then decodes the source codeword. It encodes the n_2 bits corresponding to the upper part of the graph, using a (n_2, k_2) LDPC code (denoted as C_2) (by adding k_2 parity bits), and then sends these k_2 parity bits to the destination, using a codebook of rate $R_1 = R_+ - R_-$. Our goal is to design the source and relay codes such that the lower code can approach the capacity R_- and the bilayer code can approach the capacity R_+.

For decoding at the destination, in each block, the k_2 parity bits are decoded first, then they were used to decode the n_2 upper variable nodes of the source codeword transmitted in the previous block. The destination then removes the upper part of the overall graph and updates the parity check equations. Finally, the remaining part (the lower part) of the graph is decoded to get the data.

This code structure has good performance in the channel conditions which have large gap between R_+ and R_-.

3.3.1 Bilayer-lengthened LDPC code ensemble

The BL-LDPC code is dual to the BE-LDPC code in the sense of interchanging between the role of variable nodes and check nodes. So we have the similar definitions as the previous section.

- *A lower edge* is an edge connecting a check node to a lower variable node.
- *An upper edge* is an edge connecting a check node to an upper variable node.
- *Variable node degree of a variable node* is the number of edges connected to it.
- *Variable degree of an edge* is the degree of the variable node it is connected to.
- Lower variable degree distribution $\lambda_{i,1}$ is the percentage of lower edges with degree i ($i \geq 2$).
- Upper variable degree distribution $\lambda_{i,2}$ is the percentage of upper edges with degree i ($i \geq 2$).
- $d_c, d_c^{'}$ are the number of edges in the lower and upper subgraph, respectively.

A Bilayer-Lengthened LDPC code ensemble is defined by $\left\{\left\{\lambda_{i,1}\right\}_{i\geq 2},\left\{\lambda_{i,2}\right\}_{i\geq 2},d_c,d_c^{'}\right\}$.

3.3.2 Bilayer Density Evolution

Denote $p^{(t)}$ and $q^{(t)}$ as the message pdf in the lower and upper parts, respectively, at the beginning of the t-th iteration. Denote $p^{'(t)}$ and $q^{'(t)}$ as the densities resulted from the lower and upper check update, respectively. Denote $\oplus$ as the check density-update operation, and $\oplus^d f = f \oplus f \oplus \ldots \oplus f$ (d terms), $\oplus^1 f = f$, $\oplus^0 f = 1$. Then:

$$p^{'(t)} = \left(\oplus^{d_c-1} p^{(t)}\right) \oplus \left(\oplus^{d_c^{'}} q^{(t)}\right), d_c > 1 \tag{35}$$

$$q^{'(t)} = \left(\oplus^{d_c} p^{(t)}\right) \oplus \left(\oplus^{d_c^{'}-1} q^{(t)}\right), d_c^{'} \geq 1 \tag{36}$$

The update rules for the message densities at a variable node of degree i in the lower and upper subgraphs, respectively, are:

$$p_i^{(t+1)} = \left(\otimes^{i-1} p^{'(t)}\right) \otimes p_c, i \geq 2 \tag{37}$$

$$q_i^{(t+1)} = \left(\otimes^{i-1} q^{'(t)}\right) \otimes p_c, i \geq 2 \tag{38}$$

where p_c is the density of the LLR received over the channel.

Finally, the message densities in the lower and upper parts at the beginning of the (t+1)-th iteration is updated as

$$p^{(t+1)} = \sum_{i\geq 2} \lambda_{i,1} p_i^{(t+1)} \tag{39}$$

$$q^{(t+1)} = \sum_{i\geq 2} \lambda_{i,2} q_i^{(t+1)} \tag{40}$$

The overall message error probability at the beginning of the (t+1)-th iteration is:

$$e\left(p^{(t+1)}, q^{(t+1)}\right) = \sum_{i\geq 2} \eta \lambda_{i,1} e_{i,1}(p^{(t)}, q^{(t)}) + (1-\eta) e_{i,2}(p^{(t)}, q^{(t)}) \tag{41}$$

where $\eta = \frac{d_c}{d_c + d_c^{'}}$ and $e_{i,1}(p^{(t)}, q^{(t)})$, $e_{i,2}(p^{(t)}, q^{(t)})$ are the message error probability corresponding to the densities $p_i^{(t+1)}$ and $q_i^{(t+1)}$, respectively, after one evolution iteration with inputs $p^{(t)}$ and $q^{(t)}$, respectively.

3.3.3 BL-LDPC code optimization

Step 1: Fix $d_c, d_c^{'}$, find the lower variable distributions $\lambda_{i,1}$ such that the LDPC code represented by the lower subgraph has the rate approaching to R_-.

Step 2: Fix $d_c, d_c^{'}$, and the lower variable distributions $\lambda_{i,1}$ found from step 1. Our goal at Step 2 is to maximize the rate of the overall bilayer-lengthened code: $1 - k/(n_1 + n_2)$ where k is the number of check nodes. To do this, we need to maximize the number of upper variable nodes n_2, which is given by $n_2 = d_c^{'} k \sum_{i\geq 2} \frac{\lambda_{i,2}}{i}$. Because k and $d_c^{'}$ are fixed, our problem can be formulated as follows:

$$\max_{\lambda_{i,2}} \quad \sum_{i\geq 2} \frac{\lambda_{i,2}}{i} \tag{42}$$

$$\text{subject to:} \quad \sum_{i\geq 2} \eta\lambda_{i,1} e_{i,1}(p^{(t)}, q^{(t)}) + (1-\eta) e_{i,2}(p^{(t)}, q^{(t)}) < \mu^h e\left(p^{(t)}, q^{(t)}\right) \tag{43}$$

$$\sum_{i\geq 2} \lambda_{i,2} = 1 \tag{44}$$

Step 3: Search for optimal $d_c, d_c^{'}$ over a reasonable range. For each feasible ($d_c, d_c^{'}$), repeat Step 1 and Step 2 to find the optimal code and record its performance.

3.4 Simulation results

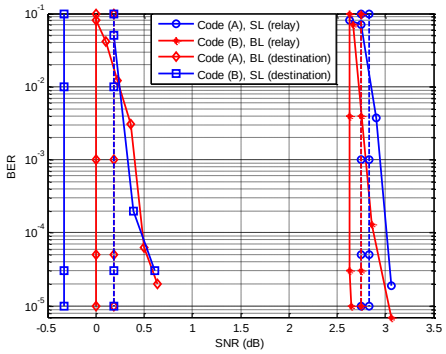


Figure 13. Comparison between expurgated code and lengthened code

In this section, the performance of the bilayer-expurgated and bilayer-lengthened LDPC codes are compared together. Code (A) (expurgated code) and code (B) (lengthened code) are compared in Figure 14. The performance of code (C) and (D) are illustrated in Figure 15 and Figure 16, respectively, to show that the expurgated code is good for the small gap between R_+ and R_-, while the lengthened code is good for the large gap between R_+ and R_-. All of these codes are designed from the optimization procedure mentioned above. The target rates (R_-, R_+) for each code are as follows: (0.3, 0.4) for code(A), (0.5, 0.7) for code (B), (0.3, 0.9) for code (C) and (0.65, 0.7) for code (D). The maximum variable degree for all codes is set to 20. The maximum number of iterations is 600, and the block length is chosen to be 100000. The solid straight lines in each figure represent the theoretical rate limits, while the dashed lines represent the convergence thresholds.

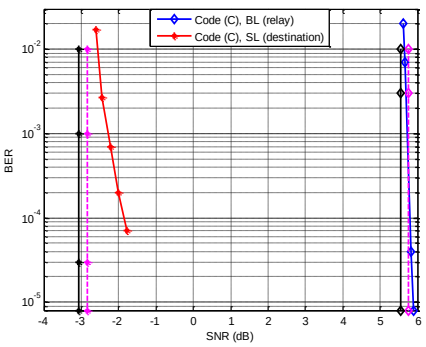


Figure 14. Bilayer-lengthened LDPC code for large SNR gap between relay and destination

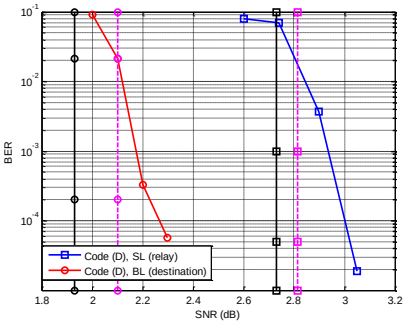


Figure 15. Bilayer-expurgated LDPC code for small SNR gap between relay and destination

The following are some important observations:

- The optimal pair ($d_c, d_c^{'}$) for the codes from (A) to (D) are (15, 4), (8, 6), (5, 33) and (15, 8), respectively.
- The expurgated code does better in the condition of small gap between R_+ and R_-, and the lengthened code does better in the condition of large gap between R_+ and R_-.

The convergence thresholds is close to the theoretical limits (less than 0.5dB), and the SNR gap between the BER curves and the corresponding convergence thresholds are also small, that confirms the asymptotic convergence.

4 Conclusion and Further Work

In this paper I do a survey on how to apply LDPC codes, which can approach the capacity-limit of the communication channels, to the cooperative communications system. Two main concepts has been introduced in this paper. The first one is using iterative decoding and factor graph representation to analyze the performance of LDPC-coded relay systems, in which, the complexity of the analysis is reduced by using factor graph decoupling method. As a result, some algorithms to design the optimized LDPC codes for relay systems have been proposed. The second concept is parity-forwarding. Based on this concept, two new kinds of LDPC codes have been presented, namely, the Bilayer-expurgated LDPC codes and the Bilayer-lengthened LDPC codes. It has been show that these two codes can simultaneously approach the capacity limits of two Gaussian channels (source-relay channel and source-destination channel) at two different SNRs. Further works can be developed from these concepts, for example, consider a multiple relay networks.

REFERENCES

[1]. Aria Nosratinia, Ahmadreza Hedayat, *Cooperative communications in Wireless Networks*, IEEE Communication Magazine, 2004.

[2]. T. M. Cover and A. A. E. Gamal, *Capacity Theorems for the Relay Channel*, Informmation Theory, IEEE Transaction on, 1979. 25(5): pp. 572–84.

[3]. A. Sendonaris, E. Erkip, and B. Aazhang, *User Cooperation Diversity Part I and Part II*, Communication, IEEE Transaction on, 2003. 51(11): pp. 1927–48.

[4]. T. E. Hunter and A. Nosratinia, *Cooperation Diversity through Coding*, Proceedings of 2002 IEEE International. Symposium on Information Theory (ISIT'02), Lausane, Switzerland, 2002. pp. 220.

[5]. K.J. Ray Liu, Ahmed K. Sadek, Weifeng Su, Andres Kwasinski, *Cooperative Communications and Networking,* Cambridge University Press, 2009.

[6]. J. N. Laneman, G. W. Wornell, and D. N. C. Tse, *An Efficient Protocol for Realizing Cooperative Diversity in Wireless Networks*, Proceedings of IEEE ISIT, Washington, DC, June 2001. p. 294.

[7]. G. Kramer, M. Gastpar, and P. Gupta, *Cooperative strategies and capacity theorems for relay networks*, Information Theory, IEEE Transaction on, 2005. 51(9): pp. 3037–3063.

[8]. R. G. Gallagher, *Low-Density Parity-Check Codes,* Cambridge, MA: MIT Press, 1963.

[9]. T. Richardson and R. Urbanke, *The Renaissance of Gallager's Low-Density Pariy-Check Codes*, IEEE Communications Magazine, 2003, 41: pp. 126–131.

[10]. F. R. Kschischang, B. J. Frey, and H. A. Loeliger, *Factor Graphs and the Sum-Product Algorithm*, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 498-519.

[11]. M. A. Khojastepour, N. Ahmed, and B. Aazhang, *Code design for* the *relay channel and factor graph decoding*, *Proceedings of 38th Asilomar Conference on Signals and Systems, Computing. (Asilomar '04),* Pacific Grove, CA, Nov. 2004. pp.2000-2004.

[12]. T. J. Richardson, M. A. Shokrolahhi, and R. L. Urbanke, *Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes*, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 619-637.

[13]. T. J. Richardson, and R. L. Urbanke, The Capacity of Low-Density Parity-Check Codes under Message-Passing Coding, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 599-618.

[14]. E. Sharon, A. Ashikhmin, and S. Litsyn, *EXIT Functions for the Gaussian Channel*, Proceedings of 40th Annual Allerton Conference on Communication, Control, Computing (Allerton '03), Allerton, IL, Oct. 2003. 2:pp. 972-981.

[15]. S. ten Brink, G. Kramer, and A. Ashikhmin, *Design of Low-Density Parity-Check Codes for Modulation and Detection*, Communications, IEEE Transaction on, 2004. 52(4): pp. 670-678.

[16]. S. ten Brink, *Convergence behavior of iteratively decoded parallel concatenated codes*, Communications, IEEE Transaction on, 2001. 49(10): pp. 1727–1737.

[17]. G. Kramer, *Communication strategies and coding for relaying*, in Wireless Communications: IMA Volumes in Mathematics and its Applications, P. Agrawal, D. M. Andrews, P. J. Fleming, G. Yin, and L. Zhang, Eds. New York: Springer-Verlag, 2007. 143: pp. 163–175.

[18]. P. Razaghi and W. Yu, *Parity-forwarding for multiple-relay networks*, in Proceedings of IEEE International Symposium on Information Theory, Seattle, WA, Jul. 2006. pp. 1678–1682.

[19]. C. Li, G. Yue, M. A. Khojastepour, X. Wang, and M. Madihian, *LDPC-coded Cooperative Relay Systems: Performance Analysis and Code Design*, Communications, IEEE Transaction on, 2008. 56(3): pp. 485-496.

[20]. P. Razaghi, and W. Yu, *Bilayer Low-Density Parity-Check Codes for Decode-and-Forward in Relay Channels*, Information Theory, IEEE Transaction on, 2007. 53(10): pp. 3723-3739.

TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

Light Weight Secure Key Generation Protocol with Hidden Generator Point using ECC

[1]Ayaz Hassan Moon and [2]Ummer Iqbal
National Institute of Electronics and Information Technology (NIELIT), J&K, INDIA
[1]dir-srinagar@nielit.gov.in, [2]ummer@nielit.gov.in

ABSTRACT

Key generation and distribution is one of the most important primitive of any security framework. This is irrespective of using a symmetric or asymmetric cryptosystem. However, while securing a WSN, its resource constraint nature cannot be ignored. Therefore Elliptical Curve Cryptography (ECC) based solutions like Elliptical Curve Digital Signature algorithm (ECDSA), Elliptical Curve Diffie-Hellman (ECDH) are becoming more and more popular in comparison to other Public crypto system like RSA. In ECC, the Generator point is treated as a public parameter along with other domain parameters. This can make communication within the WSN vulnerable to man-in-the-middle attack. The attack can be thwarted by keeping the Generator point Private and still be able to establish a common Generator Point across communication parties .It will result in establishing a light weight secure key between a sender and a receiver and achieve other security primitives like generation of MAC and Node identification. This paper discusses and analyses the generation of Shared keys using 1 hidden generator point in comparison to 2- hidden point generator and the conventional ECDH method.

Keywords: Key establishment, Hidden generator, Node Identification, Authentication, WSN, ECC

1 Introduction

Wireless communication being broadcast in nature is more prone to different kind of attacks like eavesdropping, intercept, inject and alter transmitted data. Traditional security solutions based upon public key cryptography are not suitable for wireless sensor networks [1-2]. In conventional networks, message authentication, data integrity and confidentiality are usually achieved by end-to-end security mechanism like SSH, SSL, IP-Sec etc. In end-to-end communication, it is neither necessary nor desirable for the contents of the message (beyond the necessary headers) to be made available to the intermediate routers [3].

The most common security services to be considered for WSN include Confidentiality, Authentication, Integrity, Freshness, Availability, Intrusion detection. In realizing the objectives of the most of the security primitives, Key Management is rightly regarded as the linchpin of Cryptographic mechanism.

Adoption of ECC as an alternative cryptosystem to popular public algorithm like RSA has emerged very strongly in WSN based applications. In ECC, the generator point is to be advertised publically along with other domain parameters. This can increase the vulnerability of the node-to-node communication to be subject to man-in-the-middle-attack. One of the possible solutions to this problem is to keep Generator point Hidden and still arrive at a common shared key between communicating parties.

DOI: 10.14738/tnc.34.1295
Publication Date: 15th July 2015
URL: http://dx.doi.org/10.14738/tnc.34.1295

To establish authenticated communication between sensor nodes, secure key distribution and sharing is imperative. Secure key distribution and sharing in WSN is a research challenge. Most sensor node key exchange requires key distribution before deployment. According to [4], easiest key distribution method is to equip all nodes with same key for establishing communication. But in the event of node capture, entire network is comprised.

The rest of the paper is organized as following:

Related Work, Man-in-the-middle-attack, Suitability of ECC, Suggested protocol, Performance Benchmarking, Conclusion.

2 Related Work

Key generation could be either probabilistic, deterministic or hybrid. Zhu et al proposed Localized Encryption and Authentication Protocol (LEAP) a key management protocol which supports the establishment of four types of keys for each sensor node[5]. It includes an individual key shared with base station, a pair wise key shared with another node, a cluster key shared with multiple neighbouring nodes and a group key which is shared by all the nodes. LEAP provides efficient protocol mechanism for inter-node traffic authentication. LEAP also provides schemes for sensor nodes to establish and update individual keys, pair wise shared keys, cluster keys and group keys, revocation and subsequent rekeying mechanism.

Eschenauer & Gligor proposed random-key pre-distribution scheme that relies on probabilistic key sharing among nodes within the sensor networks. It allots several keys to nodes during Initialization chain [6]. Perrig & Song [7] improved upon security of Esch & Gilgor [6] design by requiring at least two common shared keys for authenticated communication and updating communication keys for subsequent communications.

Trusted server schemes depend on a trusted and secure server such as the base station for key agreements among nodes. The server can be treated as the key distribution centre KDC. The base station is the most appropriate choice for the server and each sensor node stores only an embedded key such that a compromising/captured node cannot reveal much security information about sensor network.

The TinyPK systems described by [8] are designed specifically to allow authentication and key agreement between resource constrained sensors. The agreed upon keys may then be used in conjunction with existing cryptosystems TinySec using Diffie-Hellman key exchange algorithm [9].

Many hybrid broadcast authentication protocols have been proposed which use digital signature in base station or cluster head and use improved MAC in sensor nodes. ZHAO Xin et al have proposed hybrid broadcast authentication protocols (HBA) in wireless sensor networks by selecting Tiny ECC and GBA which is an improved version of μTESLA[10].

Public key cryptography techniques like RSA and elliptic curve cryptography (ECC) were traditionally thought to be impractical for WSN. However recently, several groups have successfully implemented public-key cryptography in WSN. In Gura Etal[4] report both RSA and elliptic curve cryptography is possible using 8 bit CPU with ECC, demonstrating a performance advantage over RSA. ECC's 160 bit keys result in shorter messages compared to 1024 RSA keys.

Xu Huang et al [11] and Ravi Kishore et al [12] demonstrated that the efficiency of ECC implementation is highly dependent on the performance of scalar multiplication. Ravi Kishore et al [13]. Proposed different algorithms based on Hidden generator to overcome man-in-the-middle attack He also suggested 2-point Hidden generator method for arriving at a shared key.

3 Man-In-the Middle Attack

A kind of attack, where in a malicious user/attacker inserts himself between two parties to intercept their active communication. It is a kind of eves dropping which can lead to interception of messages and relaying of wrong messages to both the parties [15]. It can lead to breach of confidentiality, authentication and data integrity and is therefore perceived as a serious threat to any network.

