

# NETWORK CODING FOR STAR AND MESH NETWORKS

Juma Ben Saleh

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements for

The Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

2012

© Juma Ben Saleh 2012

**CONCORDIA UNIVERSITY**  
**SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: Juma Ben Saleh

Entitled: Network Coding for Star and Mesh Networks

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

\_\_\_\_\_ Chair  
Dr. D. Ford

\_\_\_\_\_ External Examiner  
Dr. J-C. Grégoire

\_\_\_\_\_ External to Program  
Dr. T. Zayed

\_\_\_\_\_ Examiner  
Dr. A. Agarwal

\_\_\_\_\_ Examiner  
Dr. Y.R. Shayan

\_\_\_\_\_ Supervisor  
Dr. A.K. Elhakeem

Approved by \_\_\_\_\_  
Dr. J.X. Zhang, Graduate Program Director

January 12, 2012

\_\_\_\_\_  
Dr. Robin A.L. Drew, Dean  
Faculty of Engineering & Computer Science

# ABSTRACT

## Network Coding For Star and Mesh Networks

Juma Ben Saleh, Ph.D.

Concordia University, 2012

This thesis introduces new network coding techniques to improve the file sharing and video streaming performance of wireless star and mesh networks. In this thesis we propose a new XOR based scheduling algorithm for network coding in cooperative local repair. The proposed algorithm commences in three phases. In the first phase, nodes exchange packets availability vectors. These vectors are functions of the probability of correct packet reception over the channel. This is followed by a short period of distributed scheduling where the nodes execute the processing algorithm which tries to minimize the total transmission time. In the third phase, nodes transmit the encoded packets as per the decision of the scheduling algorithm. Simulation results show improvement in system throughput and processing delay for the proposed algorithm. We also study the trade-offs between file sizes, processing delays, number of users and packet availability. In the sequel we display the favorable effects of file segmentation on the performance of the proposed scheduling algorithm. Furthermore, the upper bound on the performance and the analysis of the proposed scheduling algorithm are derived.

Also, in this thesis, the effects of random network coding on code division multiple access/time division duplex (CDMA/TDD) platforms for wireless mesh networks are studied and evaluated. A multi-hop mesh network with single source and multiple receiving nodes is assumed. For reliable data transfer, a Selective Repeat ARQ

protocol is used. Two scenarios are evaluated for their efficiency. In scenario 1, but not in scenario 2, random network coding is applied to CDMA/TDD wireless mesh networks. The delay and delay jitter for both scenarios are computed. The study also focuses on the effects of uncontrolled parameters such as the minimum number of neighbors and the network connectivity, and of controlled parameters such as Galois Field (GF) size, packet size, number of Walsh functions employed at each node and the Processing Gain. The analysis and simulation results show that applying random network coding to CDMA/TDD systems in wireless mesh networks could provide a noticeable improvement in overall efficiency.

We also propose a cross layer approach for the Random Network coded-Code Division Multiple Access/Time Division Duplex (RNC-CDMA/TDD) wireless mesh networks. The proposed algorithm selects the number of assigned Walsh functions depending on the network topology. Two strategies of Walsh function assignments are proposed. In the first, nodes determine the number of their assigned Walsh functions depending on the neighbor with the maximum number of neighbors, which we call the worst case assignment. In the second, nodes determine the number of their assigned Walsh functions depending on the need for each transmission. Simulation results show the possible achievable improvement in the system performance, delay and delay jitter due to cross layer design.

## ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Ahmed K. Elhakeem for his support, help, guidance, and patience.

## DEDICATION

*To my parents, to my sisters and brothers, to my wife, and to my children*

# Contents

Chapter 1 INTRODUCTION .....	1
1.1 Introduction.....	1
1.2 Preliminaries .....	1
1.2.1 Network coding.....	1
1.2.2 CDMA/TDD .....	3
1.2.3 Cross layer design.....	5
1.3 Literature Review .....	7
1.4 Motivations and Contributions.....	13
1.5 Outline .....	14
Chapter 2 A SCHEDULING APPROACH TO NETWORK CODING FOR WIRELESS LOCAL REPAIR .....	15
2.1 Introduction.....	15
2.2 The Scheduling Algorithm Overview.....	18
2.3 Scheduling Algorithm and MAC Procedure.....	22
2.4 Performance Analysis of the Proposed Algorithm.....	34
2.4.1 The upper bound performance .....	35
2.4.2 Analysis .....	37
2.5 Results .....	46
2.6 Conclusion .....	55

Chapter 3 RANDOM NETWORK CODING FOR CDMA/TDD WIRELESS MESH NETWORKS.....	56
3.1 Introduction.....	56
3.2 System Model .....	58
3.2.1 Scenario 1.....	60
3.2.2 Scenario 2 (No network coding).....	67
3.3 Performance Analysis.....	72
3.3.1 Analysis by enumeration.....	75
3.3.2 Markov Chain Analysis .....	77
3.4 Results .....	85
3.5 Conclusion.....	95
Chapter 4 CROSS LAYER DESIGN FOR RANDOM NETWORK CODING CDMA/TDD WIRELESS MESH NETWORKS.....	96
4.1 Introduction.....	96
4.2 System Model .....	97
4.2.1. The source fixed assignment:.....	98
4.2.2. The source worst case assignment:.....	98
4.2.3. The source average assignment:.....	98
4.2.4. The intermediate fixed assignment:.....	99
4.2.5. The intermediate worst case assignment:.....	99
4.2.6. The intermediate dynamic assignment:.....	99
4.3 Simulation Results.....	105