The attacker gains a vantage position by inserting himself between two communicating parties [17]. He can therefore intrude into the communication and inject undesirable communication leading to falsification of data. MIMA attacks lead to session hijacking and is an attack on mutual authentication.

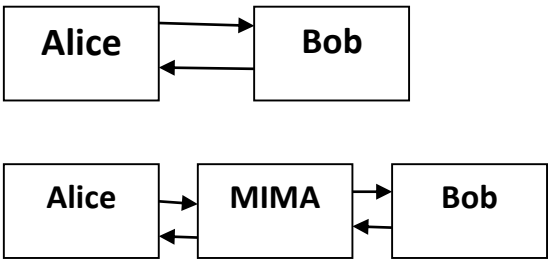


Figure 1: Depicting Man-in-the-middle attack

4 Suitability of ECC

The well-known public crypto systems RSA is based upon modular exponentiation in the integer rings. Its security is derived from the difficulty of factorizing large integers. The solution of integer factorization lies in sub-exponential algorithm [14].

Elliptical Curve Cryptography operates on groups of points over elliptic curve. Its security stems from hardness of elliptic curve discrete logarithmic problem ECDLP. The best known algorithm for solving ECDLP is expo-nential. This implies that attacking ECC is more difficult than attacking RSA. ECC can achieve same level of security as RSA with smaller key size e.g. 160 Bit ECC can provide comparable security to the conventional 1024 Bit RSA. Smaller key size often brings the advantage of faster computation efficiency and saving of bandwidth, memory and energy [14-15]. Therefore ECC is better suited for resource constrained devices like WSN. ECC based ECDSA is used to authenticate new sensor nodes when they join the networks and ECDH (ECC based Diffie-hellmin) algorithm is used to establish shared keys between sensor nodes.

Key length of RSA	Key length of ECC	Ratio of RSA/ECC
512	106	5:1
768	132	6:1
1024	160	7:1
2048	210	10:1

Figure 2: Key comparison between RSA and ECC in terms of security equivalence

4.1 Elliptical Curve Illustration

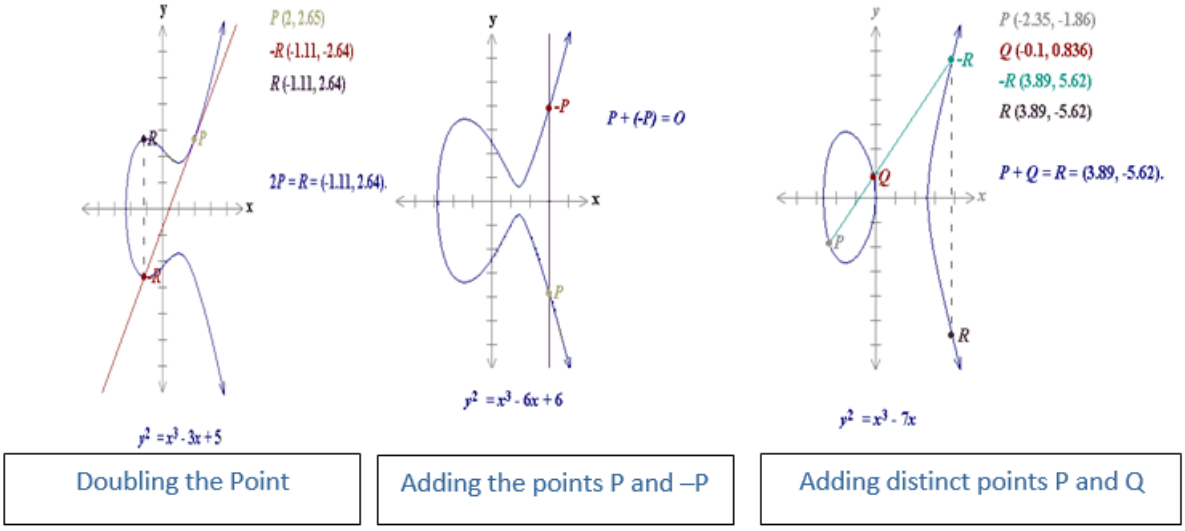


Figure 3: ECC references from www.Certicom.com

4.2 Elliptic Curve Diffie-Hellman (Ecdh) Algorithm

Two parties sharing the same elliptic curve domain parameters can establish a shared secret over an insecure channel without exchanging their respective secret keys. In ECC implementation, the hardness is derived from ECDLP [18]. The flow for establishing the shared secret is as following;

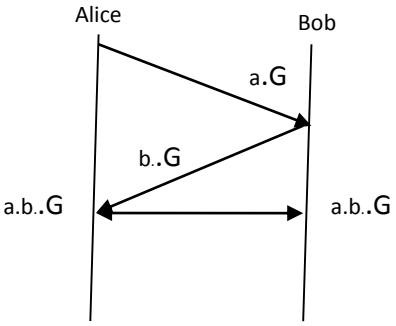


Figure 4: Elliptical curve Diffie-Hellman key generation

5 Suggested Protocols for Key Generation

Two protocols are suggested to implement the key generation with hidden generator points using ECC.

- Key generation using 2-Hidden Generator Points.
- Key generation using 1-Hidden Generator Point.

5.1 Key generation using 2-Hidden Generator Points.

This is similar to the protocol proposed by Ravi[13]. In this protocol, 2 communicating parties i.e Alice and Bob each have their respective hidden generator points Ga and Gb. After selecting their private keys X and Y , they undergo a scalar multiplication with their respective generator points resulting in a X.Ga and Y.Gb both being points on the curve. The parties under take exchanges indicated in the figure 5. and also perform operations based on scalar multiplication and multiplicative inverse . After

6 exchanges, both Alice and Bob are in possession of common generator point G = Ga + Gb , a point on the curve .

This method thwarts the man–in–middle attack as the intruder would not have any access to the either of the generator points lying with bob and Alice, since algorithm leverages the hardness of ECDLP. Extracting Generator points from the scalar multiplicative terms becomes a discrete logarithmic problem which has exponential time complexity.

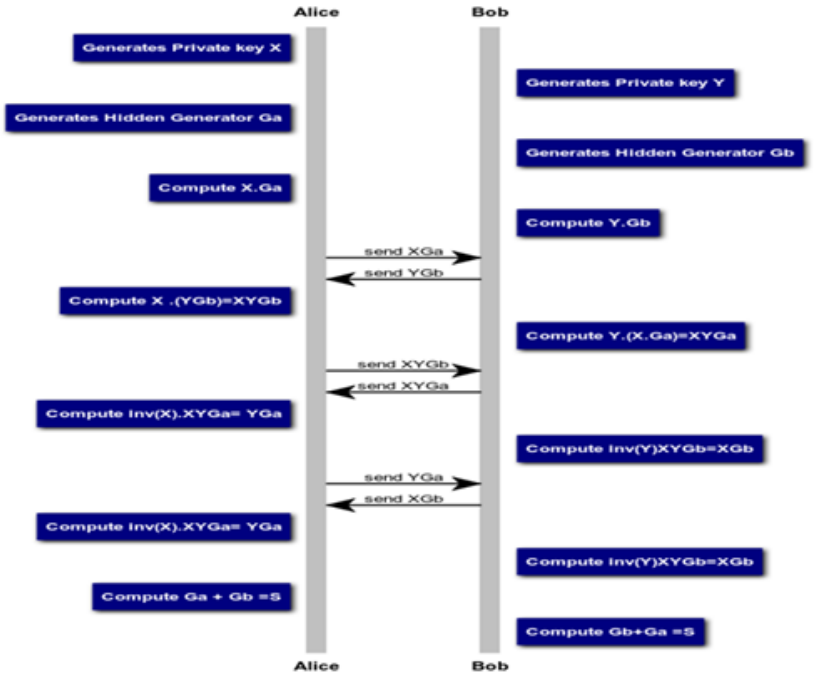


Fig 5: Shared Key generation using 2-hidden generator points

5.1.1 Generation of Shared Key:

After both the communicating parties are in know of G =Ga + Gb , a point on the curve, the following method can be adopted for adoption of a shared key:

G being a point on the curve will have x and y coordinates. Depending upon the curve choosen, the size of these coordinates can be 120, 160 , 192 bits etc.This being a scalar number , can act as a symmetric key between two parties, which can be used as a sessions key for various purposes including distribution of public keys or for encrypting a session. Message Authentication Code (MAC) which is key dependent ,

can also be generated using say x co-ordinate of the common generated point G. The shared key S can be used with any light weight symmetric cipher for achieving authentication.

5.2 Key generation using 1-Hidden Generator Point.

In this protocol, either of the communicating parties i.e Alice or Bob is supposed to have a hidden Generator point. Both the parties choose their respective private keys X and Y in the form of Scalar numbers. After performing scalar multiplication and inverse operations in a series of exchanges as illustrated in the figure 6., both the parties establish a common shared point i.e Ga .

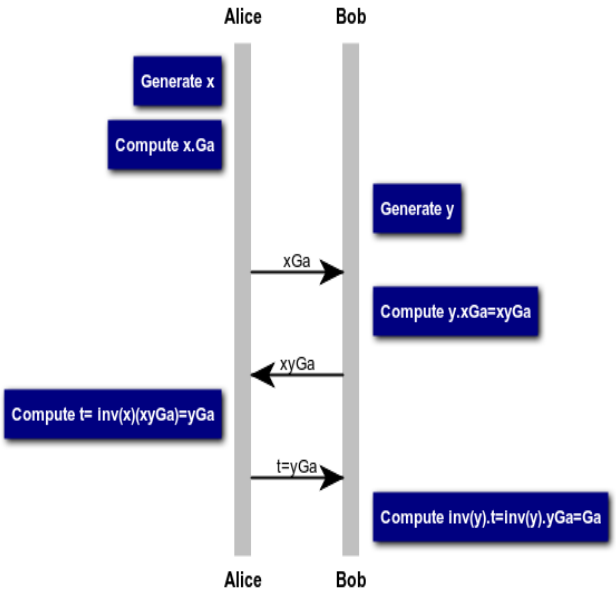


Fig 6 : Shared Key generation using 1-hidden generator point

5.3 Authentication of nodes:

After arriving at a common generator point G and shared key K, the following protocol steps can be adopted for authentication of nodes:

1. Node A calculates hash of its ID : Hash (IdA)
2. Node B calculates hash of its ID : Hash (IdB)
3. Node A calculates G.Hash (IdA), EK (IdA) and sends it to B
4. Node B decripts IdA by performing DK (IdA) and calculates G.Hash (IdA)
5. If G.Hash(IdA) calculated by node B at step 4 is same as G.Hash(IdA) of step 3 then node B authenticates node A .

The protocol can use simple Encryption (EK) and Decryption (DK) symmetric functions. Similar mechanism can be adopted for mutual authentication of nodes.

5.4 Simulation Outputs

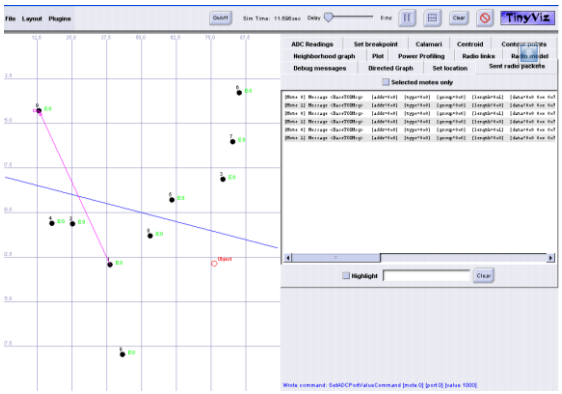


Fig.7: Tinyviz simulation of first protocol with 2-hidden generator point

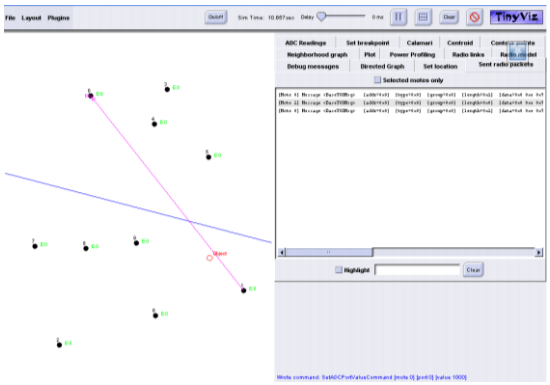


Fig.8: Tinyviz simulation of second protocol with 1- hidden generator point

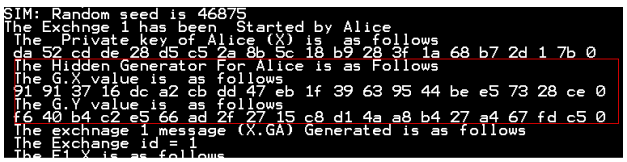


Fig.7(a): Hidden generator point at Alice

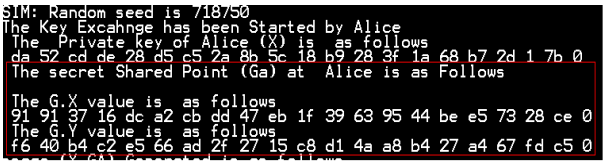


Fig.8(a): Hidden generator point of Alice

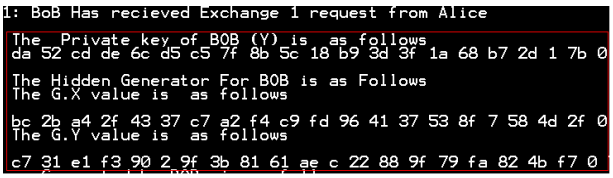


Fig.7(b): Hidden generator point at Bob

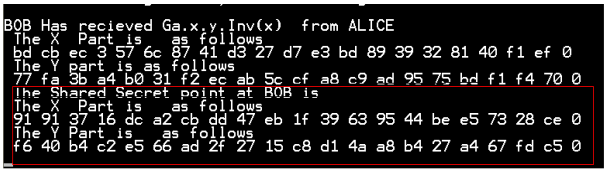


Fig.8(b): Shared hidden generator point with Bob

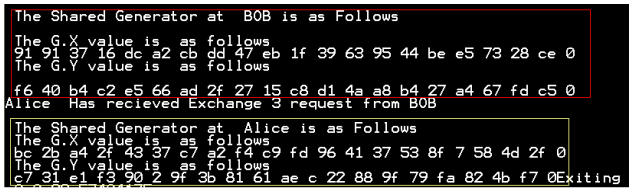


Fig.7(c): Exchange of hidden generator points between Alice & Bob

Both the protocols i.e 1-hidden generator point and 2-hidden generator point were implemented and simulated in Tiny OS[19]. A discrete event simulator TOSSIM[20] was used for simulating the NesC applications developed using TinyECC[21] for the concerned protocols. The applications were also ported on MICAZ hardware. A graphical user interface of TOSSIM, Tinyviz was used for capturing the exchanges between Alice and Bob. The simulation outputs for 2-Hidden Generator Point and 1-Hidden Generator are shown in Fig 7 and Fig 8 respectively.

6 Performance Benchmarking

The performance benchmarking of the key exchange protocol involving hidden generator points would be based on the following parameters:

1. Energy Consumption
2. Memory Consumption

3. Computational Time

In comparison to 2 -hidden Generator Points, the 1-hidden Generator Point algorithm has better performance in terms of Memory Consumption and defense against MIM as indicated in the Table 1.

Table 1: Comparative statement

S.No	Protocol	No of Exchanges	Scalar Multiplication	Point Addition	Inverse Operation	ROM	RAM	Defense against MIM
1	2 -hidden Generator Points	6	8	1	2	16482 Bytes	1890 Bytes	Yes
2	1-hidden Generator Point	3	4	-	2	15482 Bytes	1337 Bytes	Yes
3	ECDH	02	04	-	-	1487 bytes	1208 bytes	No

6.1 Energy Calculations

Energy Calculations would primarily depend on computational time taken for core ECC operations like Point Addition, Scalar Multiplication in addition to the voltage and current requirements. For calculation of energy we use E= V*i*t (joules) where V and i stand for voltage and current drawn respectively, t is the execution time for each operation . MicaZ node using Atmel AT Mega 128 L is powered by 02 AA batteries. Assuming voltage of 3 V for 02 AA batteries, and a maximum load current of 19.7 mA, the energy calculations for each operation are indicated in the Table 2.

For the purpose of capturing computational time of various key ECC operations like Point addition , Scalar Multiplication a basic setup was established using MicaZ, MIB520(programming board). A nesC program was developed for sending the time message to a TinyOS Serial Forwarder. These packets were sent on serial port through a MIB 520 programming board. The packets captured by the serial forwarder were transported to a java application.

Table 2: Energy calculations

Operations	Avg Time Taken (Seconds)	1-hidden Generator Point	Energy Consumption (1-Hidden Generator) (milli Joules)	2-hidden Generator Point	Energy Consumption (2-Hidden Generator) milli Joules	ECDH	Energy Consumption (ECDH) milli Joules
Scalar Multiplication	1.78	4*1.78 = 7.14 secs	422.38	8*1.78 14.29 sec	844.75	4*1.78 = 7.14 secs	422.38
Inverse Operation	0.11	2*0.11 =0.23 secs	14.06	2*0.11 0.22sec	13.49	nil	nil
Point Addition	1.787	NIL	nil	1.78sec	105.61	nil	nil
TOTAL		7.37 secs	436.44	16.29	963.85	7.14secs	422.38

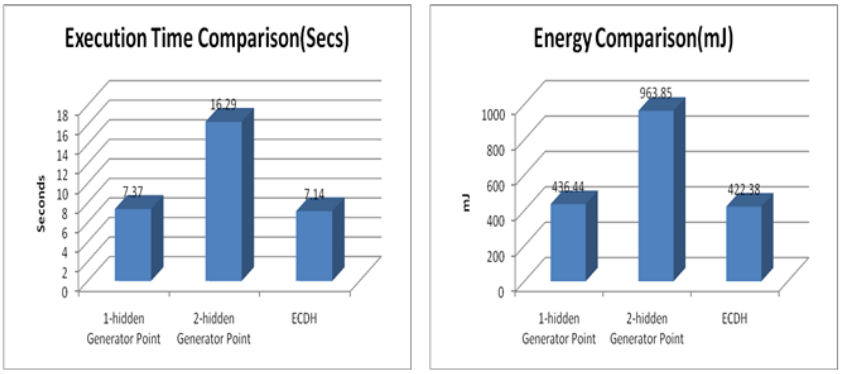


Figure 9: Execution time and energy comparison

7 Conclusion

A Hidden Generator Point in ECC can be useful to thwart Man-In-The-Middle Attack in a resource constraint WSN Network. The conventional ECDH used for generating shared keys does not offer such an advantage. The performance Benchmarking matrix of protocols discussed in the paper based upon hidden generator concepts clearly indicates that resource utilization of 1-hidden generator point protocol is comparable to that of ECDH with an added advantage of offering protection against Man-in-The-Middle attack. The energy consumption which inter alia depends upon computational time of each operation was found to be 436.44mJ in case of 1-hidden generator point as against 422.38mJ of ECDH protocol and 936.85mJ of 2-hidden generator points. The paper illustrates the generation of shared keys along with a simple authentication protocol based upon hidden generator. The concept can be further exploited in developing energy efficient security application for low power devices used in smart cities.

REFERENCES

[1] Adrian Perrig, John Stankovic, David Wagner.," *Security in wireless sensor networks",* Communications of the ACM, vol 47,no. 6, pp 53-57, June 2004.

[2] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victor Wen and David E.Culler., "*SPINS: Security protocol for sensor networks*", in proceedings of 7th International conference on mobile networking and computing, 2001, vol 8, no.5, pp 189-199, 2001.

[3] Xiaojiang Du, Hsiao-Hwa chen. , "Security in Wireless Sensor Networks", IEEE Wireless Communications, August 2008.