4.4 Conclusion .....	112
Chapter 5 CONCLUSION.....	113
5.1 Future Work.....	115
References .....	116

## List of Figures

Figure 1.1 Example of XOR network coding .....	2
Figure 2.1 Network topology .....	18
Figure 2.2 Flow chart of the scheduling algorithm .....	23
Figure 2.3 the improvement factors of: the proposed scheduling algorithm, with no NC and cooperation between nodes, with no NC and no cooperation between nodes, the upper bound, and the analysis. ....	47
Figure 2.4 The upper bound on the improvement factors .....	48
Figure 2.5 Simulation results.....	49
Figure 2.6 Processing delay (in seconds) vs. number of nodes in the new three-phase algorithm	50
Figure 2.7 Processing delay (in seconds) vs. file size in the new three-phase algorithm.....	51
Figure 2.8 Segmentation effects on the processing delay in the new three-phase algorithm .....	53
Figure 2.9 Segmentation effects on the improvement factor in the new three phase algorithm....	54
Figure 3.1 System topology as an example .....	59
Figure3.2 Block diagram of the transmitter at the intermediate nodes (scenario 1) .....	61
Figure 3.3 Block diagram of the receiver (scenario 1).....	65
Figure 3.4 The transmitted packet format .....	66
Figure 3.5 Markov chain representation of a node having two neighbors, and $w_r = 2$ , and $\Delta = 2$ ..	78
Figure 3.6 The efficiency of the scenarios vs. the minimum number of neighbors. Connectivity =4, $N=20$ nodes $g = 4$ , $gs = 8$ , number of hops = 6, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $I/SNR_{th} = 0.01$ , $m = 6$ , $w_s = 4$ , $w_r = 4$ and the maximum number of neighbors = 6. ....	87
Figure 3.7 The efficiency of the systems vs. the network connectivity, where $N = 24$ nodes, $g = 4$ , $gs = 8$ , number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $I/SNR_{th} = 0.01$ , $m =$	

6, $w_s = 4$ , $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2. .....	89
Figure 3.8 The efficiency of the systems vs. the network connectivity, where $N = 24$ nodes, $g = 4$ , $g_s = 8$ , number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 64$ , $1/SNR_{th} = 0.01$ , $m = 6$ , $w_s = 4$ , $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2. ....	90
Figure 3.9 The systems efficiency of the scenarios vs. packet size, where $N = 20$ , $g = 4$ , $g_s = 8$ , $P_G = 32$ , $1/SNR_{th} = 0.01$ , connectivity = 3, $L = 120$ packets, number of hops = 5, $m = 6$ bits, $w_s = 4$ , $w_r = 4$ , minimum number of neighbors = 2 and the maximum number of neighbors = 6. ....	91
Figure 3.10 The systems efficiency of the scenarios vs. $m$ , where $N = 24$ , $g = 4$ , $g_s = 8$ , $P_G = 32$ , $1/SNR_{th} = 0.01$ , connectivity = 3.5, $L = 120$ packets, number of hops = 5, $n_b = 1000$ bits, $w_s = 4$ , $w_r = 4$ , minimum number of neighbors = 2 and the maximum number of neighbors = 6. ....	92
Figure 3.11 The delay vs. connectivity, where $N = 20$ nodes, $g = 4$ , $g_s = 8$ , number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $1/SNR_{th} = 0.01$ , $m = 6$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2. ....	92
Figure 3.12 The delay jitter vs. connectivity, where $N = 20$ nodes, $g = 4$ , $g_s = 8$ , number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $1/SNR_{th} = 0.01$ , $m = 6$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2. ....	93
Figure 3.13 Number of Walsh functions at the source node vs. the efficiency, where $N = 24$ nodes, number of Walsh functions at intermediate nodes is 3, number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $1/SNR_{th} = 0.01$ , $m = 6$ , $w_s = 4$ , $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2. ....	94
Figure 4.1 The flow chart of the intermediate dynamic assignment presented in section 4.2.6. .	101
Figure 4.2 The efficiency of the scenarios vs. the network connectivity for some of the presented Walsh functions strategies. $N = 20$ nodes number of hops = 5, $L = 120$ packets, $n_b = 1000$ bits, $P_G = 32$ , $1/SNR_{th} = 0.01$ , $m = 6$ , $w_s = 4$ , $w_r = 4$ and the maximum number of neighbors = 6. ....	108

Figure 4.3 The efficiency of the systems vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $I/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$  and maximum number of neighbors = 6. ... 109

Figure 4.4 The delay vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops =5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $I/SNR_{th} = 0.01$ ,  $m = 6$ , and maximum number of neighbors =6. .... 110

Figure 4.5 The delay jitter vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $I/SNR_{th} = 0.01$ ,  $m = 6$ , and the maximum number of neighbors=6..... 111

## List of tables

Table 1-1 OSI Model .....	6
Table 1-2 TCP/IP Model .....	6
Table 2-1 Information table at each node describing the received and missed packets of all users .....	20
Table 2-2 Information table at each node describing the received and missed packets of all users .....	31
Table 2-3 Information table at each node describing the received and missed packets of all users .....	33
Table 2-4 A possible scenario to combine (N-2) packets that can help (N-1) nodes .....	40
Table 2-5 A possible scenario to combine (N-3) packets that can help (N-1) nodes .....	41
Table 2-6 A possible scenario of combining (N-2) packets that can help (N-2) nodes .....	43

## LIST OF SYMBOLS (LOS)

$N$	Number of considered nodes
$L$	The file size
$(N - i)$	Number of original packets constituting a coded packet (combination level)
$(N - j)$	Number of nodes that will benefit from the coded packet
$\bar{d}_s$	A vector of the received packets identities at node $s$
$\bar{n}_{v,i}$	The $v$ set of nodes at the $(N - i)$ combination level
$\bar{l}_{v,i}$	The identities of the commonly packets received in set $v$
$\bar{k}_i$	A vector of packet's identity that will be considered at the $(N - i)$ combination level
$m$	the Galois Field size
$ \bar{k}_i $	Number of packets in $\bar{k}_i$
$\bar{C}_{s,i}$	The identities of packets received by node $s$ that intersect with $\bar{k}_i$
$ \bar{C}_{s,i} $	Number of packets in $\bar{C}_{s,i}$
$p_{x,e}$	The candidate $(N - i)$ packets to be combined at node $x$ where $e = 1, 2, \dots, N - i$
$P_b$	The probability of bit error rate
$P_c$	The probability of packet success
$P_M$	The probability of $M$ nodes have received a particular packet
$P'_M$	The probability of a packet eventually being received by all $M$ nodes
$R$	The average number of nodes that will receive a certain packet
$k$	The average number of received packets at all $N$ nodes