[4] N.Gura, A.Patel, A. Wander, H.Eberele and S. Shantz.," *Comparing Elliptic Curve Cryptography and RSA on 8 bit CPU*".,in 2004 workshop on cryptographic hardware and embedded systems, August 2004.

[5] Zhu, S.,Setia,S., and Jajochia, S., "*LEAP: Energy efficient security mechanism for large-scale distributed sensor networks*", In the proceedings of the conference on computer and communications security ,03,ACM Press, Washigton DC 2003, pp 62-72.

[6] Escheanauer, L., and Gilgor, U.D., "*A Key management scheme for distributed sensor networks*.", in the proceedings of the conference on computer and communications security "02", Washington DC 2002 pp 41-47.

[7] Chan, H,Perrig, A., and Song,D," *Predistribution schemes for sensor networks*", in the proceedings of IEEE security and privacy symposium ,IEEE Computer society press, Loss Alanos 2003, pp 197-213.

[8] R. Watro, D. Kong, S.Cuti, C.Gardiner, C.Lynn and P. Kruus., " *TinyPK: Securing sensor networks with public key technology*." , in the proceedings of 2nd ACM workshop on security of adhoc sensor networks (SASN 04), pp 59-64, New York, ACM press.

[9] Q.Huang, J.Cukier, H.Kobayashi, B.Liu and J.Zhang., " *Fast authenticated key establishment protocols for self-organizing sensor networks*", in the proceedings of the 2nd ACM international conference on WSN and applications , pp 141-150,ACM Press, 2003.

[10] ZHAO Xin, EANG Xia-dong., "*Design and implementation of the Hybrid broadcast authentication protocols in WSN*", published in 2nd international conference on future generation communication and networking, 2008.

[11] Xu Huang, et al. "Fast Scalar multiplication for Elliptic curve cryptography in Sensor Networks with Hidden Generator point", 2010 International conference on Cyber-enabled distributed Computed and knowledge Discovery.

[12] Ravi Kishore et al. "*High Performance Scalar Multiplication for ECC*. In 2013 International Conference on Computed Communication and Informatics (ICCCI-2013, Jan 04-06, 2013 Coimbatore, INDIA)

[13] Ravi Kishore Kodali et al. "Implementation of ECC with Hidden Generator Point in *Wireless Sensor Network*". 978-1-4799-3635-9/14 @ 2014 IEEE.

[14] D. Hankerson et al. "*Guide to Elliptic Curve Cryptography*" Springer, 2004

[15] Bernard Menzes "*Network Security and Cryptography*", Cengage Learning

[16] Ioannis Chatzigiannakis et al. "*Elliptic Curve Based Zero Knowledge Proofs and their Applicability on Resource Constraint Devices*". Ict-2010-258885(SPITFIRE)

[17] Yi Jiang et al. "*Cluster Based Strategies for Public Key Authentication in Wireless Sensor Networks*". Chinese journal of Sensors and Actuators, Volume 20,6,2007.

[18] www.certicom.com

[19] TinyOS. http:// www.tinyos.net

[20] P. Levis, N. Lee, M. Welsh and D. E. Culler. et al *TOSSIM : "Accurate and Stable Simulation of Entire TinyOS Applications"*. SenSys 2003

[21] A. Liu and P. Ning et al. Tiny ECC: "*A Configurable Library for Elliptical Curve Cryptography in Wireless Sensor Networks"* IPSN 2008

TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

Design and Development of Lower Limb Chair Exercise Support System with Depth Sensor

[1]Toshiya Watanabe, [2]Naohiro Ohtsuka, [3]Susumu Shibusawa, [4]Masaru Kamada and [5]Tatsuhiro Yonekura

[1]Graduate School of Science and Engineering, Ibaraki University, Hitachi, Japan;
[2]East Japan Institute of Technology Co., Ltd;
[3,4,5]School of Engineering, Ibaraki University, Hitachi, Japan;
[1]12nd307a@hcs.ibaraki.ac.jp; [3]sibusawa@mx.ibaraki.ac.jp; [4]m.kamada@mx.ibaraki.ac.jp;
[5]yone@mx.ibaraki.ac.jp

ABSTRACT

Sustaining lower limb functionality is extremely important in the preventative care of the elderly. Chair exercise, in which the exerciser sits on an ordinary chair, offers a way for seniors with little physical strength to exercise without a great deal of effort. Meanwhile, Microsoft's Kinect sensor that is capable of detecting human motion without the subject having to wear any kind of a special marker are becoming widely available. Exploiting this new sensor technology, this paper describes the design and development of a prototype lower limb chair exercise support system. The system supports five different chair exercises designed to strengthen the lower limbs, recognizes and evaluates exercises based on 3D position data and joint angles for each joint obtained from the Kinect sensor. The system illustrates how to do the exercises by voice instructions and model images, and superimposes the muscles used onto an image of the exerciser in real time. The system also provides exercise assessment results and advice by voice and text. In a series of trials involving seven elderly subjects in their late 70s and early 80s, an overall average recognition rate of 89% was obtained for the five exercises. Feedback was obtained through a questionnaire given to subjects ranging in age from 50 to 65, which highlighted a number of issues that should be addressed to improve the effectiveness of the system.

Keywords: chair exercise; lower limb; elderly population; depth sensor; exercise system design; preventative care.

1 Introduction

Japan is rapidly becoming a super-aged society with over 25% of the population aged 65 or older, and the number of elderly requiring long-term care is rapidly increasing [1]. Good preventative care that enables as many people as possible to live independently in sound health is thus becoming increasingly important. The goals of preventative long-term care are to stave off the need for long-term care as long as possible, to prevent one's condition from deteriorating in cases where care is already required, and to enhance peoples' quality of life by improving their mental and physical fitness [2]. Preventative care is typically overseen by a physical therapist or other professional, but obviously there are not enough healthcare professionals to deal with the entire elderly cohort in the population. This is where support systems come in that can effectively stave off or prevent nursing care among seniors.

DOI: 10.14738/tnc.34.1387
Publication Date: 21th August 2015
URL: http://dx.doi.org/10.14738/tnc.34.1387

For the purposes of preventative care, sustaining lower limb functionality is critically important for enabling people to keep their balance, to continue walking, and to prevent falls that often lead to bedridden debilitation, and considerable interest has focused on chair exercise as an excellent way to strengthen the lower limbs. Exercise done from a sitting position in a chair is ideally suited for older people who lack stamina, because it takes the burden off of their ankles and knee joints so they can strengthen leg muscles with relatively little effort. Not surprisingly, chair exercise is used extensively in many nursing homes and elder care facilities.

Preventative care solutions and rehabilitation have become increasingly important as society ages, and have prompted a surge of research interest into systems that might assist rehabilitation and long-term preventative care. In the work of Matsukuma et al. on stand-up rehabilitation for older people [3], these authors created a game out of the repetitive stand-up and sit-down exercise that is widely prescribed to rehab patients, and they found that the addition of this entertainment aspect markedly improved the persistence and enthusiasm of patients. Hashimoto et al. [4] developed a rehabilitation site [5] that patients can access over the Internet from the comfort of their own homes enabling them to pursue rehabilitation at their own pace.

Meanwhile, Microsoft's Kinect has become readily available for tracking user movement and posture. Kinect features an RGB camera and a depth sensor that captures user movements and 3D positions for up to 20 different joint points without the need of special markers or putting a sensor on the body. Moreover, because Kinect is a compact device engineered for use in affordable games, it is already being widely deployed and used in ordinary homes, and is expected to see many new applications in medical, disability support, and other healthcare-related areas [6-10].

In this paper, we present the design methodology, development and evaluation of a lower limb chair exercise support system with the Kinect depth sensor. The system supports five different chair exercises for strengthening the lower limbs, and each exercise is recognized and evaluated using 3D positioning and joint angle data captured by Kinect. The user is first shown how to do the exercises by voice instructions and model images, then the user's use of muscles in performing the exercise is superimposed on the image in real time. The system also provides assessment results and advice by voice and text. Toe and heel exercises are implemented by estimating the toe position from Kinect depth images.

The rest of the paper is organized as follows: In Chapter 2, we review other relevant research, and Chapter 3 details the design of the Kinect-based chair exercise support system. A prototype chair exercise support system is described in Chapter 4. In Chapter 5, we evaluate the prototype system through a series of trials and consider the implications of the trial results, and Chapter 6 provides a brief recap of the study.

2 Related Work

Microsoft Kinect is a compact affordable device with built-in RGB camera, depth sensor, and other features designed specifically for use in game consoles. The ability of Kinect to capture 3D positions and movement without the need of special markers or putting a sensor on the body has attracted enormous interest among researchers for developing medical and disability support systems and devices. For example, Mentiplay et al. studied the reliability and effectiveness of Kinect for evaluating static foot postures [6], and found that Kinect is much more reliable than prevailing visual indicators for assessing static foot posture, and yielded results that are approximately equivalent to those derived from 3D motion analysis. Metcalf et al. examined markerless motion capture and measurement of hand kinematics [7], and proposed using Kinect for the measurement and analysis of

complex and hard-to-measure finger movements. These authors also emphasized that Kinect is far easier to apply in home-bound upper limb rehabilitation schemes than marker-based motion capture.

Obdrzalek et al. sought to evaluate the accuracy of pose measurements using Kinect in the context of exercise for an elderly population [8]. Meanwhile, Erazo et al. developed a rehabilitation system using Kinect based on the magic mirror game for people with upper limb impairments, and evaluated the effectiveness of the system [9]. Watanabe et al. showed the design of the lower limb chair exercise support system with Kinect, developed a prototype system, and conducted trials for elderly subjects [11, 12].

In recent years, we have seen growing interest in serious game or gamification, where game thinking and game mechanics are used in non-game contexts such as to improve user engagement in learning and training [13, 14]. In the work on stand-up rehabilitation, Matsukuma et al. developed a game that improves patient motivation and persistence to do the rehabilitation exercise which is strongly recommended [3]. Another study in this vein is a 3D virtual rehabilitation game called Dance2Rehab3D by Bruckheimer et al. [15]. This upper limb rehabilitation support system uses joint angle data provided by Kinect in an interactive 3D environment based on an aquarium as a motif. Experimental results indicate that this approach is effective at strengthening upper limbs and reducing fatigue of stroke patients.

Asakura et al. have developed a shoulder rehabilitation support system that combines image-based 3D pose measurement data with myoelectric sensing data, and emphasizes presentation of information on the user's current condition and what the user might do to improve his or her fitness [16]. Pei et al. published a study on a robotic lower limb rehabilitation musculo-skeletal model [17] featuring a lower limb rehabilitation scheme that exploits robotic capability to track complex orbits, and compared the rehabilitation efficiency of their approach with existing methods. Fasola et al. [18] used the socially assistive human-robot interaction for older adult physical exercise, and showed that elderly users have a strong preference for the relational robot in terms of enjoyableness.

Igarashi et al. devised a wearable light-emitting sensor suit as a way of supporting lower-limb motion perception by visual means [19]. By measuring myoelectric potential and joint angles associated with motion, these authors are able to show muscle activity in real time for any location site of lower limb muscle activity on the body surface.

Several commercial rehabilitation systems based on Kinect have been developed [20-22]. However, these systems show neither the design methodology on support systems nor the quantitative evaluation, and we are not aware of any comprehensive study for such systems. A rehabilitation system to publish a part of method acquiring patient motion does not show the quantitative evaluation on lower limb chair exercise [23-26].

We show the design methodology, development and evaluation of a lower limb chair exercise support system with the Kinect depth sensor in this paper.

3 System Design

3.1 Exercise Support System Requirements

Desirable exercise support system has functions to [9, 11]:

(1) provide exercise guidance using images, text, and voice instructions,
(2) verify a user's posture and form while exercising,

(3) evaluate the exerciser's efforts and offer advice in the form of video, text, and speech,
(4) motivate users and encourage users to continue exercising,
(5) save exercise records,
(6) enable the therapist to modify or tailor the application system to the exerciser's fitness level, and
(7) include game elements to motivate and encourage users.

Requirements (1) and (2) show images of the exerciser together with instructions on how the exercise is done correctly, so this makes the user aware of what must be done to adhere to the proper form of the exercise. Video and audio showing the effectiveness of the exercises in everyday life situations gives users a better understanding of the specific benefits of the exercises.

Requirements (3), (4), and (7) help users understand the benefits of the exercise and improve their motivation by evaluating their form and providing instant feedback. Encouraging and motivating users to stick with the exercise program is extremely important. Negative expressions might have the contrary effect of discouraging elderly users, so negative feedback is avoided in an effort to bring users back to the program and persist with the exercise.

3.2 System Overview

In this work, a prototype chair exercise support system was developed that employs a Kinect sensor to capture 3D position data of the user that is used to assess and critique the form of the user performing the exercise. Figure 1 shows a schematic overview of the system. As one can see from the figure, the Kinect sensor is positioned directly in front of the user who is seated in a chair, and video data of the user captured by Kinect is sent to a computer for processing. Audio and text feedback are provided in real time that accompany video images and exercise data that are displayed on a screen or monitor.

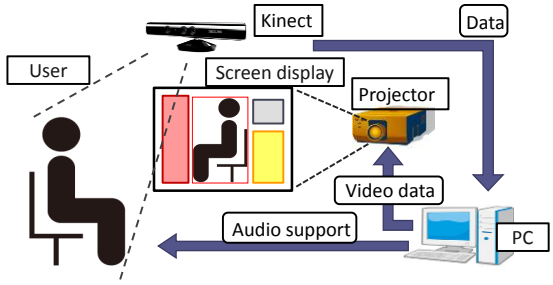


Figure 1: System overview

Table 1: System functions

	Function	Description
(1)	Display images of the exerciser	Display RGB image captured by Kinect.
(2)	Display muscles used	Display image of muscles used as the user exercises superimposed on the RGB image.
(3)	Display effects of exercises	Visually display scenes illustrating effects of exercise in everyday life.
(4)	Evaluate exercises	Evaluate exercise on a 4-point scale: excellent, good, fair, or poor. Evaluation results are presented in voice and text for each exercise.
(5)	Save exercise records	Record support system use data, joint angle and position data for each exercise, evaluation data.

3.3 Exercise System Functions

Table 1 lists the primary functions of the system. As outlined in Section 3.1, the basic functions of the exercise support system are (1) to display images of the exerciser, (2) to explain in images and text exactly how to do the exercises, (3) to highlight the beneficial effects of the exercises, (4) to display evaluation results of the exercises in speech and text, and (5) to record and store data for each exercise.

3.4 Exercise Recognition Method

The system supports five different exercises for strengthening the lower limbs that are done while sitting in a chair: toe lifts, heel lifts, knee extensions, thigh lifts, and leg-open exercises. Just two criteria are used to identify or recognize these different exercises: 3D position data of various joints captured by Kinect, and joint angles derived from the 3D Kinect data. For the purposes of exercise recognition,

here we employ the following abbreviations for joints: toe = To, ankle = A, knee = K, hips = Hi, hip on open leg side = m_Hi, and hip on stationary leg side = s_Hi.

3.4.1 Toe-Lift Exercise

The toe-lift exercise is done by raising the toes while leaving the heels planted firmly on the ground. If the toe position estimated from the depth image is represented by To and the ankle joint position derived from Kinect is represented by A, the toe-lift exercise is recognized using the ankle joint angle ⍰A between a straight line connecting A-To and a horizontal line and the toe height yT. The ankle joint angle and toe height used to recognize the toe-lift exercise are illustrated in Figure 2.

Changes in the ankle joint angle and toe position when doing the toe-lift exercise are shown in Figure 3. As one can see in the figure, the value of the ankle joint angle decreases as the toe position increases when doing the exercise.

(a) Joint angle (b) Toe position

Figure 2: Joint angle and toe position used to recognize the toe-lift exercise.

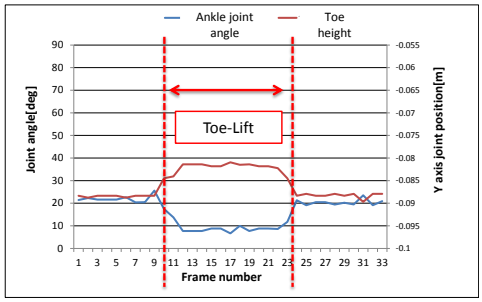
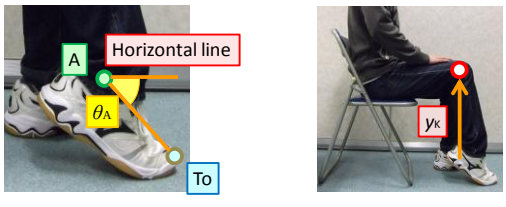


Figure 3: Angle and position change during the toe-lift exercise.

3.4.2 Heel-Lift Exercise

The heel-lift exercise is done by placing weight on the balls of the feet and raising the heels. If the toe position estimated from the depth image is represented by To and the ankle joint position derived from Kinect is represented by A, the heel-lift exercise is recognized using the ankle joint angle θ_A between a straight line connecting A-To and a horizontal line and the ankle height y_K. The joint angle and joint position used to recognize the heel-lift exercise are illustrated in Figure 4.

Changes in the ankle joint angle and knee position when doing the heel-lift exercise are shown in Figure 5. One can see in Figure 5(a) that the ankle joint angle increases as you do the heel-lift exercise. While the ankle height does not change that much, the knees are elevated to a moderate degree, as shown in Figure 5(b). The system recognizes the heel-lift exercise by focusing on the changing ankle joint angle θ_A and knee joint height y_K of the exerciser, and determines that the exercise is being performed when these values satisfy the relevant thresholds.



(a) Joint angle (b) Joint position

Figure 4: Joint angle and position used in the heel-lift exercise

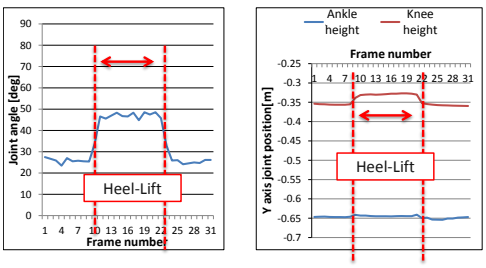
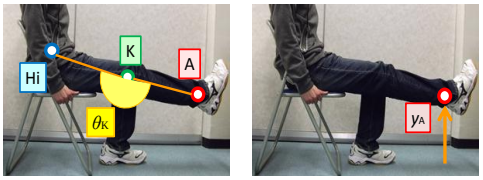


Figure 5: Changing joint angles and positions when doing the heel-lift exercise

3.4.3 Knee-Extension Exercise

The knee-extension exercise is done by rising one leg at a time and extending or straightening the knee. Where Hi, K, and A represent hip, knee, and ankle, respectively, the knee-extension exercise is recognized using the knee joint angle ?K formed from these three joints and the ankle height yA. The joint angle and joint positions used to recognize the knee-extension exercise are illustrated in Figure 6.