$P(L' = D)$	The distribution of the window size utilized for local repair
$P'_N$	The probability that a packet is received by all $N$ nodes
$T$	The required number of transmissions for a given window size
$T_{max}$	The maximum required number of transmissions for a given window size
$T_{min}$	The minimum required number of transmissions for a given window size
$\Psi_{up}$	The upper bound on the improvement factor
$B_{N-i,j,h}$	The average number of possible coded packets at combination level $(N - i)$ , and $j$ nodes that will not benefit from the coded packet including the transmitted node and $h$ packets scheduled for $j \geq 2$
$A_M$	The average number of packets received by $M$ nodes
$R_{N-i,i,h}$	The remaining nonscheduled packets received by $(N - i)$ nodes
$G$	The total number of transmissions
$\Psi$	The improvement factor
$g$	Number of Walsh functions at intermediate node
$g_s$	Number of Walsh functions at the source node
$\alpha_{j,l,k}$	The $j^{th}$ randomly chosen coefficient from a finite field to generate the $l^{th}$ image of the $i^{th}$ NCP by node $k$
$X_j$	The $j^{th}$ original data packet
$Y_{i,l,k}$	The generated $l^{th}$ image of an NCP related to group $i$ of the original data packets by node $k$

$w_g$	The $g^{th}$ Walsh function at a certain node
$Z_k$	The signal generated by node $k$ after Walsh spreading
$C_{S_k}$	The used scrambling code at node $k$
$V_k$	The signal transmitted by node $k$
$w_s$	The send window size
$w_r$	The receive window size
$w_q$	The number of assigned Walsh functions at node $q$
$M_d$	The number of neighbors of a receiving node $d$
$r_d(t)$	The received signal at the intermediate node $d$
$S$	The total transmission power at each node
$N_0$	The noise power spectral density
$N'$	The total power of the AWGN after CDMA spreading
$E_b$	The energy per bit
$R_b$	The bit rate
$n_b$	Number of bits per packet
$F$	Number of time slots required until all nodes receive the entire file for scenario 1
$F'$	Number of time slots required until all nodes receive the entire file for scenario 2
$W$	The total bandwidth
$\eta_{NC}$	The efficiency of the system with NC
$\eta_{NNC}$	The efficiency of the system without NC



$T_{n,x,H}$	The time slot when node $x$ has received the original data packet $n$ in topology $H$
$T_{n,H}$	The time slot when the $n^{th}$ original data packet should have been transmitted from the source node for topology $H$
$D_{n,x,H}$	The delay of the $n^{th}$ original data packet at node $x$ for a certain generated topology $H$
$D_{n,H}$	The average delay of the original data packet $n$ averaged over all intermediate nodes for topology $H$
$D_H$	The average delay of all packets at all intermediate nodes for topology $H$
$\delta$	The number of generated topologies
$D$	The average delay of all packets at all nodes for all topologies
$\bar{J}_{n,x,H}$	The delay jitter between packets $n$ and $n+1$ at node $x$ for topology $H$
$\bar{J}_{x,H}$	The delay jitter for the data file at node $x$ for topology $H$
$\overline{\sigma_{x,H}}$	The delay jitter
$\sigma$	The average delay jitter
$P_{c,M}$	The probability of packet success at the subject node for a given $M$ neighbors
$\xi$	Network connectivity
$Q$	The maximum number of neighbors of any node in the network
$J$	The minimum number of neighbors of any node in the network
$\varrho$	The total number of possible topologies for $N$ nodes
$\epsilon_1$	The number of nodes that have $J$ neighbors

$C_2$	The number of nodes that have $(J + I)$ neighbors
$\Omega$	The total possible topologies that may exist for $(N-1)$ nodes
$P(M/\xi)$	The probability that the subject node has $M$ neighbors
$\Phi$	The total possible topologies if $M$ nodes out of the $(N - I)$ nodes have $U$ neighbors
$I_1$	The number of neighbors of the first neighbor of the subject node
$I_2$	The number of neighbors of the second neighbor of the subject node
$P(U/M, \xi)$	The probability that the $M$ neighbors of the subject node have $U$ neighbors
$\mu_U$	The average number of independent images received by the subject node per each receiving time slot for a given $U$
$B_{M,\xi}$	The average number of independent images received by the subject node per each receiving time slot for a given $M$ and $\xi$
$B_\xi$	The average number of independent images received by the subject node per each receiving time slot for a given $\xi$
$F_\xi$	The average number of time slots needed for the subject node to receive the entire file ( $L$ original packets) for a given $\xi$
$w_t$	The window length of NCPs that have been received by all the neighbors of the subject node
$w_r$	The number of the outstanding NCPs that have been received by all of the neighbors of the subject node but have not yet been acknowledged by the subject node

$\Delta$	The ratio between the number of assigned Walsh functions at the source node and the number of assigned Walsh functions at the intermediate nodes
$\Gamma$	The number of states of the Markov chain
$\gamma_{i,j,M}$	The transition probability to move form state $i$ to state $j$ at the subject node for a certain number of neighbors $M$
$\psi_{i,j}$	The total possibility of combinations of transmitted NCPs from $\lambda$ neighbors such that the availability of packets at the subject node moves from state $i$ to state $j$
$d$	The Euclidean distance between state $i$ and state $j$
$ i $	The cardinality of the current state
$P(s_i/M, \xi)$	The probability of state I for a given number of neighbors ( $M$ ) and network connectivity ( $\xi$ )
$\beta$	The number of decodable NCPs per a receiving time slot
$C_M$	The average number of decoded NCPs at the subject node per each receiving time slot for a given $M$
$C$	The average number of decodable NCPs at the subject node per each receiving time slot
$P_{c_s,M}$	The probability of a source packet success at a node in the first hop for a given number of neighbors of the node
$P'(s)$	The probability that a certain node in the first hop receives the source packet for a given $\xi$

$\partial$	The number of hops in the network
$\omega$	The average number of nodes per hop
$P(s)$	The probability that at least one node in the first hop receives the source packet