Changes in the knee joint angles and ankle positions when doing the knee-extension exercise are shown in Figure 7. One will note in the figure that the value of the knee joint angle increases as the ankle position is elevated when doing the knee-extension exercise. Thus, the system recognizes the knee-extension exercise by focusing on the changing knee joint angle ?K and ankle joint height yA of the exerciser, and determines that the exercise is being performed when these value satisfy the relevant thresholds



(a) Joint angles (b) Joint positions

Figure 6: Joint angles and positions used to recognize the knee-extension exercise

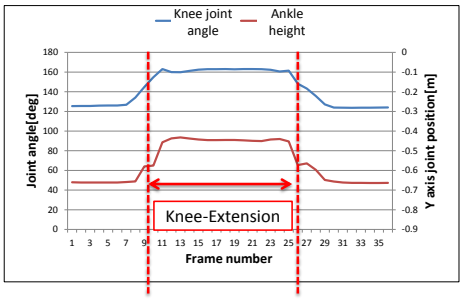
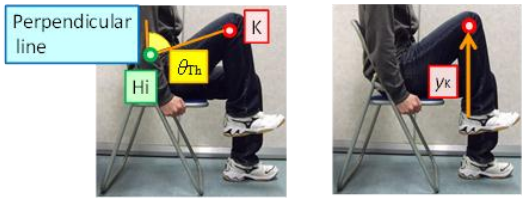


Figure 7: Changes in joint angle and position when doing the knee-extension exercise

3.4.4 Thigh-Lift Exercise

The thigh-lift exercise is done by pulling your knees in toward your chest and elevating your thighs. If knees and hips are represented by K and Hi, respectively, the thigh-lift exercise is recognized by using the thigh joint angle ?Th derived from a straight line connecting Hi and K and a perpendicular line, and by the height of the knees yK. The joint angles and joint positions used to recognize the thigh-lift exercise are shown in Figure 8.

Changes in the thigh joint angle and knee joint position when doing the thigh-lift exercise are illustrated in Figure 9. One can see that the value of the thigh joint angle decreases as the knee position is elevated when doing the exercise. The system recognizes the thigh-lift exercise by focusing on the changing thigh joint angle ?Th and knee joint height yK of the exerciser, and determines that the exercise is being performed when these values satisfy the relevant thresholds.



(a) Joint angles (b) Joint positions

Figure 8: Joint angles and positions used in the knee-lift exercise

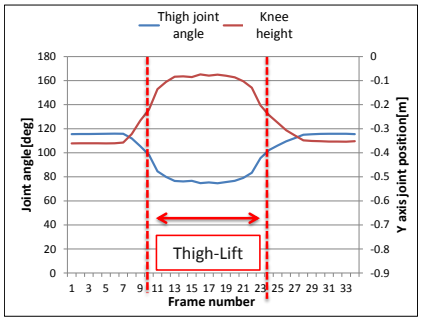
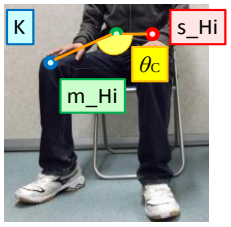


Figure 9: Changing joint angles and positions during the thigh-lift exercise.

3.4.5 Leg-Open Exercise

The leg-open exercise is done by opening the hip joint by moving one leg laterally to the side. Where K = knees, m_Hi = hip on the leg-open side, and s_Hi = the hip on the opposite side, the leg-open exercise is recognized by using the hip joint angle θ_C created by straight lines connecting m_Hi and K and connecting m_Hi and s_Hi, and position x_K in the lateral direction the knee is moving. The joint angles and joint positions used to recognize the leg-open exercise are shown in Figure 10.

Changes in the hip joint angle and knee joint position when doing the leg-open exercise on the right foot are shown in Figure 11. One can see in the figure that the value of the hip joint angle increases while the position of the right foot moves laterally along the Kinect X axis, that is, to the right from the exerciser's perspective. The system recognizes the leg-open exercise by focusing on the changing hip joint angle θ_C and knee joint position x_K of the exerciser, and determines that the exercise is being performed when these values satisfy the relevant thresholds.



(a) Joint angles

(b) Joint positions

Figure 10: Joint angles and positions used to recognize the leg-open exercise

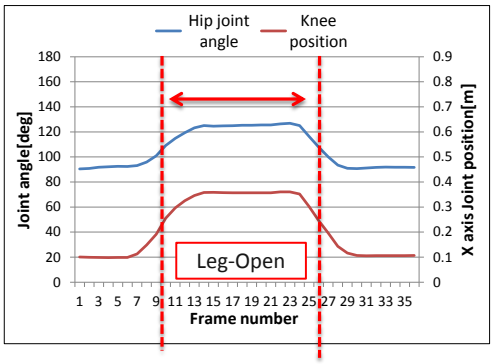


Figure 11: Changing joint angles and positions for recognizing the leg-open exercise for the right foot

A summary of the joint angles and joint positions used to identify five exercises is presented in Table 2.

Table 2: Exercise joint angle and joint position summary

Exercise	Joint angle	Joint position
Toe–lift	Ankle joint	Toe height
Heel–lift	Ankle joint	Ankle height
Knee–extension	Knee joint	Ankle height
Thigh–lift	Thigh joint	Knee height
Leg–open	Hip joint	Knee horizontal position

3.5 Evaluating Exercises

Each of the exercises was evaluated based on the joint angles and joint positions of the exercisers. Here we introduce two thresholds, θ_1 and θ_2, to subdivide the joint angle θ of the exerciser. In other words, the two thresholds are used to divide joint angle θ of the exerciser into three regions, which are assigned evaluations 0, 1, and 2.

$$\text{Joint angle evaluation 0:} \quad \theta < \theta_1 \tag{1}$$

$$\text{Joint angle evaluation 1:} \quad \theta_1 \le \theta < \theta_2 \tag{2}$$

$$\text{Joint angle evaluation 2:} \quad \theta_2 \le \theta \tag{3}$$

Next, we introduce two thresholds, w_1 and w_2, with respect to joint position w (w = x, y, z) of the exercise being done by the exerciser for the rectangular coordinate system (x, y, z). The two thresholds are used to divide joint position w of the exerciser into three regions, which are assigned evaluations 0, 1, and 2.

$$\text{Joint position evaluation 0:} \quad w < w_1 \tag{4}$$

$$\text{Joint position evaluation 1:} \quad w_1 \le w < w_2 \tag{5}$$

$$\text{Joint position evaluation 2:} \quad w_2 \le w \tag{6}$$

Based on the two thresholds for these two criteria—joint angle and joint position—an exerciser's data can be divided into nine regions. These nine regions are evaluated according to a four-value scale as shown in Table 3, so an exercise can be evaluated as excellent, good, fair, or poor. Referring to the table, one can see that if the user gets "2" for both joint angle and joint position, the exercise is evaluated "excellent." If a user gets a "1" and a "2" for the two criteria, the exercise is evaluated "good." If a user gets "1" for both joint angle and joint position, the exercise will be evaluated as "fair," and if a user gets "0" for either joint angle or joint position, the evaluation for the exercise will be "poor."

Table 3: Exercise joint angle and joint position summary.

Joint angle / Joint position	Evaluation "0"	Evaluation "1"	Evaluation "2"
Evaluation "0"	poor	poor	poor
Evaluation "1"	poor	fair	good
Evaluation "2"	poor	good	excellent

4 Implementing a Prototype System

A prototype system was implemented using Visual C++ 2008, Open CV, and Open NI. The PC used in this work had an Intel(R) Core(TM) i5-2400 CPU with 4 GB of memory. The processing speed was more than adequate to run the prototype implementation without any delays or problems. Figure 12 is a photo of the system in operation. One will observe in the photo that the system is set up toward the back of the room on the right, and the exercise situation is displayed on the screen to the left. The Kinect sensor is deployed directly in front of the screen to assist the user in the foreground do leg exercises.

Figure 13 shows typical screenshots of support system output. The screen on the left is the RGB image of the exerciser with superimposed muscles used by the exerciser. The screen at the upper right provides an example of how each exercise should be done, and the screen at the lower right illustrates the beneficial effects of each exercise. In addition, there is a textual assessment of the effects of each exercise in the lower left hand corner of the screen.

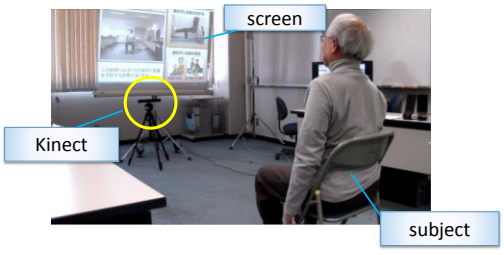


Figure 12: Prototype system in operation

Figure 13: Chair exercise support system output screen.

5 Prototype System Trials and Considerations

To verify the effectiveness of the prototype system, we conducted trials to assess the recognition accuracy of the chair exercise system and also to gauge the response of trial subjects. Figure 14 shows the experimental setup. The Kinect sensor is installed at a height of 72 cm above the floor, while the subject sits on a folding chair (the seat is 40 cm above the floor) positioned 2.5 meters from the Kinect. To enhance the recognition accuracy, the subjects kept their shoes on during the trials and loose trouser legs were secured with rubber bands around the ankles. This not only improved the toe position estimation accuracy, it also reduced the adverse effects of loose trouser legs on the Kinect ankle joint position estimation and improved the ankle joint estimation accuracy.

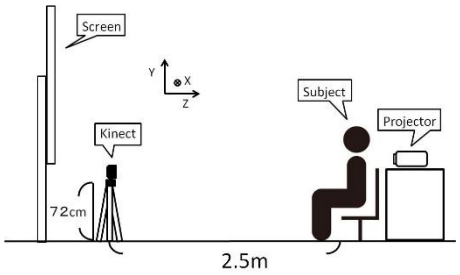


Figure 14: Trial setup

5.1 Chair Exercise Recognition Accuracy Trials

These trials were conducted to assess the recognition accuracy of lower limb chair exercises intended for seniors, and to verify the effectiveness of the proposed exercise recognition method. Working with seven subjects ranging in age from latter 70s to early 80s, we asked the subjects to perform five repetitions of each of the five lower limb exercises on the left and the right. As the subjects did the exercises, we recorded the number of times they executed the exercises correctly, the changes that occurred in the joint angles and positions as they did the exercises. The exercises were done in the following sequence (1) toe-lift, (2) heel-lift, (3) knee-extension, (4) thigh-lift, and (5) leg-open, starting on the right foot and with a three-beat interval between each exercise.

Average recognition rates for the chair exercises are shown in Figure 15. If the rate of correct execution is the number of times an exercise is done correctly divided by the number of times the exercise was attempted, then the average recognition rate is the total correct rate of each exercise divided by the number of exercisers

$$\text{Average recognition rate for each exercise}(\%) = \frac{\text{Total correct rate of each exercise}}{\text{Number of subjects}} \times 100 \quad (7)$$

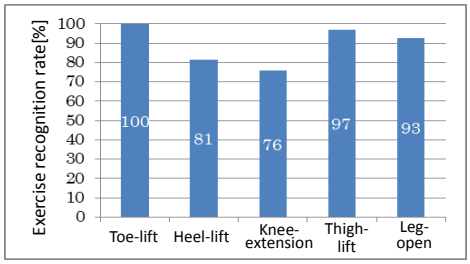


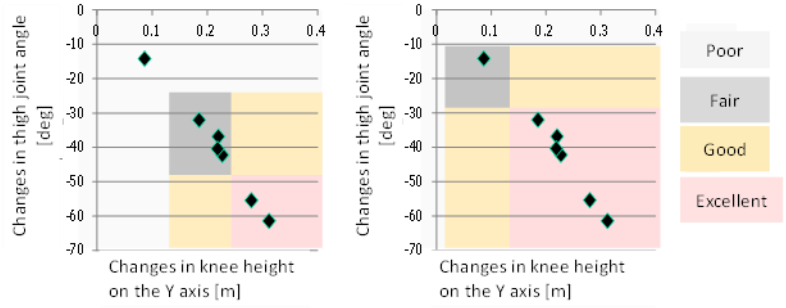
Figure 15: Average recognition rate of the chair exercises

As one can see in Figure 15, the recognition rate for three of the exercises—toe-lift, thigh-lift, and leg-open—is over 90%, and the average recognition accuracy for all five exercises is 89%. The accuracy of the prototype chair exercise system is thus more than sufficient to support at least three of the chair exercises.

Yet, the recognition accuracies for the heel-lift and knee-extension exercises were only 81% and 76%, respectively. The low recognition rate for the heel-lift exercise is attributed to inadequate tracking of the foot region by Kinect, which results in low accuracy for the foot joint position. One thing we could do to improve the accuracy would be to have seniors warm up with a stepping exercise before starting the chair exercise; this would improve the foot joint position accuracy captured by Kinect. The poor recognition rate for the knee-extension exercise is attributed to the fact that the knee joint is obscured when the subject's foot comes up, which causes Kinect's estimated joint position to be off. We are now reconsidering the exercises to come up with a solution to these problems.

5.2 Threshold Trials of Thigh-lift

Figure 16 shows changes in the thigh joint angles and knee heights for the seven subjects doing right thigh-lift exercises. As one can see in the figure, four colors show four evaluation levels.



(a) Threshold setting for healthy seniors **(b) Threshold setting for seniors with lower limb disabilities**

Figure 16: Changes during the thigh-lift exercise

In Figure 16, the thigh joint angle of the subject with the greatest amount of change in the thigh joint angle is reduced by an average of 61 degrees, and the knee height of the subject is elevated by 31 cm. These values are practically the same as those for a young person doing the same thigh-lift exercise, which reveals that there are at least some seniors close to 80 years old who perform the same exercises as young people. This also illustrates the importance of identifying exercise evaluation thresholds for senior citizens who are in good physical condition that will help them sustain their physical functions. Figure 16(a) shows threshold settings that would be appropriate for healthy physically fit seniors.

Where $\theta = -\theta_{Th}$ with respect to thigh joint angle θ_{Th}, joint angle evaluations for two thresholds θ_1 and θ_2, where $\theta_1 < \theta_2$, are set according to conditions (1), (2), and (3) of Section 3.E. Similarly, joint position evaluations for two thresholds y_1 and y_2 (where $y_1 < y_2$) for the knee height $w = y_K$ are set according to conditions (4), (5) and (6). So, for example, the thigh-lift exercise thresholds for seniors in good physical condition shown in Figure 16(a) are set as follows:

$$\theta_1 = 24 \text{ deg}, \ \theta_2 = 48 \text{ deg}, \ y_1 = 13 \text{ cm}, \ y_2 = 24 \text{ cm}$$

Now assuming that the subject with the least amount of change in the thigh joint angle is Subject E, the thigh joint angle for this subject performing the exercise is reduced by an average of only 14 degrees while the knee height of the subject is elevated by 9 cm. The changes in joint angle and joint position when Subject E performs the thigh-lift exercises are shown in Figure 17. The ability to set the

thresholds at more modest levels for this elderly individual would provide easier exercises that this user could accomplish. Figure 16(b) illustrates the threshold settings that might be appropriate for this less fit individual.

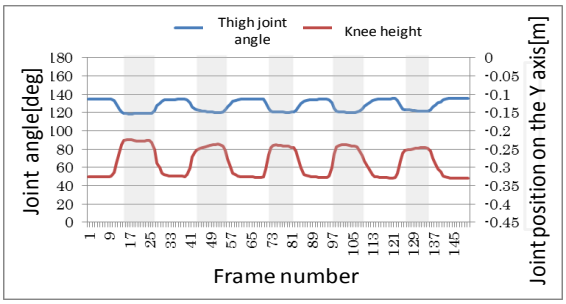


Figure 17: Changes in joint angle and joint position when Subject E does thigh-lift exercises.

The thigh-lift exercise thresholds for seniors requiring the easier exercises shown in Figure 16(b) are set as follows:

$$\theta_1 = 11 \text{ deg}, \ \theta_2 = 28 \text{ deg}, \ y_1 = 2\text{cm}, \ y_2 = 13 \text{ cm}$$

Currently, these thresholds values must be set by the system developer.

5.3 Chair Exercise Support System Evaluation Trials

To assess the effectiveness of the prototype chair exercise system's support functions, we asked the subjects to fill out a questionnaire evaluating the system. Five male subjects ranging in age from 50 to 65 worked out on the exercise support system, then evaluated the system by responding to a series of questions on a scale of one to five. For the purposes of this trial, the exercises were done in the following sequence (1) toe-lift, (2) heel-lift, (3) knee-extension, (4) thigh-lift, and (5) leg-open, starting on the right foot and each exercise was sustained for 5 seconds.

Appendix A.1 shows the questions that were asked on the questionnaire after using the system. The average points received for each item on the questionnaire are shown in Figure 18. The subjects were asked to respond to each item on a scale from one (strongly disagree) to five (strongly agree). The average for all items on the questionnaire was 4.3 points. The average score was 4.8 on both Question 1 (Were you able to understand the five types of chair exercises provided by the system?) and on Question 10 (Do you think there is no need to learn a lot preliminary stuff before using the system?). It is clear from these responses that the chair exercises supported by the prototype system are fairly easy to understand and perform.

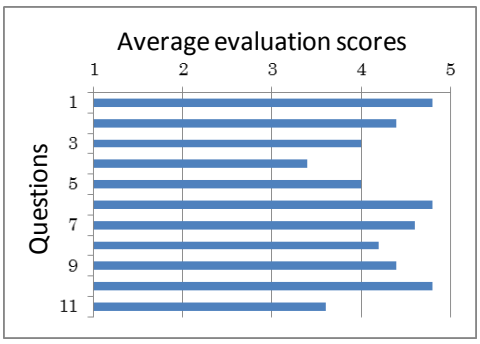


Figure 18: Evaluation trial results

In response to Question 4 (Were you aware of which muscles are used when doing the exercises?), the average score was 3.4, which was fairly low. We attribute this low rating to the fact that the subjects were so engrossed in doing the chair exercises correctly that they did not have time to think about the muscular effects. In order to promote exercises that encourage users to use their muscles correctly, we would have to provide support functions and exercise programs that are designed to draw attention to the muscles and how they are used.

The questionnaire also provided space for the subjects to offer their own free assessments of the system, and several of the subjects made the observation that "an exercise program you could use often or everyday would be especially useful." Clearly, this would require a number of different exercise programs tailored for different fitness objectives. For example, by offering a number of different programs to help strengthen not just the lower limbs but other parts of the body, this would go a long way toward stimulating continuity and spontaneity of users. By working together with physical therapists and other healthcare professionals, it should be possible to create exercise programs that are even more effective than the prototype system we present here.

Several of the subjects also commented that "compared to how I lived my life in the past, I want to figure out a way to stay physically active and healthy." We found that seniors derive a sense of security and joy when they are physically fit, and this is clearly linked to their persistence and motivation to continue exercising. An additional display function enabling users to compare their own past data with their current exercise data would help motivate users stick with the exercise program.

Finally, establishing ties with other seniors through exercise is extremely important to sustain interest and motivate users to continue exercising. Working out or exercising as a group enables seniors to express themselves and project their individuality. The exercise support system can thus be seen as a basic social mechanism for getting the elderly out of their homes and rooms and bringing them together for a social activity.

6 Conclusions

In this study we designed and built a prototype lower limb chair exercise support system using a depth sensor, and evaluated the functionality and effectiveness of the system through trials on a number of elderly subjects. The system provides examples of how each exercise should be done using voice instructions and model images, while also showing the muscles used superimposed onto real-time images of the exerciser performing the exercises. The system also provides exercise assessment results and advice by voice and text.

To evaluate the prototype system, we tested its recognition accuracy on elderly subjects ranging in age from late 70s to early 80s. The overall recognition accuracy for all five chair exercises was 89%, but the recognition rate was lower for two of the exercises—the heel-lift exercise and the knee extension exercise—due to relatively poor Kinect estimates of joint positions. To get feedback from actual users, we also conducted an assessment trial of the system on subjects ranging in age from 50 to 65. An overall score of 4.3 was obtained on a five-point scale, but a couple of the questions drew lower scores. On the free comments section of the questionnaire, subjects mentioned that they "would like an exercise program they could do every day" and are looking "for a way to stay physically active and healthy."

Building on the work done so far, there are several aspects of the system we hope to enhance. First, the exercise recognition accuracy of the system could be improved. We also identified a number of additional capabilities that would enhance spontaneity and encourage seniors to persist with the exercise routine: a game element that encourages users to exercise, additional exercise programs

tailored for other purposes that strengthen other muscles, and a way to visualize a user's current exercise capabilities compared with exercise data from the past. The system should also be adjustable so the exercise administrator can easily adapt the system to the fitness level of the exercisers.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to the staff of the Hitachi City Social Services Division for the Elderly, and the residents at the Kashima Retirement Home and Sukegawa Community Center.

REFERENCES

[1] Cabinet Office, 2015 Annual Report on the Aging Society, http://www8.cao.go.jp/kourei/whitepaper/w-2015/zenbun/ index.html, Accessed June 18, 2015.

[2] Ministry of Health, Labour and Welfare, Manual of Long-Term Care, http://www.mhlw.go.jp/topics/2009/ 05/tp0501-1.html, Accessed June 18, 2015.

[3] Matsukuma, H., Fujioka, S., Nakajima, A., Kaneko, K., Kajiwara, J., Hayashida, K., and Hattori, F., Research and development of serious games to support stand-up rehabilitation exercises, Journal of Information Processing, 53(3), 1041-1049, March 2012.

[4] Hashimoto, Y., Munesawa, T., Mitsuto, R., Tanabe, K., Masumoto, K., Morimoto, R., Sawada, K., and So, B. C., Internet technology in cognitive rehabilitation: review and pactice, Trans. of the Institute of Electronics, Information and Communication Engineers, J95-D(5), 1091-1099, May 2012.

[5] Cognitive Rehabilitation Anywhere, http://reha.heteml.jp , Accessed June 18, 2015.

[6] Mentiplay, B. F., Clark, R. A., Mullins, A., Bryant, A. L., Bartold, S., and Paterson, K., Reliability and validity of the Microsoft Kinect for evaluating static foot posture, Journal of Foot and Ankle Research 2013，6-14，2013.

[7] Metcalf, C. D., Robinson, R., Malpass, A. J., Bogle, T. P., Dell, T. A., Harris, C. and Demain, S. H., Markerless motion capture and measurement of hand kinematics: validation and application to home-based upper limb rehabilitation， IEEE Trans. on Biomedical Eng.，60(8), 2184-2192，Aug. 2013.

[8] Obdrzalek, S., Kurillo, G., Ofli, F., Bajcsy, R., Seto, E., Jimison, H., and Pavel, M., Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population, The 34th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society EMBC'12, 1188-1193, 2012.

[9] Erazo, O., Pino, J. A., Pino, R., Asenjo, A., Fernández, C., and Asenjo, A., Magic mirror for neurorehabilitation of people with upper limb dysfunction using Kinect, The 47th Hawaii Int. Conf. on System Sciences HICSS, 2607–2615, Jan. 2014.

[10] Webster, D. and Celik, O., Systematic review of Kinect applications in elderly care and stroke rehabilitation, Journal of NeuroEngineering and Rehabilitation, 11(108), 2014.

[11] Watanabe, T., Ohtsuka, N., Shibusawa, S., Kamada, M., and Yonekura, T., Design of lower limb chair exercise support system with depth sensor, The 11th IEEE Int. Conf. on Ubiquitous Intelligence and Computing, 104-111, Dec. 2014.

[12] Watanabe, T., Ohtsuka, N., Shibusawa, S., Kamada, M., and Yonekura, T., Motion detection and evaluation of chair exercise support system with depth image sensor, Int. Workshop on Future Trends in Computing System Technologies and Applications, 800-807, Dec. 2014.

[13] Tolentino, G. P., Battaglini, C., Pereira, A. C. V., Oliveria， R. J., and Paula, M. G. M., Usability of serious games for health, The Third Int. Conf. on Games and Virtual Worlds for Serious Applications VS-GAMES, 172-175, May 2011.

[14] Moya, S., Grau, S., Tost, D., Campeny, R., and Ruiz, M., Animation of 3D avatars for rehabilitation of the upper limbs, The Third Int. Conf. on Games and Virtual Worlds for Serious Applications VS-GAMES, 168-171, May 2011.

[15] Bruckheimer, A. D., Hounsell, M. S., and Soares, A. V., Dance2Rehab3D: A 3D virtual rehabilitation game, The 14th Symp. on Virtual and Augmented Reality, 182-190, May 2012.

[16] Asakura, R., Miyasaka, J., Kondo, K., Nakamura, Y., Akita, J., Toda, M., and Sakurazawa, S., Design of a rehabilitation supporting system integrating myoelectric sensing and image-based pose sensing, Trans. of the Institute of Electronics, Information and Communication Engineers, J97-D(1), 50-61, Jan. 2014.

[17] Pei, Y., Kim, Y., Obinata, G., and Hase, K., Design of motion trajectory and external force on foot based on musculo-skeletal model in robot-assisted lower limb rehabilitation, Trans. of the Japan Society of Mechanical Engineers, Series C, 77(781), 236-250, Sept. 2011.

[18] Fasola, J. and Mataric, M. J., Using socially assistive human–robot interaction to motivate physical exercise for older adults, Proc. of the IEEE, 100(8), 2512-2526, Aug. 2012.

[19] Igarashi, N., Suzuki, K., Kawamoto, and H., Sankai, Y., A Wearable light-emitting sensor suit for supporting the lower-limb motion perception, Journal of Information Processing, 53(4), 1360-1371, April 2012.

[20] Reflexion Health, http://reflexionhealth.com/ , Accessed July 22, 2015.

[21] Virtualrehab, http://www.virtualrehab.info/ , Accessed July 22, 2015.

[22] Seeme, http://www.virtual-realityrehabilitation.com/products/ seeme/what-is-seeme, Accessed July 22, 2015.

[23] Jintronix, http://www.jintronix.com/ , Accessed July 22, 2015.

[24] Patent WO 2013090554 A1, Method and system for evaluating a patient during a rehabilitation exercise, Publication date June 20, 2013. Also, Patent US 20140371633 A1, Publication date Dec. 18, 2014.

[25] Tao, G., Archambault, P., and Levin, M. F., Evaluation of a virtual reality rehabilitation system for upper limb hemiparesis, Int. Conf. on Virtual Rehabilitation ICVR, pp.164-165, Aug. 2013.

[26] Norouzi-Gheidari, N., Levin, M. F., Fung, J., and Archambault, P., A research protocol exploring the use of haptic forces for stroke rehabilitation, Int. Conf. on Virtual Rehabilitation ICVR, pp.220-221, Aug. 2013.

APPENDIX

A.1 Items on the Chair Exercise Support System Evaluation Questionnaire

Table A.1: Questionnaire Items.

	Questionnaire items
1	Were you able to understand the five types of chair exercises provided by the system?
2	By continuing with the exercises, to you think this would have beneficial effect on preventative care?
3	Did the muscle display function help you understand how muscles are trained and strengthened?
4	Were you aware of which muscles are used when doing the exercises?
5	Were displays of illustrated effects of exercise easy to understand?
6	Was the 4-grade exercise scale useful for measuring exercise progress?
7	Was the real-time color exercise evaluation display useful for measuring exercise progress?
8	Did you think the system was easy to use?
9	Do you think the typical user could quickly figure out how to use this system?
10	Do you think there is no need to learn a lot of preliminary stuff before using the system?
11	Would you be interested in using this system again?

TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

A Double Threshold Energy Estimation Approach to Optimize Spectrum Sensing in Cognitive Radio Network

H.Venkatesh kumar[1] and M.N.Giriprasad[2]
[1]*Dept of ECE, Nagarjuna College of Engineering and Technology, Bangalore, Karnataka, India.*
[2]*Dept of ECE, JNTU College of Engineering,Anatapur, Andhra Pradesh, India.*
[1]venkateshkumar.h@gmail.com, [2]mahendragiri1960@gmail.com,

ABSTRACT

In cognitive radio network, detection of Spectrum is a new innovation to analyses exploitation of underutilized range to overcome the issue of spectrum shortage. One of the vital Spectrum detecting strategies for cognitive radio is energy detection. In this paper, energy detection method is proposed for cooperative and non-cooperative cognitive radio. In this work we introduce new scheme for the spectrum sensing which is based on the improved double threshold method. Results demonstrate that detection probability increases whenever signal to noise ratio (SNR) and false alarm probability increases. Here, we discuss about advancement of threshold value along with energy identification for enhancing the outcome of spectrum sensing. Setting threshold value to reduce spectrum sensing fault, shrinks collision probability with primary user, enhance the value of available spectrum, hence enhancing aggregate spectrum efficiency. In any case, when deciding threshold level, spectrum sensing limitation should additionally be fulfilled since it promises least protection level of Primary User(PU) and utilization level of empty range. To minimize spectrum detecting lapse for given range detecting imperative, we determine an ideal adaptable threshold level by using the spectrum detecting lapse capacity and imperative which is given by imbalance condition. Simulation results demonstrate that the proposed plan gives better spectrum sensing results.

Keywords: Cognitive Radio; Primary user; Secondary user; Energy detection; Double threshold Algorithm.

1 Introduction

Cognitive radio is a novel methodology for enhancing the usage of one of the valuable normal resources, the radio range. The cognitive radio (CR) can be considered as another kind of software - defined radio. CR can be termed as an intelligent wireless communication [1]. In cognitive radio, PU is characterized as authorized client or licensed user who has the privilege to use a specific part of spectrum. Then again, secondary user (SU) or CR clients don't have the permit for the use of spectrum yet can use it at whatever point PU is most certainly not present. Cognitive user shifts the transmission to a distinctive frequency or modulation parameters, consequently never cause interference to PU at whatever point it is available. In this manner, SU must have the ability to sense the spectrum and check whether PU is utilizing it and at the same time make changes in the radio parameters to use the unused part of spectrum. Till now, Spectrum detecting is the most key assignment for the implementation of CRs since they have to sense hole in the spectrum band, furthermore, choose whether to utilize the range band or not. A wide range of systems were proposed for distinguishing

DOI: 10.14738/tnc.34.1406
Publication Date: 25th August 2015
URL: http://dx.doi.org/10.14738/tnc.34.1406

proof of the PU signal transmission. Range detecting strategies can be classified to: Energy Identification, Cycle stationary Identification, and Matched Filter Identification.

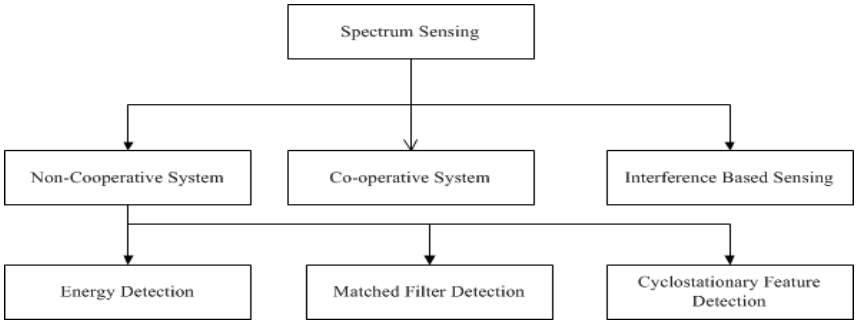


Figure 1 Classification of Energy Identification Method

Among these, energy Identification is generally used since it doesn't require the earlier information of primary signals and have less complexity contrasted with other strategies. In this paper, we have concentrated on the method of energy Identification and its uses in cognitive radio network. The fundamental function of spectrum analyzing is to identify holes in the spectrum. So the secondary user (SU) can access to the unused channel under the condition that don't cause interference to primary users (PU). In the meantime the SU handles the primary users in order to have the capacity to rapidly exit when the PU reuses the band. One of the greatest difficulties for SS is distinguishing the weak PU signaling low Signal to Noise Ratio (SNR) environment. In low SNR environment, the execution of spectrum detecting performance will be decreased [2].

In this paper the energy detection method is applied for detecting the spectrum in cognitive radio systems. Simulation is carried out and perception demonstrates to us that energy detection is upgraded when there is an increment in the SNR or increment in likelihood of false alert increments.

2 Literature Survey

As cognitive networks are gradually developed and positioned under various high-performance networking edges, various algorithms have been planned for energy recognition and spectrum identifying. We discuss below a detail survey of these efforts.

In [1] Yonghua Wang, Pin Wan et al proposed a Stochastic Resonance (SR) spectrum detecting structure for the cognitive radio. They discussed about, an Evstigneev-type monostable stochastic resonance framework which is connected to energy detection of spectrum detecting to build the framework yield SNR, consequently upgrading the low SNR environment energy detection performance. Simulation results show that on account of steady false alert probability, the detection probability of spectrum sensing taking into account monostable SR is higher than that of the customary energy detection schemes, particularly in low SNR environment.

In [2] Sobron, I.; Diniz et al presented an adaptive method for spectrum sensing and energy detection. The detecting was performed through energy detection executed by each cognitive user. The fundamental commitments of the paper were: (i) another cost-function that characterizes another test measurement in view of a energy single parameter for single-node and community situations, (ii) another type of accumulating the data from distinctive neighboring hubs that relies on upon the standardized SNRs of the hubs.

L. Rugini, P. Banelli [3] in their work introduced energy detection technique for small sample size. By using Gaussian method, they derived a new, simple, and accurate mathematical expression for the minimum number of samples required to achieve a desired probability of detection and false alarm.

J.-g. Huang, and C.-k. Tang [4], in this paper, the detection was balanced to decrease location mistake because of clamor vulnerability. The inadequate nature of vitality change is misused to revise the judgment result. Reproduction results demonstrate that under certification the benefit of the customary vitality location, the proposed differential vitality recognition can viably enhance exact detection execution of the unmoving range for the subjective clients continuously

3 System Model

Here we are using energy detection for the use of cooperative analysis of spectrum then the secondary user pass on sensing results to fusion center by two methods which is discussed below.

3.1 Data fusion

All cognitive individuals enhance the signal received by the primary users then they pass on these signals to fusion center [3], [4]. Here secondary users do not require process of complex detection, the bandwidth of reporting channel and the bandwidth of sensing channel should be nearly same. Various fusion methods are getting applied to the fusion center for an example maximal ratio combining (MRC) and square law combining (SLC)). The information of the channel state ranging from primary to secondary user and in turn passed on to fusion center is required in MRC technique .In SLC we require information of the channel state which is passed by secondary user to fusion center considering fixed amplification at every secondary user. In case, different amplification factor is considered, information of the channel state through primary to secondary users and then from secondary to the fusion center is required. Here we are proposing a framework which is having two-user or multiple-user in cooperative spectrum analysis through data fusion [5], [6]. Here we had not discussed much about analytical study on identifying capacity in spectrum analysis.

An energy detector determines the energy used for a signal to reach the receiver from the transmitter. There are two types of energy detectors. They are the analog energy detector and the digital energy detector.

In the analog energy detector, it consists of a pre-filter, squarer, and an integrator. The prefilter reduces the noise and the integrator gives the measure of the energy the signal uses to reach the receiver from the transmitter.

3.2 Decision Fusion

Each SU settles on a decision on the PU's action and the individual decisions are accounted for to the combination focus over a reporting channel .Capacity of complex sign handling is required at every SUs. The combination guideline at the combination focus can be OR, AND, or Majority principle, which can be summed up as the "k-out of-n rule" [7]. Two principle assumptions are made:

- There is no error in reporting channel; and
- The SNR insights of the got primary signs are known at SU. In [7-8], detection performance has been researched by considering reporting errors with OR combination manages under Rayleigh fading channels.

In proposed model, it is assumed that the energy detection is implemented at every SU. Energy detector is consisting of a finite time integrator and a squaring law device. Output of the integrator at any time instant is input energy to squaring device over the time interval 'T'. Noise pre-filter limits noise bandwidth, input noise to the squaring device has the spectral density as flat and band-limited

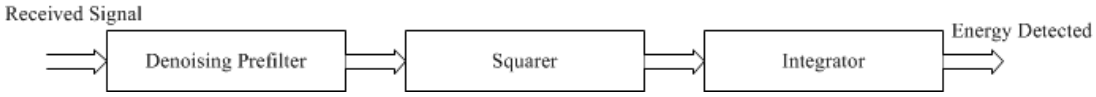


Figure 2 Analog Energy Detection Technique

Analog energy detector contains the denoising filter, squarer and integrator whereas the digital energy detector is same as the analog energy detector except that it contains an additional analog to digital converter. Figure 2 shows the architecture of Analog detector and figure 3 digital energy detector

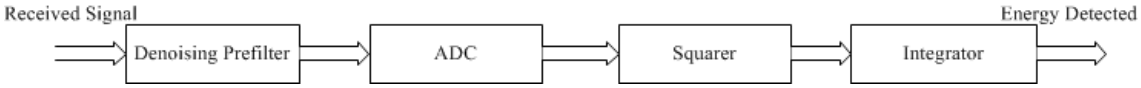


Figure 3 Digital energy detection technique

Energy detection method is one of the sub-optimal Signal detecting procedures which have been massively utilized as a part of radio interchanges. The recognition strategy can be performed in time area and additionally in frequency domain. Figure1 demonstrates the energy detection process with the theories as take after

$$H_0 = Y[m] = W[m] \quad \text{; Absence of signal} \tag{1}$$

$$H_1 : Y[m] = X[m] + W[m] \text{ ; Presence of signal} \tag{2}$$

Here, $m = 1, 2, \ldots, M$; where M is the window under surveillance Here $X[m]$ represent sample of target signal which is having definite power $'u'$ and $W[m]$ is a sample noise that is considered to be additive white Gaussian noise (AWGN) which is having 0 mean and change same to the signal power. Hypothesis 'H0' shows nonappearance of the primary user and the frequency band of interest only has noise whereas 'H1' points for existence of primary user. So for the both state hypotheses numbers of significant cases are:-

- $H1$ would be TRUE in case of existence of primary user i.e. $P(H1/H1)$ is considered as possibility of detection
- $H0$ would be TRUE in case of existence of primary user i.e. $P(H0/H1)$ is considered as possibility of misdetection
- $H1$ would be TRUE in case of existence of primary user i.e. $P(H1/H0)$ is considered as possibility of false alarm

The possibility of detection is of main worry as it provides the possibility of suitably sensing the occurrence of primary users in frequency band. Possibility of non-detection is the accompaniment of detection possibility. The aim of the analyzing schemes is to optimize the detection possibility for a low possibility of wrong alarm. There is a trade-off among these two possibilities in general. Receiver Operating Characteristics (ROC) provides much vital information with regards to the behavior of recognition possibility with varying false alarm possibility $(Pd\ v/s\ Pf)$ or non-detection possibility $(Pm\ v/s\ Pf)$.

The energy is estimated by:

$$E = \sum_{n=0}^{N} |x(n)|^2 \tag{3}$$

Now the Energy is matched to a threshold for examination which hypothesis would be true.

$$E > \lambda \implies H_1$$

$$E < \lambda \implies H_0 \tag{4}$$

The detecting is accomplished to make definite if any action of the primary user for a particular band of frequency occurs, as suggested by binary hypothesis testing, and that can be mapped as:

H_0: The idle primary user

H_1: The working primary user

Missed detection senses busy channel as an idle channel and selects the hypothesis $H0$, which causes harmful interference to the primary user whereas false alarm senses idle channel as busy channel and selects the hypothesis $H1$, which cases the secondary user to miss the opportunity for efficient spectrum utilization[9].

Based on this the performance of the detection technique can be defined by the following two possibilities. The possibility of unused detection,

$$P_{md} = P(H_0 = H_1) \tag{5}$$

And the possibility of false alarm

$$P_{fa} = P(H_1 = H_0) \tag{6}$$

Possibility of Recognition for AWGN Channel

Possibility of recognition Pd and false alarm Pf can be assessed respectively by

$$P_d = P(V > \frac{V_{th}}{H_1})$$
$$P_f = P(V > \frac{V_{th}}{H_0}) \tag{7}$$

Where Vth is the threshold

Conventional Single-Threshold Power Recognition Alogorith

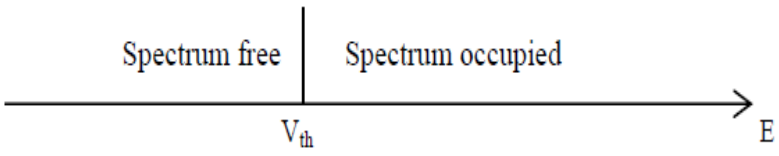


Figure 1 Single threshold method

Neyman-Pearson criterion [10] is mostly used in traditional single-threshold energy estimation algorithm. In Figure. 4, we have single detection threshold. The received signal energy defined as V is higher than the estimation threshold Vth then it is assumed the presence of primary user, represented as$H1$, on the contrary, here primary user represented as $H0$.