## LIST OF ABBREVIATION (LOA)

Ack	Acknowledgment
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BS	Base Station
CAOR	Coding Aware Opportunistic Routing
CDMA	Code Division Multiple Access
CDMA/TDD	Code Division Multiple Access/Time Division Duplex
CLD	Cross Layer Design
CRC	Cyclic Redundancy Check
DCAR	Distributed Coding-Aware Routing
DS	Direct-sequence
DS-CDMA	Direct-sequence code division multiple access
DS/FH	Hybrid direct-sequence/ Frequency-hop
DS/FH CDMA	Hybrid direct-sequence/ Frequency-hop code division multiple access
FDD	Frequency-Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FH	Frequency-hop
FH-CDMA	Frequency-hop code division multiple access

HARQ	Hybrid Automatic Repeat Request
GF	Galois Field
LCM	Linear code multicast
LNC	Linear network coding
MAC	Multiple Access Control
MS	Mobile subscriber
NC	Network coding
NC-CCPR	Centralized Network Coding for Cooperative peer-to-peer Local Repair
NC-DCPR	Distributed Network Coding for Cooperative peer-to-peer Local Repair
NCP	Network Coded Packet
OFDMA	Orthogonal Frequency Division Multiple Access
OMNC	Optimized Multipath Network Coding
OSI	Open systems interconnection
QoS	Quality of Service
RNC	Random linear network coding
RNC-CDMA/TDD	Random Network coded-Code Division Multiple Access/Time Division Duplex
TDD	Time-Division Duplex
TDMA	Time Division Multiple Access
WLAN	Wireless Local Area Network
WMN	Wireless mesh network

WWAN      Wireless Wide Area Network

# Chapter 1 INTRODUCTION

## 1.1 Introduction

Applying network coding (NC) could provide several benefits, such as higher speed, reliable communication, etc. Without NC, intermediate nodes merely replicate received packets and forward them to their neighbors. However, with NC, intermediate nodes code/combine a number of received packets and then send the resulting coded packet to their neighbors. This ability of coding is an asset of the NC and helps a node, who receives from different neighbors, to make use of the intentional and unintentional transmissions from neighboring nodes.

This chapter introduces some concepts necessary for complete understanding of NC and its possible benefits. This chapter will also outline the contributions and topics of subsequent chapters.

## 1.2 Preliminaries

In this section, we present an overview of several topics related to the thesis, particularly NC, CDMA/TDD systems and cross layer design (CLD).

### 1.2.1 Network coding

Recently, NC was introduced to deal with the information forwarding in a network. NC was initiated by Ahlsweede [1]. Without NC, intermediate nodes just replicate and forward the received packets to their neighbors. On the other hand, with



NC, intermediate nodes perform coding/combining of a number of received packets and then send the coded packets to their neighbors. The basic way to create a coded packet is to perform the operation XOR on a certain number of the original packets [2-4]. Then a neighbor node receiving such a coded packet will be able to find one of the component packets, provided that it has all other original packets constituting the coded packet. For example, in figure 1.1, node  $n_1$  has received packets  $P_1$ ,  $P_2$ ,  $P_3$ , and  $P_4$ , node  $n_2$  has received packets  $P_1$ ,  $P_3$ , and  $P_4$ , node  $n_3$  has received packets  $P_2$ ,  $P_3$ , and  $P_4$ , node  $n_4$  has received packets  $P_1$ ,  $P_2$ , and  $P_3$ , and node  $n_5$  has received packets  $P_1$ ,  $P_2$ , and  $P_4$ .

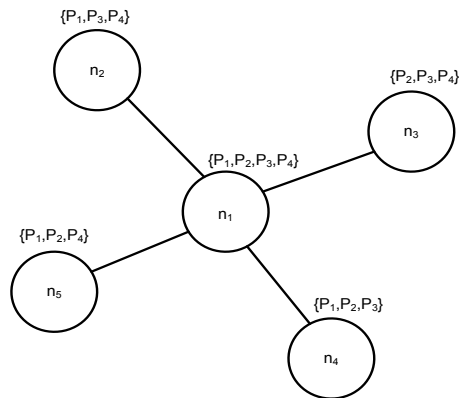


Figure 1.1 Example of XOR network coding

Let us assume that all nodes need all four packets. If  $n_1$  encodes all four packets by performing the operation XOR on all of them together and transmitting the coded packet, then any node of the neighboring nodes ( $n_2$ ,  $n_3$ ,  $n_4$ , and  $n_5$ ) that receives the coded packet will be able to get the original packet it is missing.

In a different technique, packets received at a certain intermediate node are multiplied by a set of coefficients. These coefficients are preset, used by all nodes and for

all packets [5]. This linear network coding (LNC) is sufficient for small and static networks. For large and dynamic networks, a random linear network coding (RNC) is preferable, where each intermediate node chooses its coefficients randomly and independently from a finite field [6] whenever a coded packet is formulated. The received packets are weighted according to the chosen coefficients and combined to create the coded packet to be transmitted. These coefficients are also transmitted as headers in the coded packet for decoding purposes at the receiving node.

### **1.2.2 CDMA/TDD**

Some different techniques used to access the shared channel in multi-user systems are Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and spread spectrum technique (Code Division Multiple Access (CDMA)). The essential idea of spread spectrum is to disperse the spectrum of the information signal over a huge bandwidth. The three well known spread spectrum techniques are direct-sequence (DS), frequency-hop (FH), and hybrid direct-sequence/ frequency-hop (DS/FH). In the direct-sequence code division multiple access (DS-CDMA), users share the entire bandwidth and transmit their data at the same time, but each user is assigned a different code (Walsh code or Walsh function) to access the channel [7]. Each user has a unique Walsh code, which is orthogonal to the Walsh codes of the other users in the cell. The chip rate of the Walsh code is higher than the data rate which causes spreading of the data. The ratio between the chip rate and the data rate is known as Processing Gain. The receiving user, depending on the channel, receives the overlapped signals that were

transmitted by his different neighbors. The receiver decodes the intended data by de-spreading the overlapped received signals using the same Walsh code used at the intended transmitter. DS-CDMA channel access technique has been used widely, e.g. in W-CDMA [8], IEEE 802.11b, etc. In the frequency-hop code division multiple access (FH-CDMA), the carrier frequency is periodically changed [7]. Such a technique is used in handheld transceivers (Walkie-talkie) and in the military. In the hybrid direct-sequence/ frequency-hop code division multiple access (DS/FH CDMA), both DS and FH are used to make use of the benefits provided by the two techniques [7]. This type of channel access is often used in the military.