The estimation possibility, false alarm possibility, and miss possibility can be evaluated [11], respectively:

$$p_d = Pr\left(V > \frac{V_{th}}{H_1}\right) = Q_u(\sqrt{2\gamma}, \sqrt{V_{th1}}) \tag{8}$$

$$p_f = \Pr\left(V > \frac{V_{th}}{H_0}\right) = \frac{\Gamma(u,\frac{V_{th}}{2})}{\Gamma(u)} \quad (9)$$

$$p_m = \Pr\left(V \le \frac{V_{th}}{H_1}\right) = 1 - p_d \quad (10)$$

Here is the SNR (Signal-Noise Ratio) received by cognitive user, Vth is the detection threshold, $Qu\ (a, b)$ is normalized Marcum function with the order u . $\Gamma(a, b)$ is a non-complete gamma function; $\Gamma(a)$is complete gamma function.

4 Improved Double-Threshold Energy Detection Algorithm

We add another detection threshold within the conventional single-threshold energy detection algorithm, and it becomes a double-threshold energy detection algorithm with two detection thresholds (V_{th0} and V_{th1}).

The primary user will be detected if and only if $V > V_{th1}$, and will not be presented if and only if $V < V_{th0}$, corresponding to $H1$ and $H0$, respectively.

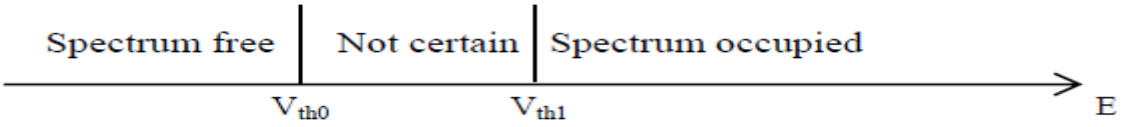


Figure 5 Double threshold methods

In this model, two thresholds V_{th0} and V_{th1} are used to help the decision of the secondary user.

- If energy value exceeds V_{th1}, then this user reports $H1$, which means that it 'sees' the primary user. If E is less than V_{th0}, decision H0 will be made.

Otherwise, if E is between V_{th0} and V_{th1}, then we also allow the secondary user reporting its observational energy value[12].

- So in our model, the fusion center receives two kinds of information: local decisions and observational values of the secondary users, i.e. local energy values.

Following are the performing schemes of the double threshold energy detection cooperative spectrum sensing method. When the detected energy V is in (V_{th0}, V_{th1}), this result is invalid because of easy to mistaken. It needs redetection.

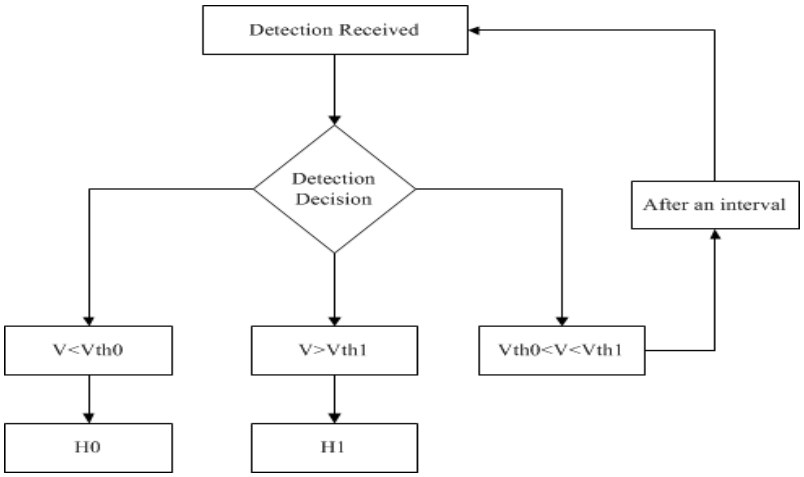


Figure 6: Double threshold energy detection

From our discussion the performance indicator for the detection probability, false alarm probability and missing probability for double threshold method can be calculated using

$$p'_d = \Pr(V' > V_{\frac{th1}{H_1}} = Q_u(\sqrt{2\gamma'}, \sqrt{V_{th1}}) \quad (11)$$

$$p'_f = \Pr\left(V' > \frac{V_{th1}}{H_0}\right) = \frac{\Gamma(u, \frac{V_{th1}}{2})}{\Gamma(u)} \quad (12)$$

$$p'_m = \Pr\left(V' \leq \frac{V_{th0}}{H_1}\right) = 1 - p'_d \quad (13)$$

Here p'_d is the correct detection probability when the primary user presents.

$p'_f$is the probability of the primary user detected presently, but in fact it does not present.

p'_m is the probability of the primary user perhaps may not be detected, but in fact it does present. According to our proposed Improved Double Threshold Energy Detector we are using two thresholds values (V_{th0}, V_{th1}) in improved energy detector. Here by adding the advantage of less probability of collision of Double threshold algorithm ,with advantage of better Detection of Improved Energy detection of spectrum sensing , we are getting the better performance in the Energy detection method of spectrum sensing .Expressions for detection probability, false alarm, collision probability and spectrum non-available probability are

$$P_c = \Pr\left(V < \frac{V_{th0}}{H_0}\right) \quad (14)$$

$$P_{na} = \Pr\left(V > \frac{V_{th0}}{H_0}\right) \quad (15)$$

$$P_f = \Pr\left(V > \frac{V_{th1}}{H_0}\right) \quad (16)$$

$$P_d = \Pr\left(V > \frac{V_{th1}}{H_1}\right) \quad (17)$$

5 Results and Discussion

Receiver operating characteristics(ROC) plot for energy detector based spectrum sensing:

Pm=probability of missed detection

Pd= probability of detection

Pf= probability of false alarm

Pc= probability of collision

Detection probability (Pd), False alarm probability (Pf) and missed detection probability (Pm) are the key measurement metrics that are used to analyze the performance of spectrum sensing techniques. The performance of a spectrum sensing technique is illustrated by the receiver operating characteristics (ROC) curve which is a plot of Pd versus Pf (or) Pd versus Pm.

The performance of energy detector is analysed using ROC curves. Monte-Carlo method is used for simulation. The plot of Probability of false alarm versus Probability of detection for different values of probability of false alarm is illustrated in Figure.7 and it can be interpreted from Figure.8 that the performance of energy detector improves with increase in SNR and increase in probability of false alarm respectively.

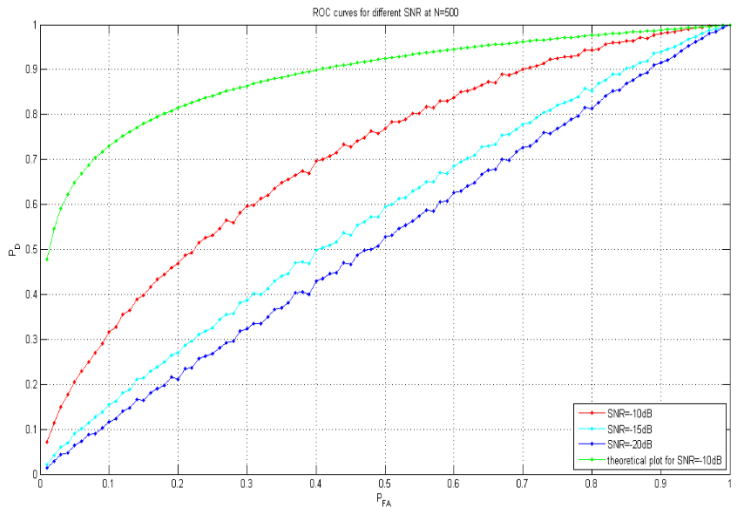


Figure 7 ROC curve for different SNR

Here we have taken probability of false alarm is (0, 1), N=500 and the SNR at three different values -10dB,-15dB,-20dB.from the Figure.7 it is observed that detection performance improved by increasing SNR value.

Table 1 PD, PF and Threshold values for SNR Variation

PD	PF	TH1
0.7107	0.001 -0.9	0.8901
0.5723	0.001 -0.9	0.8901
0.5248	0.001 -0.9	0.8901

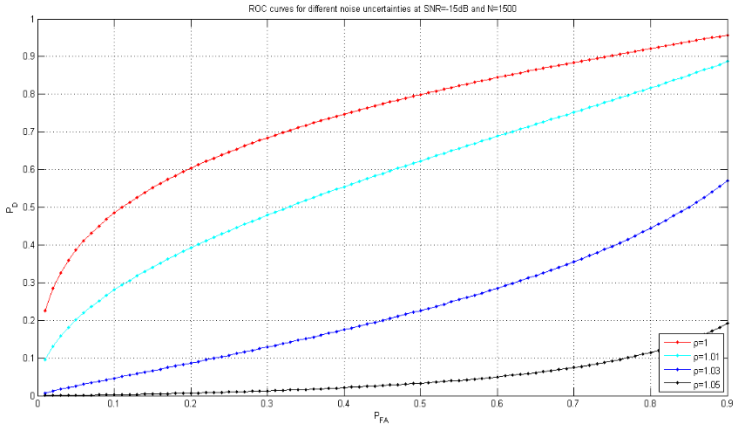


Figure 8 ROC CURVE FOR NOISE VARIATION OVER AWGN

Figure.8. above illustrates the ROC (Receiver Operating Characteristics) curves i.e. PD versus PFA using Energy detection method for spectrum sensing. This conventional method uses squaring operation. The graph is plotted for different SNR values over AWGN channel and it shows that with increase in SNR (Signal to Noise Ratio), the probability of detection increases

Table 2 PD, PF and Threshold values for SNR variation over AWGN

PD	PF	TH1
0.6829	0.001 -0.9	0.9669
0.8961	0.001 -0.9	0.9669
0.9346	0.001 -0.9	0.9669

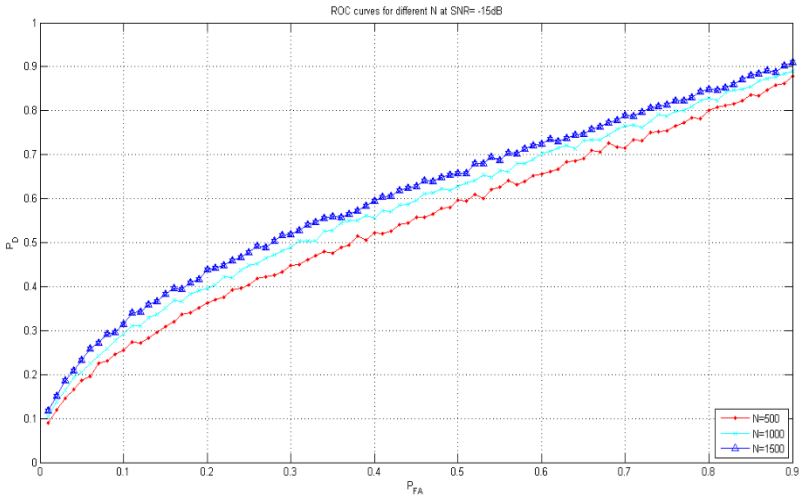


Figure 9 ROC Curve For Varied Samples

From figure 9 shown for varied number of samples, we can observe that the detection performance of Improved Double Threshold Energy detector is improved compared to Energy detector algorithm in double threshold Energy detection, as the number of samples increases the probability of detection also increases.

Table 3 PD, PF and Threshold values for sample variation

PD	PF	TH1
0.8750	0.001 -0.9	0.9556
0.8942	0.001 -0.9	0.9806
0.900	0.001 -0.9	1.0351

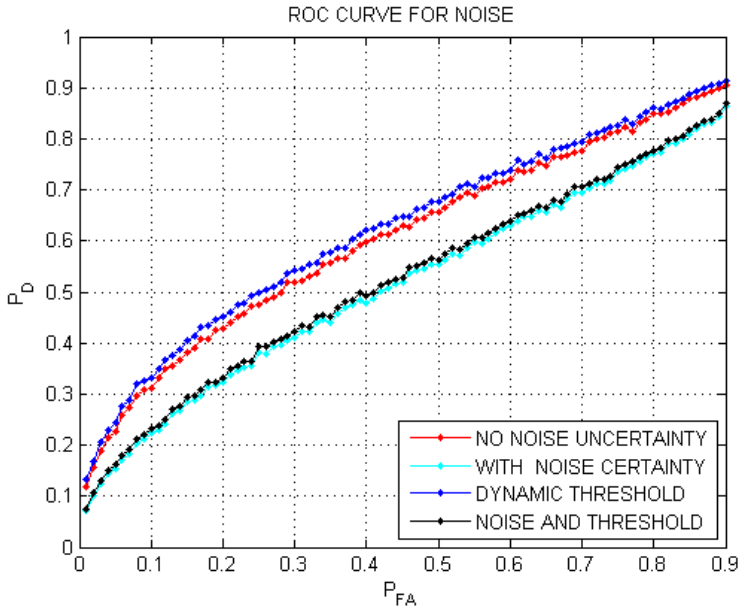


Figure 10 ROC Cure for No Noise Uncertainty, Noise Certainty, Dynamic Threshold, Noise and Threshold Variation.

From the plot shown above noise variation, we can observe that the detection performance of Improved Double Threshold Energy detector is improved compared to Energy detector algorithm, with this improvement the enhancement of this Hybrid detector gives better detection performance

Table 4 PD PF and Threshold Values For No Noise Uncertainty, Noise Certainty, Dynamic Threshold, Noise and Threshold Variation noise Certainty, and uncertainty and Threshold Variation

PD	PF	TH1
0.1155	0.001 -0.9	0.9669
0.0698	0.001 -0.9	0.9862
0.1267	0.001 -0.9	0.9650

6 Conclusion

In this work, an improved double threshold energy detection method is applied for spectrum sensing to improve the SNR and detection probability in cognitive radio. It can be seen by the simulation results that by keeping the false alarm probability as constant the energy detection performance is greater compared to other techniques. Simulation results are carried out under various parameters i.e. varied number of samples, noise and threshold. Our main aim of the work is to sense the spectrum when the SNR of PU is less, here in the simulation results for SNR = -10 dB, the probability of detection is achieved 1. In the same manner, when the number of samples are increased the PD is also increased.

REFERENCES

[1]. Chen W. The spectrum sensing and interference estimation techniques of Cognitive Radio. M.S.Thesis. Chengdu: University of Electronic Science and Technology of China. 2010.

[2]. Sobron, I.; Diniz, P.S.R.; Martins, W.A.; Velez, M., "Energy Detection Technique for Adaptive Spectrum Sensing," Communications, IEEE Transactions on , vol.63, no.3, pp.617,627, March 2015.

[3]. He D, Lin Y, He C, et al. A Novel Spectrum-Sensing Technique in Cognitive Radio Based on Stochastic Resonance. IEEE Transactions on Vehicular Technology. 2010; 59(4):1680-1688.

[4]. Yonghua Wang, Pin Wan, Qin Deng, Yuli Fu, “Spectrum Sensing Based on Monostable Stochastic Resonance in Cognitive Radio Networks” TELKOMNIKA, Vol.13, No.2, June 2015, pp. 487 ~ 493.

[5]. Lin Y, He C, Jiang L, et al. A Cyclostationary-Based Spectrum Sensing Method Using Stochastic Resonance in Cognitive Radio. 2010 IEEE International Conference on Communications Workshops (ICC). Shanghai. 2010: 1-5.

[6]. L. Rugini, P. Banelli, and G. Leus, “Small sample size performance of the energy detector,” Communications Letters, IEEE, vol. 17, no. 9, pp.1814–1817, September 2013.

[7]. Althunibat, S.; Di Renzo, M.; Granelli, F., "Optimizing the K-out-of-N rule for cooperative spectrum sensing in cognitive radio networks," Global Communications Conference (GLOBECOM), 2013 IEEE , vol., no., pp.1607,1611, 9-13 Dec. 2013.

[8]. Powell , Copps et al , “Notice of proposed rulemaking and order” Federal Communications Commission Washington, D.C. 20554.

[9]. Di H, Chen H, Lingge J, et al. Spectrum Sensing Approach Based on Optimal Stochastic Resonance Technique under Color Noise Background in Cognitive Radio Networks. 2010 IEEE International Conference on Communications Workshops (ICC). Shanghai. 2010: 1-4.

[10]. Zahabi, S.J.; Tadaion, A.A.; Aissa, S., "Neyman-Pearson Cooperative Spectrum Sensing for Cognitive Radio Networks with Fine Quantization at Local Sensors," Communications, IEEE Transactions on , vol.60, no.6, pp.1511,1522, June 2012.

[11]. L.-l. Zhang, J.-g. Huang, and C.-k. Tang, "Novel energy detection scheme in cognitive radio," in Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on, Sept 2011, pp. 1–4.

[12]. Collins, S.D.; Sirkeci-Mergen, B., "Localization ROC Analysis for Multiband Spectrum Sensing in Cognitive Radios," Military Communications Conference, MILCOM 2013 - 2013 IEEE , vol., no., pp.64,67, 18-20 Nov. 2013.

TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 4
ISSN: 2054 - 7420

Survey of Probe Set and Probe Station Selection Algorithms for Fault Detection and Localization in Computer Networks

[1]Balaji Patil and [2]Vinay Kumar Pathak
[1]UTU, Dehradun. Maharashtra Institute of Technology, Pune, India;
[2]Computer Sci. & Engg. Dept. HBTI, Kanpur, India;
balaji.patil@mitpune.edu.in; vinay@vpathak.in

ABSTRACT

Probing has evolved as a promising approach for fault diagnosis in a network management. It is based on the principle of actively sending out probes in the network to infer the health of network components. Probes are test transactions whose success or failure depends on the health of the probed network components. Probing technique for fault localization involves placement of probe stations (Probe stations are specially instrumented nodes from where probes can be sent to monitor the network) which affects the diagnosis capability of the probes sent by the probe stations. Small probe sets is desirable in order to minimize the costs imposed by probing, such as additional network load and data management requirements.

In this paper we have presented an overview of various probe set selection algorithms for network fault detection and localization. We have evaluated these algorithms on a sample network for better understanding.

Keywords- Active probing, fault detection, fault localization, probe set selection algorithms.

1 Introduction

Today, high-speed communication network play an increasingly important role and lead to a demand for a higher quality network management level. It becomes important to provide an efficient solution to monitor the network for availability and performance.

Some key challenges on fault management are as follows [13]: (1) Incomplete and inaccurate information of real network (2) Real-time fault detection and localization to achieve system's automatic repair (3) Generation of minimal traffic by management station to reduce pressure on the network (4) Existence of multiple faults in the network.

Network monitoring generates huge information that needs to be processed and diagnosed to detect/localize the failure. This information is generated by either monitoring tools [1,2,3,4,5] or by network entities themselves (in the form of alarms) [6,7,8,9]. Fault management task usually include two phases: fault detection and fault diagnosis. Fault detection is to discover if there is at least one faulty component in the system. Fault diagnosis is aim to find all the faulty components by sending additional probes to the region of interests. Fault Detection is carried out by two ways i.e. Active Monitoring and Passive Monitoring.

DOI: 10.14738/tnc.34.1023
Publication Date: 30th August 2015
URL: http://dx.doi.org/10.14738/tnc.34.1023

1.1 Active Monitoring

Active monitoring deploys probing methods to gather health status and performance statistics of network entities in the managed system [1].

Probing based techniques have various advantages over passive monitoring techniques, such as (1) Less instrumentation (2) Capability to compute end-to-end performance (3) Quicker localization, etc. Developing probing based monitoring solution involves solving two major problems, namely probe station placement and probe set selection.

Different criteria's are imposed on probe set selection for fault detection and fault localization [4]. Probe set for fault detection is selected such that all elements in the managed network are probed. On the other hand, fault localization requires minimal probe set that can uniquely diagnose the suspected network element failure. Probes for failure detection are sent periodically and thus the management traffic produced should be low enough that it does not affect the performance of other applications. Fault localization is done only when some problem is encountered. Thus probes for fault localization should be selected such that the fault localization can be done in minimum amount of time and at the same time the network in the identified problem areas should not be overwhelmed with the management traffic.

1.2 Preplanned Probing

Preplanned probing involves offline selection of probes those are periodically sent out in the network [2]. The results are then analyzed to infer the network state. This approach requires probe set selection such that every failure in the network can be uniquely localized. It is practically difficult envisaging all possible failures that might occur and come up with probe sets to detect those failures. Also, sending this large number of probes at a periodic interval generates large amount of management traffic. The disadvantage of this approach is that because probes are sent at periodically at scheduled intervals, there might be considerable delay in obtaining information when problem occurs. As it is desirable to detect and localize failures quickly as they occur, this delay might not be acceptable. Moreover, this delay will potentially delay in next step of fault localization.

1.3 Adaptive Probing

Active probing initially selects probes for fault detection [2]. The probe stations send these probes and observe the network. Additional probes are sent out to obtain further information about the problem, and this process may repeat - as more data is to be obtained, which probes to send next decision is important, until finally the problem is completely determined. It greatly reduces management traffic and provides more accurate and timely diagnosis.

The outline of the paper is as follows. Section II provides the basic framework and notation. Section III covers different probe set selection algorithms for fault detection and localization. Section IV covers different probe station selection algorithms for fault detection and localization. Section V gives tabular comparison of algorithms. Section VI refers to proposed work and section VII covers the conclusion.

2 Notations

This section focuses on understanding basic concepts involved in fault management.

2.1 Faults

Event, defined as an exceptional condition occurring in the operation of hardware or software of a managed network, is a central concept pertaining to fault diagnosis. Faults (also referred to as

problems or root causes) constitute a class of network events that can cause other events but are not themselves caused by other events. In a network fault may be a particular node or link (or both) is down.

Assuming there are finite set O of objects in a network. The fault can be any subset of f $\subseteq$ O. A single element in *f* represents a network failure and similarly an empty set represents no failures in the network.

2.2 Dependency Matrix

A dependency matrix captures the relationships between system states and probes [10].

Consider a sample network as shown in Figure 1 where N_2 and N_9 are probe stations.

Given any set of nodes $N = \{N_1, N_2,..., N_n\}$ and probes $P = \{p_1, p_2,...,p_r\}$, the dependency matrix $D_{P,N}$ is given by:

$$D_{P,N}(i; j) = 1 \text{ if } N_j \cap N(p_i) \neq \phi$$

$$= 0 \text{ otherwise:}$$

$D_{P,N}$ is an *r*-by-*n* matrix, where each row represents a probe and each column represents a node.

Table 1 is a dependency matrix for a sample network shown in Figure 1. The sample network has four available probes as shown in Figure 1.

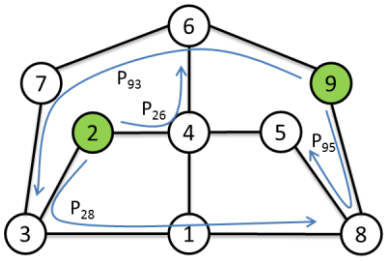


Figure 1: Sample network showing few available probes

Table 1: Dependency matrix for Sample network shown in Figure 1.

Nodes→ Probes↓	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
p_{21}	1	1	1	0	0	0	0	0	0
p_{23}	0	1	1	0	0	0	0	0	0
P_{24}	0	1	0	1	0	0	0	0	0
P_{25}	0	1	0	1	1	0	0	0	0
P_{26}	0	1	0	1	0	1	0	0	0
P_{27}	0	1	1	0	0	0	1	0	0
P_{28}	1	1	1	0	0	0	0	1	0
P_{29}	0	1	0	1	0	1	0	0	1
P_{91}	1	0	0	0	0	0	0	1	1
P_{92}	0	1	0	1	0	1	0	0	1
P_{93}	0	0	1	0	0	1	1	0	1
P_{94}	0	0	0	1	0	1	0	0	1
P_{95}	0	0	0	0	1	0	0	1	1
P_{96}	0	0	0	0	0	1	0	0	1
P_{97}	0	0	0	0	0	1	1	0	1
P_{98}	0	0	0	0	0	0	0	1	1

2.3 Fault Detection

Fault detection is to discover if there is at least one faulty component in the system. Which means the deployment of measurements must cover all nodes in the target network. Once any network anomalies are detected then the fault diagnosis process gets started. Fault diagnosis aims to find all the faulty components by sending additional probes to the region of failure.

Detection: Given $D_{P,F}$, find P^* that minimizes $|P'|$, where $P' \subseteq P$ such that there is at least one '1' in every column of $D_{P',F}$ [10].

By monitoring the probes we will be able to know about fault in the network as soon as the probe gets failed; but fault cannot be exactly localized with this much information.

2.4 Fault Localization

Fault localization requires finding the smallest probe set such that every fault has a unique probe signal, since in that case exactly which fault has occurred can be determined from the probe results [10]. Since the probe signal of fault f_j is the column c_j of $D_{P,F}$, each fault has a unique probe signal if and only if each column in $D_{P,F}$ is **unique**; i.e. differs from every other column. Since two columns c_i, c_j differ if and only if there is some entry where one of them has the value '1' while the other has the value '0' (i.e. there is some probe which is affected by one of the faults but not the other), fault localization can be expressed using the number of non-zero elements, denoted by n_{ij}, in $c_i \oplus c_j$, where $\oplus$ denotes exclusive-OR.

Localization: Given $D_{P,F}$, find P^* which minimizes $|P'|$, where $P' \subseteq P$ satisfies $\forall f_i , f_j \in F, n_{ij} \geq 1$ [10].

Referring to sample network in Figure 1 and Table 1, fault detection requires finding the smallest number of rows such that every column has at least one '1'. In above sample network example, this means the smallest set of probes which pass through every node, so that, no matter which node fails, there is a probe that will detect it. The following set of 3 probes suffices:

Nodes→ / Probes↓	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
P_{25}	0	1	0	1	1	0	0	0	0
P_{28}	1	1	1	0	0	0	0	1	0
P_{93}	0	0	1	0	0	1	1	0	1

Since no single probe passes through all the nodes, this is clearly a smallest subset for fault detection. However this set fails for the task of fault localization because, for example, failures in nodes N_4 and N_5 cannot be distinguished from each other and failures in nodes N_6 and N_7 cannot be distinguished from each other - they generate the same signal, since their columns are identical. However the following set of 4 probes is a minimal set for fault localization. Minimal set is the least number of probes which uniquely localize the fault in the network.

Nodes→ / Probes↓	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
P_{28}	1	1	1	0	0	0	0	1	0
P_{26}	0	1	0	1	0	1	0	0	0
P_{93}	0	0	1	0	0	1	1	0	1
P_{95}	0	0	0	0	1	0	0	1	1

Since all 9 columns are unique, the results of these 4 probes allow us to determine exactly which node has failed. For example, if p_{26} and p_{93} both fail, then we infer that node N_6 has failed.

3 Survey of Fault Detection and Localization Algorithms

In this section, we discuss various algorithms for fault detection and localization using preplanned and active probing. We have considered the sample network in Figure 1 for elaboration of different algorithms of probe set reduction.

3.1 Greedy Search Algorithm

3.1.1 Approach

There are various techniques researchers have suggested that are based on greedy algorithm [10]. The simplest one tries to identify a probe that covers maximum number of nodes.

3.1.2 Implementation

Consider sample network in Figure 1, node N2 and N9 are probe stations. Below are the probes that originate from these two probe stations.

Table 2: Greedy Search - nodes covered through available probes.

Available Probes	Nodes it Cover	Available Probes	Nodes it Cover
P_{21}	$\{N_1,N_2,N_3\}$	P_{91}	$\{N_1,N_8,N_9\}$
P_{23}	$\{N_2,N_3\}$	P_{92}	$\{N_3,N_4,N_9\}$
P_{24}	$\{N_2,N_4\}$	P_{93}	$\{N_3,N_6,N_7,N_9\}$
P_{25}	$\{N_2,N_4,N_5\}$	P_{94}	$\{N_4,N_6,N_9\}$
P_{26}	$\{N_2,N_4,N_6\}$	P_{95}	$\{N_5,N_8,N_9\}$
P_{27}	$\{N_2,N_3,N_7\}$	P_{96}	$\{N_6,N_9\}$
P_{28}	$\{N_1,N_2,N_3,N_8\}$	P_{97}	$\{N_6,N_7,N_9\}$
P_{29}	$\{N_2,N_4,N_6,N_9\}$	P_{98}	$\{N_8,N_9\}$

Above information can also be reframed such that each node is represented by a set of probes passing through it

Table 3: Greedy Search - probes covering each node in the network

Node	Probes Covering this node	Node	Probes Covering this node
N_1	$\{P_{21},P_{28},P_{91}\}$	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$
N_3	$\{P_{23},P_{27},P_{28},P_{93}\}$	N_7	$\{P_{27},P_{93},P_{97}\}$
N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$
N_5	$\{P_{25},P_{95}\}$		

Since N_2 and N_9 are probe stations, we can define Non-probed nodes as

NPN= { $N_1,N_3,N_4,N_5,N_6,N_7,N_8$}

As per greedy search algorithm, first step is to identify a node with minimum cardinality, node (N_5) which is covered by least number of probes (P_{25}, P_{95}). Out of these probes, select a probe that has got maximum cardinality (P_{25}) – one which can probe maximum number of other nodes. After each step the algorithm updates Node list and Probe set.

Table 4: Greedy Search algorithm - iteration wise progress on sample network

Step	Minimum Cardinality Node	Probes covering this node	Selected probe with max cardinality	NPN
I	N_5	$\{P_{25},P_{95}\}$	P_{25}	$\{N_1,N_3,N_6,N_7,N_8\}$
II	N_1	$\{P_{21},P_{28},P_{91}\}$	P_{28}	$\{N_6,N_7\}$
III	N_7	P_{93}	P_{93}	$\{\}$

Thus probe set {P_{25}, P_{28}, P_{93}} can detect any failure in the sample network shown in Figure 1.

3.2 Additive Search Algorithm

3.2.1 Approach

In this form of greedy search each probe is evaluated in terms of their localization quality [5]. Localization quality of a set of probes is defined as amount of information provided by a probe set for faults in a network.

3.2.2 Implementation

The localization decomposition $S_{P,F}$ is a collection of groups $\{G_1,...,G_k\}$, where each group G_i contains the faults $f_i \in F$, that cannot be distinguished from one another by P. Then localization quality of P is defined as the conditional entropy $H(F/G)$, where F is random variable denoting fault and G the random variable denoting which group of $S_{P,F}$ contains the fault.

$$Q(P,F) = H(F/G)$$

If the faults are independent and equally likely, then

$$Q(P,F) = \sum_{i=1}^{k} \frac{ni}{n} \; logni$$

Where n_i is the number of faults in group G_i of $S_{P,F}$ and $n=|F|$.

Input: Dependency matrix $D_{P,F}$, with rows $p_1,p_2,...,p_r$

Output: Probe set P' (possibly non-minimal size)

Algorithm:

$P' = \phi$ = empty set

While $S_{P',F} \neq S_{P,F}$

$$p^* = argmin_{p \in p \setminus p'} Q\left(p' \cup \{p\}, F\right)$$

$$P' \leftarrow P' \cup \{p*\}$$

Output P'

As an example, consider the dependency matrix shown in Table 1 corresponding to sample network in Figure 1.

Every iteration of this algorithm will select a probe with minimum $Q(P,F)$ and calculate decomposing induced by this probe. This is continued until selected probe finally results into decomposition into singleton set.

Following Table 5 shows minimum probe set, its corresponding $Q(P,F)$ and decomposition induced by each probe f_n denotes failure in Node N_n.

Table 5: $Q(P,F)$ value and decomposition induced by each probe

Probe	$Q(P,F)$	Decomposition
P_{28}	2.17	$\{f_1, f_2, f_3, f_8\},\{f_4,f_5, f_6, f_7,f_9\}$
P_{26}	1.27	$\{f_1, f_3, f_8\},\{f_2\},\{f_4, f_6\}\{f_5, f_7,f_9\}$
P_{93}	0.44	$\{f_1, f_8\},\{f_2\},\{f_3\},\{f_4\},\{f_5\},\{f_6\},\{f_7,f_9\}$
P_{95}	0	$\{f_1\},\{f_2\},\{f_3\},\{f_4\},\{f_5\},\{f_6\},\{f_7\},\{f_8\},\{f_9\}$

Thus probe set $\{P_{28}, P_{26}, P_{93}, P_{95}\}$ can localize any fault in the sample network shown in Figure 1.

3.3 Subtractive Search Algorithm

3.3.1 Approach

Subtractive search begins with the assumption that all probes are available for fault detection [5]. Gradually probes are dropped from this available set such that the localization quality is always maintained - same as obtained from the original set.

3.3.2 Implementation

Consider sample network as in Figure 1. As per Subtractive Search algorithm, initially all probes are considered to be Selected Probes (SP).

Probes are removed from SP until there is at least one probe available to discover any particular node i.e. $N_i \neq \{\Phi\}$

The probes are being considered for removal in every iteration still the removal of that probe doesn't affect the localization quality; it will get removed from the SP.

Table 6: Subtractive search algorithm - iteration wise progress on sample network

Probe	Localization quality		Probe	Localization quality	
P_{21}	Node	Probes it Cover	**P_{91}**	**Node**	**Probes it Cover**
	N_1	$\{P_{28},P_{91}\}$		**N_1**	**$\{P_{91}\}$**
	N_2	$\{P_{23},P_{24},P_{25},P_{26},P_{27},P_{28},P_{29}\}$		**N_2**	**$\{P_{29}\}$**
	N_3	$\{P_{23},P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29},P_{94}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{23}	Node	Probes it Cover	P_{93}	Node	Probes it Cover
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{24},P_{25},P_{26},P_{27},P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{24},P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29},P_{94}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{24}	Node	Probes it Cover	P_{94}	Node	Probes it Cover
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{25},P_{26},P_{27},P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},,P_{92},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{25},P_{26},P_{29},P_{92},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{25},P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{26},P_{29},P_{92},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{96},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{25}	Node	Probes it Cover	**P_{95}**	**Node**	**Probes it Cover**
	N_1	$\{P_{28},P_{91}\}$		**N_1**	**$\{P_{91}\}$**
	N_2	$\{P_{26},P_{27},P_{28},P_{29}\}$		**N_2**	**$\{P_{29}\}$**
	N_3	$\{P_{27},P_{28},P_{93}\}$		**N_3**	**$\{P_{93}\}$**
	N_4	$\{P_{26},P_{29},P_{94}\}$		**N_4**	**$\{P_{29}\}$**
	N_5	$\{P_{95}\}$		**N_5**	**$\{P_{95}\}$**

	N_6	$\{P_{26},P_{29},P_{93},P_{94},P_{96},P_{97}\}$		**N_6**	**$\{P_{29},P_{93},P_{96},P_{97}\}$**
	N_7	$\{P_{27},P_{93},P_{97}\}$		**N_7**	**$\{P_{93},P_{97}\}$**
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		**N_8**	**$\{P_{91},P_{95},P_{98}\}$**
P_{26}	Node	Probes it Cover	P_{96}	Node	Probes it Cover
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{27},P_{28},P_{29}\}$			$\{P_{29}\}$
	N_3	$\{P_{27},P_{28},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93},P_{97}\}$
	N_7	$\{P_{27},P_{93},P_{97}\}$		N_7	$\{P_{93},P_{97}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{27}	Node	Probes it Cover	P_{97}	Node	Probes it Cover
	N_1	$\{P_{28},P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{28},P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{28},P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93}\}$
	N_7	$\{P_{93},P_{97}\}$		N_7	$\{P_{93}\}$
	N_8	$\{P_{28},P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95},P_{98}\}$
P_{28}	Node	Probes it Cover	P_{98}	Node	Probes it Cover
	N_1	$\{P_{91}\}$		N_1	$\{P_{91}\}$
	N_2	$\{P_{29}\}$		N_2	$\{P_{29}\}$
	N_3	$\{P_{93}\}$		N_3	$\{P_{93}\}$
	N_4	$\{P_{29},P_{94}\}$		N_4	$\{P_{29}\}$
	N_5	$\{P_{95}\}$		N_5	$\{P_{95}\}$
	N_6	$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$		N_6	$\{P_{29},P_{93}\}$
	N_7	$\{P_{93},P_{97}\}$		N_7	$\{P_{93}\}$
	N_8	$\{P_{91},P_{95},P_{98}\}$		N_8	$\{P_{91},P_{95}\}$
P_{29}	**Node**	**Probes it Cover**			
	N_1	**$\{P_{91}\}$**			
	N_2	**$\{P_{29}\}$**			
	N_3	**$\{P_{93}\}$**			
	N_4	**$\{P_{29},P_{94}\}$**			
	N_5	**$\{P_{95}\}$**			
	N_6	**$\{P_{29},P_{93},P_{94},P_{96},P_{97}\}$**			
	N_7	**$\{P_{93},P_{97}\}$**			
	N_8	**$\{P_{91},P_{95},P_{98}\}$**			

Finally, four probes that are covering all nodes in the network are selected probes

$$SP = \{P_{29}, P_{91}, P_{93}, P_{95}\}$$

3.4 Constraint Satisfaction Problem (CSP) Algorithm

3.4.1 Approach

In constraint satisfaction problem approach authors have defined some constraints to find out the healthy and suspected set [13]. These constraints map the nodes to healthy set and suspected set.

3.4.2 Implementation

Consider the dependency matrix D, obtained for the sample network in Figure 1 is represented by Table 1.

Assuming a single node failure may occur in the managed network, let Rpi refer to response of the probe *pi* where *i = 1 … K*. *K* is the number of probes in the fault detection probe set. Rpi may take value only from the set *{0, 1}.* If the probe *pi* fails, then Rpi = 0; otherwise Rpi = *1*.

If a single probe fails (Rpi = 0) and other probes succeed (Rpi =*1*), then the initial suspect and healthy sets are constructed as follows-

$$F = \{n \mid n \in (pi - (pi \cap pj))\}$$

$$G = \{n \mid n \in ((N - pi) \cup (pi \cap pj))\}$$

Where *i, j =1...k, i ≠ j.*

Using greedy search scheme, first three probes are selected that can completely cover all nodes in the managed network. Should one or both of these probes report the occurrence of a failure, the fault localization function will be invoked promptly. The information carried by the detection probes will be fully exploited as follows:

Nodes→ / Probes↓	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
P_{25}	0	1	0	1	1	0	0	0	0
P_{28}	1	1	1	0	0	0	0	1	0
P_{93}	0	0	1	0	0	1	1	0	1

The following set of 3 probes suffices available probes and nodes it covers:

Available Probes	Nodes it Cover
P_{25}	{N_2,N_4,N_5}
P_{28}	{N_1,N_2,N_3,N_8}
P_{93}	{N_3,N_6,N_7,N9}

- If P_{25} has failed and P_{28} and P_{93} has succeeded

Suspect set F = P_{25} – ($P_{25} \cap (P_{28}+P_{93})$) = {$N_4$, N_5}; Healthy set G = P_{45} = {N_1, N_2, N_3, N_6, N_7, N_8}

- If P_{25} and P_{28} had failed and P_{93} has succeeded

Suspect set F = $P_{25} \cap P_{28} - P_{93}$ = {N_2}; Healthy set G = {N_1, N3, N4, N5, N6, N7, N8, N9}

3.5 Binary Search Algorithm

3.5.1 Approach

As name suggests, binary search algorithm uses binary search to localize faulty nodes in the network [4]. If any probe used for fault detection fails, then all nodes in the path of failed probe are treated as suspected nodes. In order to identify the failed node from these suspected nodes, additional probes are sent recursively until the search narrows down to failed node.