A communication system with two parties, that provides communication in two directions, e.g. the base station (BS) and mobile subscriber (MS) in cellular system, is known as a duplex communication system. The Frequency-Division Duplex (FDD) and the Time-Division Duplex (TDD) are the two duplex techniques that are used in communication systems. In TDD mode the uplink and downlink are separated in time, which implies that the two parties transmit at different times but using the same frequency [9]. TDD is preferred over FDD due to its design simplicity, where the same set of electronic equipment is used, at BS or MS for both uplink and downlink [9]. Also, since the uplink and downlink use the same frequency, the uplink and downlink channels are identical. As a result, this can be used for power control [9].

In CDMA/TDD systems, users use the CDMA to access the channel with uplink and downlink separated in time. CDMA/TDD systems have been a subject of great interest in cellular mobile systems [10], [11], [12] , and [8]. In addition to Walsh functions, spread spectrum systems use scrambling code for security. In cellular systems,

the BS is responsible for controlling the entire process, e.g. the Walsh code assignments to all MS, power control, and so forth. Recently, CDMA/TDD has been applied to wireless mesh networks (WMNs) [13]. In WMNs, there is no BS, and nodes can communicate using routing techniques. In CDMA/TDD WMNs, the identities of the used Walsh functions and the scrambling code at the transmitting node are transmitted in the packet for the decoding purposes. In chapter 3, more details about the packet format of CDMA/TDD systems for WMNs are introduced.

### **1.2.3 Cross layer design**

There are different communication networks, such as computer, television and cellular networks, etc. A user in a network could request a service supported by a different network. For example, one could check his/her e-mail on a cell phone. Thus, communication is a very complex process. Not only that, but a network also needs to be updatable to accommodate any newly developed service. Therefore, a complete network architecture is required. Layered architectures, such as the seven-layer OSI reference model and TCP/IP, have divided the networking tasks into layers, and each layer is responsible to provide a certain service to the layer above it [14]. The layers of the seven-layer OSI reference model from top to bottom are application layer, presentation layer, session layer, transport layer, network layer, data link layer and physical layer, as shown in table 1-1.

Table 1-1 OSI Model

Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer

Table 1-2 shows the layers of TCP/IP model. The layers of the TCP/IP model from top to bottom are application layer, transport layer, internet layer and network interface.

Table 1-2 TCP/IP Model

Application layer
Transport layer
internet layer
Network interface

In layered network architecture each layer provides a service to the next upper layer. Beyond that, the lower layer does not interfere with the inner workings of the upper layer. On the other hand, in cross layered networks there is coupling between layers, so as to improve the overall performance. For example, routing techniques inherent to network layer can make use of information from physical and data link layers. This information could be probability of packet transmission errors in various links, etc. In this thesis, we investigate the combined effect of NC and cross layer design.

### 1.3 Literature Review

NC is a research topic that has received wide attention recently. In [1], the *max flow min cut* theorem is presented. The authors in [1] show that, with NC for single source multicast networks it is possible to achieve throughput equal to the minimum of the maximum flows for the cut that separates source from destinations. A generic *linear code multicast* (LCM) was proposed [5]. In this algorithm, the output of the intermediate nodes is basically a linear combination of its inputs. It was shown that, for a single-source multicast network and finite symbol size, using LCM, a rate that equals the receiver's max-flow can be achieved for every receiver. In [15] an algebraic approach for LNC is proposed. The algebraic approach reduces the time required to find the linear solution. Work in [16] shows that it is possible to find an optimal LNC for a given network topology. The problem with this algorithm is the processing time which increases exponentially with the network size; also, this algorithm is efficient only for single source

networks. In [17], the authors introduce the randomized NC. The authors in [18] show the possible achievable benefit that randomized NC offers compared with traditional routing protocols. In [6] a distributed RNC approach is presented. With this approach, intermediate nodes make random linear combinations of their incoming packets and transmit these coded packets in their outgoing links over some finite field. In this work it was shown that, using this NC algorithm in multi-source multicast networks, maximum capacity can be achieved.

Although NC was originally investigated for wired networks, applying NC in wireless networks could provide better improvement in the system throughput. In this thesis, we restrict the study to works in NC in wireless networks. Many works in NC for wireless networks have been proposed. Most of the works were either to improve the system throughput or to reduce the required energy. In [19] and [20], an opportunistic scheduling for wireless network coding is presented. In this opportunistic scheduling, nodes send coded packets to receivers within their power range through good channel condition. This in turn will improve the transmission rate. The authors in [21] have focused on how to implement NC in a WMN. COPE, which is the first practical opportunistic network coding approach for wireless networks, was introduced in [2]. In [22], [23] a theoretical formulation of the COPE throughput is derived. The problems with COPE [2] are the conventional shortest path routing protocol and queuing technique which may not utilize the coding opportunity, thus limiting the benefit of NC. A recent work [24] has proposed a more efficient COPE architecture for NC in wireless networks. The authors in [24] use different queuing and scheduling algorithm compared to [2] in order to improve the system throughput and make COPE more efficient for real time applications. In [25], a

practical localized network coding in WMNs (BFLY) for multiple unicast sessions is presented. In the BFLY algorithm, only some determined nodes are allowed to perform coding. Based on the knowledge of local topologies and routing of a particulate flow, a selected node will decide whether to perform NC or not. The problem with BFLY is that scheduling is not considered to improve the possibility of having network coding opportunity at the selected node. In [26], a NC based algorithm for broadcast in mobile Ad hoc networks is presented. This algorithm is very similar to [2] in the way it works. The problem with this algorithm is that the selected forwarding nodes are chosen independently on the coding opportunities at these nodes which will limit the possible achievable gain. Different routing techniques have been proposed in order to utilize the maximum possible improvement provided by NC. The authors in [27] present a Distributed Coding-Aware Routing in wireless networks (DCAR). Path selection depends on the coding possibility at each intermediate node of selected path. In [28], a Coding Aware Opportunistic Routing (CAOR) is presented. This approach tries to increase the possibility of having coded packets when the route for a certain flow is selected. The problem with CAOR is the need for global information of the current traffic distributions, such information may change very fast. MORE, a MAC independent opportunistic routing protocol for erroneous wireless network, is presented in [29]. In MORE, the source node first breaks up the file into batches, each of a certain number of packets. When the source is allowed to transmit, it transmits a random linear combination of a batch and a list of the forwarding nodes that can participate to forward the packet. The nodes in the forwarding list that hear the transmitted coded packets will transmit a random linear combination of the received packets. This randomness will ensure that