3.5.2 Implementation

Assuming optimal probe set {P_{25}, P_{28}, P_{93}} used for fault detection is derived using greedy algorithm mentioned in section 3.1 for the sample network shown in Figure 1 and its dependency matrix in Table 1.

Say P_{28} has failed making Suspected Node set

SN= {N_3, N_1, N_8) - excluding probe station N_2

Starting from probe station, each node on the failure path are numbered sequentially,

N_2 (1)→ N_3 (2)→ N_1 (3) → N_8(4)

And binary search is applied to identify mid (Target) node in the path such that additional probe can be sent to this node

*TargetNode = (StartNodePosition+EndNodePosition)/*2

TargetNode = (1 + 4)/2 = round (2.5) = 3

Hence, additional probe P_{21} is selected to check if node identified by position 3 (N_1) is faulty.

If P_{21} passes, then clearly failure is with node N_8 and if P_{21} fails, search is repeated with EndNodePosition set to TargetNode.

Next *TargetNode* = (1+3)/2 = 2

Hence, additional probe P_{23} is selected to check if node identified by position 2 (N_3) is faulty.

If P_{23} passed, then clearly failure is at node N_1 else failure is at node N_3. This situation is represented by the Figure 2.

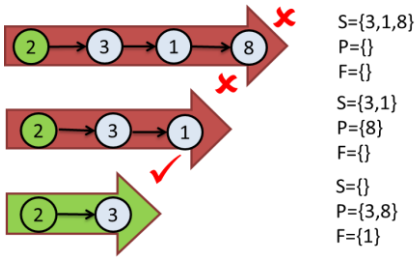


Figure 2: Probes sent in a binary search fashion on a failed probe path 2→8 to identify failed node on the path

3.6 Max Search Algorithm

3.6.1 Approach

In Max search algorithm implementation, probes are iteratively selected from available probe set such that selected probes cover maximum number of uncovered nodes till all the suspected nodes are covered [4]. This is represented in Figure 3.

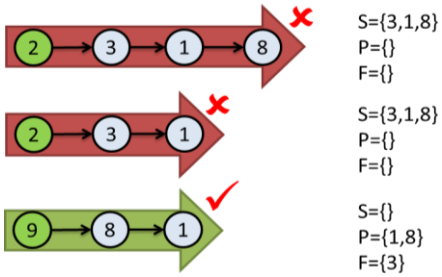


Figure 3: Probes sent in a Max search fashion on a failed probe path 2→8 to identify failed node on the path

3.6.2 Implementation

Again assuming probe P_{28} has failed resulting into Suspected Node set as

SN ={ N_3,N_1,N_8 } - excluding probe station N_2

From dependency matrix shown in Table 1, probes that cover maximum nodes from SN are P_{21} = {N_2, N_3, N_1} and P_{91} = {N_9, N_8, N_1}

These additional probes are sent to localize the failure node on the path.

Thus, if P_{21} pass and P_{91} fails, Faulty Node (F)

$F = P_{91} – (P_{91} \cap P_{21}) = \{N_8\}$ – excluding probe station N_9

If P_{21} fails and P_{91} pass, Faulty node

$F=P_{21} – (P_{21} \cap P_{91}) = \{N_3\}$

If both P_{21} and P_{91} fails,

$F=P_{21} \cap P_{91} = \{N_1\}$ is faulty

3.7 Min Search Algorithm

3.7.1 Approach

The Min search algorithm works on the concept of selecting a probe for each Suspected Node (SN) set such that the selected probe goes through minimum number of other nodes in the suspected node set [4].

3.7.2 Implementation

Again assuming probe P_{28} has failed resulting into suspected node set as

SN = {N_3, N_1, N_8} - excluding probe station N_2

From dependency matrix Table 1, probes that cover minimum nodes from SN are P_{12}= {N_1, $N2$}, P_{13}={N_1,N_3} and P_{43}={N_3,N_4 }

These additional probes are sent to localize the failure node on the path.

Thus, if any of the above probes fails, it can uniquely identify failure in nodes N_2 and N_3. Whereas success of all three probes will uniquely identify failure in node N_5. This situation is shown in Figure 4.

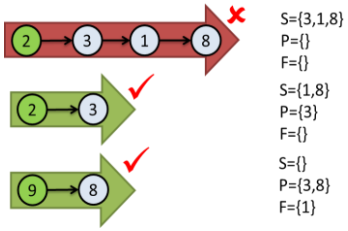


Figure 4: Probes sent in a Min search fashion on a failed probe path 2→8 to identify failed node on the path

4 Survey of Probe Station Selection Algorithms

In this section we have presented some of the recent approaches proposed for probe station selection/placement. Location and responsibilities assigned to probe stations must be decided while building an active probing solution. These decisions are based on nature of routes, nature of targeted

failures, availability of dependency information etc. [11]. Below we discuss various such factors that contribute to the overall decision making of probe station selection:

- **Nature of targeted failures**: Probe station selection depends on the nature of faults to be diagnosed viz. a node failure or an edge failure. A single probe station might not be sufficient to detect all of node and edge failures. For instance, consider the network shown in Figure 5. Consider node 1 to be a probe station; it can detect any single node failure in this network. However, it can detect failure of only those links that are used in reaching other nodes in the network, i.e., the links shown in red.

- **Maximum numbers of failures:** In a connected network consisting of k failures, a set of probe stations can localize any k non-probe-station node failures if and only if there exists k independent probe paths to each non-probe-station nodes. Figure 6 shows 3 independent (node disjoint) paths to node 5 from probe station 1. Even if there are failures in two paths, node 5 can still be probed.

- **Probe station failure:** The assumption of fault tolerant probe station may not be practical and hence probe station selection problem becomes even more challenging. In case of probe station failure, probe stations are selected such that there exists k independent paths to each of the probe station's as well.

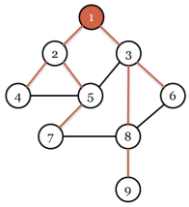


Figure 5: Link failures not being covered by Probe station 1[3]

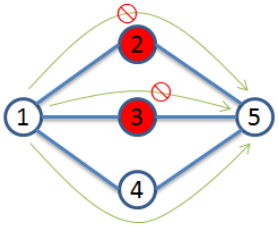


Figure 6: k Independent paths allow detection of k node failures

- **Topological constraints:** Another important criterion involved in probe station selection is the topological constraint. The node with less connectivity needs special treatment. Special topology structures like chains and rings also demand specific probe station placement requirements. One approach to simplify this problem could be to devise a solution by reducing the network into smaller sub-networks connected by such specific network structures like rings, chains, leaves, etc.

4.1 Shadow Node Reduction Algorithm (SNR)

4.1.1 Approach

In SNR a node is selected as probe station having maximum connectivity and minimum shadow node set.

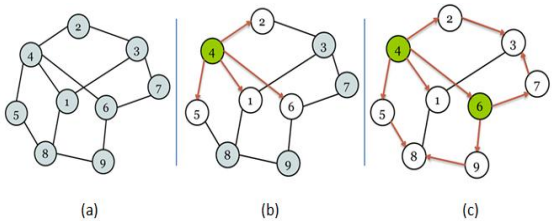


Figure 7: Probe station selection to detect any two node failures

4.1.2 Implementation

Figure 7 shows an example of how the probe station selection algorithm selects probe stations to detect any two node failures in the network. Figure 7(a) shows a network topology with nine nodes

considering all nodes as uncovered nodes. Figure 7(b) shows node 4, being the node with largest degree, as the selected probe station removing neighboring nodes 2, 6, 1, and 5 from the uncovered node set. Figure 7(c) shows node 6 as the next selected probe station, which removes neighboring nodes 9 and 7 from the uncovered node set. Nodes 3 and 8 are not neighbors of any probe station, but they have two independent probe paths from probe station 4 and 6 as shown in the Figure 7(c). Thus both nodes 3 and 8 are also removed from the uncovered node set. Thus the probe station placement at nodes 4 and 6 can detect any two node failures in the network.

4.2 Min. Hitting Set Reduction Algorithm

4.2.1 Approach

This approach uses algorithm for probe station selection using a reduction of the probe station selection problem to the Minimum Hitting Set problem. This approach reduces no. of probe stations to an optimum value [14].

4.2.2 Implementation

Referring to Figure 8: Consider a network of five nodes (1, 2, 3, 4, 5). Form all possible pairs of size two viz. {1, 2}, {1, 3} …. {4, 5} (Section - C). Now we draw a mapping between a 'Node' and a pair, Section C', if the pair can provide two independent paths to that particular node. For e.g. if node 1 gets two independent paths from pairs {1, 2} and {1, 3}, draw the mappings between node 1 and the two pairs. Identify and draw such mappings for each node in the network. Rename each pair using some naming convention, viz. rename {1, 2} as 'a'(Section –S). Now for each node in the network, write the pairs that can give two independent paths for that node e.g. for node 5, the set will be {a, d, i, j}. Each such set makes for an element in S.

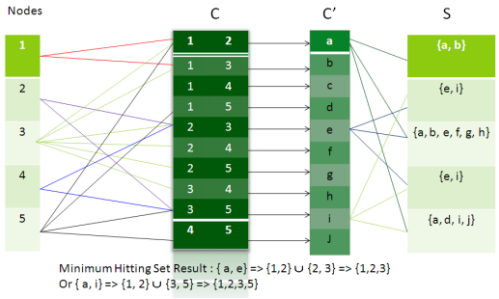


Figure 8: Minimum hitting set reduction algorithm.

Now if you look at the right half of the Figure 8, you see the familiar problem of k-sized Minimum Hitting Set Problem, which can be solved by using any approximation algorithms for the MHSP. It also addresses issues involved while selecting probe stations such as link failures and probe station failures.

5 Tabular Comparison of Algorithms

In this section we have summarized most of the recent conventional algorithms for Probe set selection and Probe station selection algorithms. These algorithms play a vital role in development of optimal solution for fault detection and localization in networks. The tabular representation in Table 7 is focused on algorithms used, limitations and future directions for the research in the field of fault localization.

Table 7: Comparative study of various probe set selection and probe station selection algorithms

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantages	Limitations
1	**Greedy Algorithm**	Fault detection	Quadratic $O(r^2)$ r- no of nodes in network	Identify a node that covers maximum number of nodes. i.e. Maximum co-cordiality	Maximize the information gain Probes set generated- (P25, P28, P93)		May not necessarily result in minimal set but as compared to Additive search gives better results.
2	**Additive Search**	Fault Localization	Quadratic O(r) r- no of nodes in network	Localization quality of a probe set is defined by the probe set provided for fault in network	-Select a probe with minimum QPF Probes set generated- (P28, P26, P93, P95)		May not necessarily result in minimal set.
3	**Exhaustive Search**	Fault detection	Exponential K^n worst case complexity K- no of distinct probes passing through a nodes	A node can be diagnosed if there is at least one probe passing through it. Probes can be added incrementally In all feasible combinations until the minimum set is reached.			Impractical for large networks.
4	**Subtractive Search**	Fault detection	Linear O(N) N- no of nodes in network	Initially all probes are available for fault detection gradually probes are dropped from the available set such that the localization quality is always maintained.	Select probes keeping localization quality always maintained. Probes set generated- (P29, P91, P93, P95)		May not necessarily result in minimal set. Effectiveness in finding minimal set depends on the order in which probes are explored
5	**Max Search**	Fault detection & localization	$O(N^2)$ N- no of nodes in network	Probes are iteratively selected from available probe set such that selected probe covers maximum number of uncovered nodes till all the suspected nodes are covered.	Selects probes that covers maximum no of suspected nodes.		- As compared to min and Binary search, it takes more time (round trips) to localize a fault.
6	**Min Search**	Fault localization	$O(N^2)$ N- no of nodes in network	Selecting a probe for each suspected node set such that the selected probe goes through minimum no. of other nodes in the suspected node set.		Min search diagnoses all suspected nodes in parallel. Smaller probe size	-
7	**Binary Search**	Fault localization	$O(N^2)$ N- no of nodes in network	Probes are sent in binary search fashion. On a failed probe path, probe is first sent from the probe station half way on the probe path. If this probe fails, further diagnosis is done on the first half of the probe path. On the other hand, if the probe succeeds, then the later half of the probe path is diagnosed in similar fashion.	Probes are sent in binary search fashion.		No of probes and iterations required to localize failure is more as compared to min search.
8	**CSP-based model**	Fault detection & localization	-	This approach is based on Constraint Satisfaction Problem (CSP) techniques to find an appropriate and optimal collection of available probes for the fault identification.	To find an appropriate & optimal collection of available probes for the purpose of fault identification Probes set generated- (P25, P28, P93)	It performs better than Greedy and subtractive	Proper Constraints should set to get optimal solution Very few people had applied.
9	**Active probing**	Fault localization	$< O(N^2)$	In active probing selection of later probes depends on the results of earlier probes. The results of the probes are analyzed to infer what problems might be occurring identifying those and useful probes are to be send next are determined and send.	In active probing the selection of later probes depends on the results of earlier probes	Fewer probes can be used than the pre-planned probing.	Requires a more complicated technology to determine which probes to send

Probe Station Selection Algorithms

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantage	Limitations
1	SNR Algorithm	Probe station selection	$O(N^{MAXPSSETSIZE})$ N-no of node MAXPSSETSIZE-max no of probe stations	In the first iteration probe stations are added such that each non probe station node that is either a neighbor of a probe station or has two independent probe paths. The nodes are added in this fashion by incrementally increasing the overall diagnostic capability, till *k* faults in the network can be localized.	Select a highest node degree as first probe station.	Computationally feasible to implement.	The algorithms in this paper ensure the availability of k independent probe paths but do not aim to optimize probe traffic or the localization time.

SN	Algorithm	Scope	Complexity	Mechanism	Seed Idea	Advantages	Limitations
2	Greedy Approximation Algorithm	Probe station selection	O (nk) -n no of nodes -k no of failures	For Minimum Hitting Set problem Greedy approximation algorithm is used. Inapproximability results show that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for Minimum Hitting Set problem under plausible complexity assumptions.	A novel reduction of the Minimum Probe Station Selection problem to the Minimum Hitting Set problem.	Computationally feasible to implement.	The algorithms in this paper ensure the availability of k independent probe paths but do not aim to optimize probe traffic or the localization time.
3	Exhaustive Search	Probe station selection	Exponential	Probe station placement should be such that Probe stations are able to send enough probes to all the nodes to localize the fault.	Is a combinatorial approach		Computationally too expensive to be deployed practically for large networks.

6 Way Ahead

In this section we would like to describe the motivation for survey of these algorithms. Distributed networks are complex in nature and day by day its utilization and dynamism is increasingly changing which makes these networks more prone to failures. To cope up with these issues network fault management is important. After surveying the existing research in this field it gave us pointers for future research.

Most of the researchers have worked in isolation on the problem related either to probe station or to probe set selection algorithms only. We intend to integrate both these methods and develop a method which outperforms the results achieved through individualist approach. Experimentation will be done using OMNeT++ simulator as it supports the required functionality and is one of the most popular open source software with a very good GUI.

7 Conclusion

In this paper we have evaluated popular strategies used by researchers for solving the problem of minimizing the probe set for fault detection and localization.

The survey indicates that the greedy search approach for probe set selection is better and optimized than the other available active probing algorithms for fault detection and localization. Use of pre-planned probing is not an efficient approach for monitoring distributed networks due to the overhead of increase in management traffic. Adoptive probing is a hybrid approach which will be the best option for network fault management as it overcomes all the drawback of both active and passive monitoring.

REFERENCES

[1] M. Brodie, I. Rish, S. Ma, Optimizing probe selection for fault localization, In the 12th International Workshop on Distributed Systems Operations Management, 2001.

[2] M. Brodie, I. Rish, S. Ma, G, Grabarnik, N. Odintsova, Active probing, Technical Report IBM, 2002.

[3] M. Natu, A. S. Sethi, Active probing approach for fault localization in computer network", In E2EMON'06, Vancouver, Canada, 2006.

[4] M. Natu, A. S. Sethi, Efficient probing techniques for fault diagnosis, Second International Conference on Internet Monitoring and Protection, IEEE, 2007.

[5] M. Brodie, I. Rish, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, K. Hernandez, Adaptive diagnosis in distributed systems, Technical Report IBM, 2002.

[6] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, D. Ohsie, High speed and robust event correlation, IEEE communications Magazine 34 (5) (1996) 82-90.

[7] R. Gardner, D. Harle, Alarm correlation and network fault resolution using Kohonen Self-Organizing map, Globecom 97 proceedings, pp. 1398-1402, 1997.

[8] A. T. Bouloutas, G. W. Hart, M. Shwartz, Fault identification using a FSM model with unreliable partially observed data sequences, IEEE Transactions on Communications, 41(7):pp. 1074-1083, 1993.

[9] C. Wang, M. Schwartz, Identification of faulty links in dynamic-routed networks, IEEE Journal on Selected Areas in Communications,11 (3) 1449-1460, 1993.

[10] Mark Brodie, Irina Rish, Sheng Ma, Alina Beygelzimer, Natalia Odintsova, Strategies for Problem Determination using Probing, In IEEE INFOCOM.

[11] M. Natu and A. S. Sethi. Probe station placement for robust monitoring of networks. Submitted to Journal of Network and Systems Management.

[12] M. Steinder and A. S. Sethi. A survey of fault localization techniques in computer networks.Science of Computer Programming, Special Edition, 53(2): 165-194, Nov., 2004.

[13] Abduljalil A. Mohamed and Otman Basir. A New Probing Scheme for Fault Detection and Identification, in IEEE 2009.

[14] Deepak Jeswani, Nakul Korde, Dinesh Patil, Maitreya Natu and John Augustine. Probe Station Selection Algorithms for Fault Management in Computer Networks, in IEEE 2010.

[15] Yongjin Liu, Yanan Wang and Fangping Li. Fault Management of Computer Networks based on Probe Station Selection Algorithms, in International Coriference on Educational and Network Technology (ICENT 2010)

[16] Likun Yu, Xuesong Qiu,Yan Qiao, Xingyu Chen,Yanguang Liu. Optimizing Probe Selection Algorithms For Fault Localization , in IEEE 2010.

[17] Maitreya Natu · Adarshpal S. Sethi · Errol L. Lloyd. Efficient probe selection algorithms for fault diagnosis , Springer Science+Business Media, LLC 2008.

[18] Likun Yu, Lu Cheng, Yan Qiao,Yiguo Yuan,Xingyu Chen. An Efficient Active Probing Approach Based On The Combination Of Online And Offline Strategies, , in IEEE 2010.

[19] Francois Baccelli, Sridhar Machiraju, Darryl Veitch, *Fellow, IEEE*, and Jean Bolot. Probing for Loss: The Case Against Probe Trains,in IEEE Communications Letters, Vol. 15, No. 5, May 2011

[20] Mark Brodie , Irina Rish , Sheng Ma , Natalia Odintsova, Active probing strategies for problem diagnosis in distributed systems, IJCAI, 2003.