nodes in the forwarding list that have received the same transmission will not transmit the same data. The challenging points with MORE are how many transmissions a source node should transmit for a batch and when the forwarder nodes should stop transmitting coded packets. Recently, Optimized Multipath Network Coding (OMNC) was proposed in [30]. In the OMNC protocol, coding, broadcasting and multipath routing are optimized in order to maximize the system throughput of erroneous wireless networks. A distributed rate control algorithm is used at each node in order to match the encoding rate with the broadcasting rate, depending on the channel congestion status in the neighborhood. The authors in [30] show that, OMNC outperforms the techniques in [29]. In [31] and [32], the authors have proposed a routing algorithm for multicast in wireless networks that can achieve the maximum flow. Another way to increase the possible achievable gain offered by NC can be realized by adjusting the data rate or by adjusting the transmitted power. The work in [33] studies the effects of data rate selection on NC in WMNs. It was shown that, by selecting a suitable rate for the whole network, coding opportunities will increase at the intermediate nodes.

In all the works discussed, NC executes coding on digital data, for example packets and bits. However, NC is possible in the physical layer where signals are mixed instead [34, 35]. In physical layer network coding [34], senders are picked so that the transmitted signals by these nodes will interfere with each other at some nodes. Nodes store the received signal for the decoding process for the next received interfered signal. Nodes that receive the interfered signals will decode them using the previous stored signal and transmit the new received signal that results from the decoding process. In [34], the authors have assumed that the interfered signals are synchronized and it is not clear how

the algorithm will deal with interfered signals that are not synchronized. Work in [35] has considered the situations where the interfered signals are not synchronized and channels have different phase shifts. In this research we will not investigate physical layer network coding.

Some works on CLD of NC and multiple access control (MAC) for Ad hoc networks are presented in [36, 37]. In [36, 37], the authors have derived the conflict-free network realizations and activate them separately to optimize the network throughput. Also, they have derived a practical solution to construct the network codes on the simplified sub-tree graphs using graph coloring and then converting them to conflict-free transmission schedules. For interference and energy aware network coding where most of NC opportunity involve only two unicast sessions (pairwise intersession network coding), a jointly optimal coding, scheduling and rate control algorithm is proposed in [38]. The authors in [38] have studied the impact of imperfect scheduling and random and deterministic packets arrival and departure on the system throughput. the authors in [39] have coupled several MAC and scheduling schemes with different network coding combination strategies. They show that the impact of the MAC layer with RNC is small on the packet delivery ratio, where packet delivery ratio is defined as the ratio between the number of received and decoded packets to the number of subject packets at the destination node averaged over all nodes in the network. They have also proposed a timing strategy for packet combinations which offer a high improvement on the system throughput. With timing strategy, a timer is activated after receiving each innovative packet (innovative packet is defined as a coded packet that has different coefficients compared with previously received coded packets at a particular node). After the timer

expires, a node will send a coded packet with certain probability thus leading to a transmitted coded packet that includes more different innovative packets. However, timing strategy introduces some delay due to the timer. The problem with the timing strategy is how to adjust the timer, which reflects on a compromise between delay and the possible improvement in the system throughput.

The upper bound on the throughput gain of NC and broadcasting in wireless networks were investigated in [40-43]. In [40], the authors show that for the network model of Gupta and Kumar [44], NC and broadcasting provide a constant factor improvement in the throughput. Upper and lower bounds on the throughput for the physical model of Gupta and Kumar [44] using NC is presented in [41]. In [42], the authors have proved that the benefit of NC in terms of system throughput and energy for a single multicast scenario is upper bounded.

Most of the works in NC in cooperative local repair use RNC [6] to construct coded packets, e.g. [45-48]. In [45], two heuristic algorithms have been designed. The first one is *centralized network coding for cooperative peer-to-peer local repair* (NC-CCPR), where they have assumed that all nodes have the accurate information about one hop and two hop neighbors. The second one is *distributed network coding for cooperative peer-to-peer local repair* (NC-DCPR). The problem with this algorithm is the header size in the case of large file. In [46], [49] and [47] a structured network coding is presented. The idea of this algorithm is to have some zero coefficients of coded packets so, decoding is possible even in the case of small number of received packets. The authors in [48] proposed a hierarchical network coding scheme where packets in video stream are coded

with different importance. Thus, even when a small number of coded packets are received, a receiver will be able to recover the most important packets.

## **1.4 Motivations and Contributions**

In general, for most of the communication networks (for example, cellular networks, internet networks, and so forth) bandwidth is limited and expensive. Also, for a given limited bandwidth, a certain level of Quality of Service (QoS) is required. This introduces many challenges to network designers. For this reason, efficient use of the bandwidth is a very important issue. NC, introduced by Ahlswede [1], is a technique that can help with the above-mentioned challenges. This thesis addresses the possible advantages of NC to efficiently use the available bandwidth for some different scenarios.

First, we propose a new XOR based scheduling algorithm for NC in cooperative star-oriented local repairs. The proposed scheduling algorithm is for a group of nodes that can hear each other directly. The algorithm makes use of knowledge of the packets availability at neighboring nodes to improve the overall network throughput. The proposed algorithm uses NC to determine which node should transmit and in which time slot (sequential MAC) to provide the best improvement. Also, the upper bound on the improvement factor and the analysis of the proposed algorithm are derived.

Second, this thesis presents the first study of the effects of RNC on CDMA/TDD platforms for WMNs. The study focuses on the effects of uncontrolled parameters, such as the minimum number of neighbors and the network connectivity, and of controlled

parameters, such as the Galois Field (GF) size, the packet size, the number of Walsh functions employed at each node, and the Processing Gain.

Last, we investigate the effects of cross layering between network and data link layers for the Random Network coded-Code Division Multiple Access/Time Division Duplex (RNC-CDMA/TDD) WMN platforms. Different Walsh function assignment strategies at nodes are presented and tested.

## **1.5 Outline**

The rest of this thesis is prepared as follows. In chapter 2, a new XOR based scheduling algorithm for NC in cooperative local repair is presented. Also, the performance of the above new technique is numerically evaluated.

In chapter 3, the effects of RNC on CDMA/TDD platforms for WMNs are studied and evaluated. Two scenarios are evaluated for their efficiency. In scenario 1, RNC is applied to CDMA/TDD system but not scenario 2. A numerical evaluation of the RNC on CDMA/TDD WMNs is presented.

In chapter 4, a cross layer design for the RNC-CDMA/TDD for WMNs is presented and evaluated using computer simulation. Two strategies of Walsh function assignments are proposed.

Finally, conclusion is presented in chapter 5.

Part of this thesis has been published in [50], [3], [51] and [52].

## **Chapter 2 A SCHEDULING APPROACH TO NETWORK CODING FOR WIRELESS LOCAL REPAIR**

### **2.1 Introduction**

Data broadcast is a convenient way of one to all transmissions. Occasional data loss may occur due to channel impairments, etc. Forward Error Correction (FEC) [53] and Automatic Repeat Request (ARQ) [14] techniques have been utilized extensively to mitigate channel effects. Recently, NC has been proposed [4], [54] to improve the performance of FEC and ARQ techniques. Nodes cooperation, or local repair, is another way to overcome the channel effects [55]. In local repair, the BS broadcasts the intended file to all nodes in its range. Due to channel effects, some packets of the transmitted file are probably missed at some nodes. The nodes will cooperate with each other to deliver the missing packets to each node. Local repair is a good choice in scenarios where retransmissions cannot solve problems such as poor channel quality between the BS and the subscribers, and also, in scenarios where a set of nodes with multiple network interfaces (i.e. Wireless Wide Area Network (WWAN) and Wireless Local Area Network (WLAN)) that are looking for a data transmitted by the source node in WWAN [49].

NC was initially proposed by Ahlsweede [1], who showed that the maximum capacity in a network can be achieved by the appropriate mixing of data in the intermediate nodes. Many studies in NC have proved that applying NC to traditional networks could provide significant improvement in overall system throughput [1-3, 5, 6, 15, 17, 18, 24, 28, 29, 32, 45-49, 56]. Most of the studies in NC for cooperative local repair use RNC [6] to

construct coded packets [45-49]. In RNC, each intermediate node chooses its coefficients randomly and independently from a finite field. The received packets are weighted according to the chosen coefficients to create the transmitted coded packet. These coefficients are transmitted as a header in the coded packet for decoding purposes. For large files, the header size becomes significant, which wastes bandwidth. In addition, the destination node will not be able to decode the coded packet until it receives a certain number of different coded packets. These are the two most challenging problems caused by using RNC.

In [45], two heuristic algorithms were designed. The first is centralized network coding for cooperative peer-to-peer local repair (NC-CCPR), where all nodes are assumed to have accurate information about their one-hop and two-hop neighbors. The second algorithm is distributed network coding for cooperative peer-to-peer local repair (NC-DCPR). The problem with this algorithm is the header size when there are large files. A structured NC is presented in [46] and [47], which is based on the concept that there are some zero coefficients of coded packets, so that, decoding is possible even when the number of received packets is small. The problem in this algorithm is the assumption that all nodes in the same ad-hoc network watch the same video. This assumption has been relaxed in [49]. The authors of [48] propose a hierarchical NC scheme wherein packets in a video stream are coded with different levels of importance. Thus, even when a small number of coded packets are received, a receiver will be able to recover the most important packets. Most of the work on NC for cooperative local repair use RNC and try to improve data quality. Although most authors try to reduce the downside of the RNC,

such as the added NC header and the ability of a receiver to decode a coded packet, the effects of RNC is still there.

In this chapter, XOR based NC is used instead of RNC for its simplicity and amenability to local repair in scenarios where a group of nodes can hear each other. In XOR NC, users utilize the availability of some packets at some nodes and their unavailability at other nodes to combine some packets and transmit the combined packet in a single transmission, which would be decodable at most nodes. The basic concept of XOR NC is that one node may combine a number of packets in its possession by performing operation XOR on corresponding bits of the combined packets. The proposed scheduling algorithm selects a set of nodes for local repair transmissions that will minimize the total required number of transmissions. Also, a node that has been selected to transmit a coded packet will be able to know its transmission time slots (sequential MAC determination). Furthermore, all nodes that can hear each other will be able to know the constituting packets of all coded packets (no need to transmit any additional information to the received node about which packets were XORed together).

The outline of this chapter is as follows. The scheduling algorithm overview is in section 2.2. In section 2.3, the scheduling algorithm and MAC procedure of the algorithm are presented, and the upper bound on the improvement factor of the proposed algorithm and the analysis are presented in section 2.4. Numerical and simulation results are presented in section 2.5. The conclusion is in section 2.6.



## 2.2 The Scheduling Algorithm Overview

In this work we restrict the discussion to those algorithms that involve a selected BS. Presence of BS or some control station is justified in many applications like cellular systems even in some WMNs. Most previous works assume a BS (Wireless Wide Area Network (WWAN)) transmits a data file to nodes, and then these nodes use Wireless Local Area Network (WLAN) for local repair [49]. Let us assume the BS transmits a file of  $L$  packets to all nodes in its coverage area. Due to channel effects, some packets will not be received at some nodes. A group of such nodes that can hear each other will cooperate locally to overcome the channel effects.

The proposed algorithm proceeds in three phases. First, the nodes exchange their reception reports vectors. The reception reports describe the received and missed packets at each neighbor. The IEEE 802.11 protocol for MAC channel access is utilized by all the nodes to relay the reception reports.

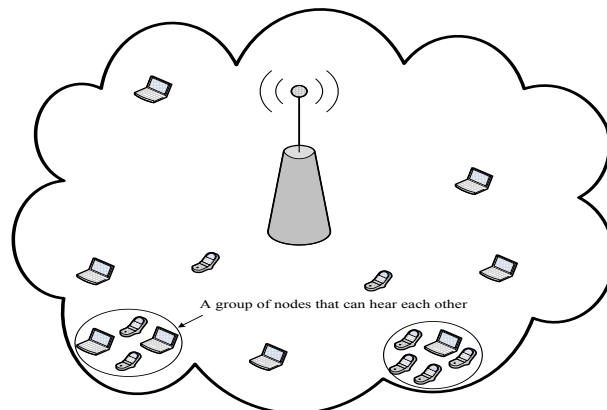


Figure 2.1 Network topology

As an example, assume that in table 2-1, the number of nodes that can hear each other is 4 and that the file size is 10 original data packets. We assume all local repair channels herein are error free. After receiving the reception reports' transmissions, each node will have a table describing the state of all the nodes, including itself, which we call an information table (table 2-1). Observe that in table 2-1, a 'one' means that a node has received the packet correctly and a 'zero' means a node did not receive that packet. Following reception of all the reports, every node will be able to configure the information table. In the second phase various nodes will work independently but use the same scheduling algorithm to determine the best packet combinations that will minimize the number of transmissions and thereby maximize the system throughput. The scheduling algorithm will also determine the identities of the nodes that will transmit the combined packets and their transmission times (sequential MAC determination). Thus, we do not transmit any extra data for coefficient description or any information about which packets have been combined together. Meanwhile, the BS keeps synchronization by means of pilot signals. This pilot will trigger the transmissions of the scheduled combined packets one after another. The BS will allow enough time for the distributed scheduling algorithm to finish. Some nodes may take slightly less time to execute the distributed scheduling algorithm, but they will have to wait for the starting pilot of the BS as indicated above. In the third phase, based on the results of the scheduling algorithm in the second phase, i.e. each node will know the packets that it is supposed to combine, if any, the transmissions of the selected nodes of the combined packets will take place in an order decided by the scheduling algorithm in the second phase. For example, in table 2-1, let us assume packet 1 is 100110101, packet 2 is 101010101, and packet 3 is 011101101.

Table 2-1 Information table at each node describing the received and missed packets of all users

Packet identity	1	2	3	4	5	6	7	8	9	10
Nodes identities										
1	1	0	1	0	1	1	1	1	1	0
2	0	1	1	1	0	1	1	0	0	0
3	1	1	0	0	1	0	1	1	1	0
4	1	1	1	1	0	1	1	0	1	1

From table 2-1, Node 4 will be able to combine packets 1, 2 and 3 (1's in all the combined packets means the node has initially received those packets). The transmitted coded packet by node 4 can be presented as follows:

$$\begin{aligned}
 &100110101 \\
 &\oplus 101010101 \\
 &\oplus \underline{011101101} \\
 &= 010001101
 \end{aligned}$$

The transmitted coded packet by node 4 that created by performing the operation XOR on packet 1, packet 2, and packet 3 is 010001101. During the transmissions in the third phase, neighboring nodes that receive such coded packet further perform the operation XOR on the received coded packet with the available original data packets at the receiving node that constitute the coded packet in order to get the missing original data packet. For example, node 1 performs the operation XOR on the received coded packet and packets 1 and 3 to get missing data packet 2 as shown below

$$\begin{array}{r}
010001101 \\
\oplus 100110101 \\
\oplus \underline{011101101} \\
= 101010101
\end{array}$$

The resulting bit sequence above is the data packet 2 and thus, packet 2 is received at node 1. Similarly, node 2 performs the operation XOR on the received coded packet and packets 2 and 3 to get missing data packet 1 as shown below

$$\begin{array}{r}
010001101 \\
\oplus 101010101 \\
\oplus \underline{011101101} \\
= 100110101
\end{array}$$

Similar, node 3 performs the operation XOR on the received coded packet and packets 1 and 2 to get missing data packet 3.

This chapter presents a new XOR based scheduling algorithm for NC in cooperative local repair for a group of nodes can hear each other in wireless star-oriented networks, as shown in figure 2.1. This chapter also presents the associated sequential MAC transmission technique that results from the new cooperative repair algorithm.

## 2.3 Scheduling Algorithm and MAC Procedure

The scheduling algorithm presented here was designed to benefit the maximum number of nodes. The algorithm first tries to combine the maximum number of packets that can aid the maximum number of nodes. The flow chart of the scheduling algorithm is shown in figure 2.2. Let us assume a sub network of  $N$  nodes that can hear each other directly (nodes within the small circuits in figure 2.1), each node  $s$  where ( $s = 1, 2, \dots, N$ ) has received a set of packets of the file transmitted by the BS. Obviously, the packets that are received by all  $N$  nodes will not be candidates for transmission in local repair and will be discarded from the repair window (point A on the flow chart). The window length  $L$  is thus updated by excluding the packets received by all  $N$  nodes. For example, from table 2-1, packet 7 will be excluded. Also, packets that are received by only one node in the sub network, such as packet 10 in table 2-1, will not be considered for NC combination. For those packets, single broadcast transmissions are more efficient since NC cannot provide any improvement (point B in the flow chart).

We denote  $\bar{d}_s$  as a vector of the received packets identities at node  $s$  in the sub network, excluding the packets received by all nodes and packets received by one node only. For instance, from table 2-1,  $\bar{d}_3 = [1 \ 2 \ 5 \ 8 \ 9]$ . Let  $(N - i)$  be the combination level where ( $i = 1, 2, \dots, N - 2$ ) (point F on the flow chart). If a node tries to combine (by XOR) the maximum number of packets, which is  $(N - 1)$  packets, this implies ( $i = 1$ ). We denote the number of nodes that will benefit from the coded packet as  $(N - j)$  nodes, where ( $j = 1, 2, \dots, N - 2$ ), (point E on the flow chart).

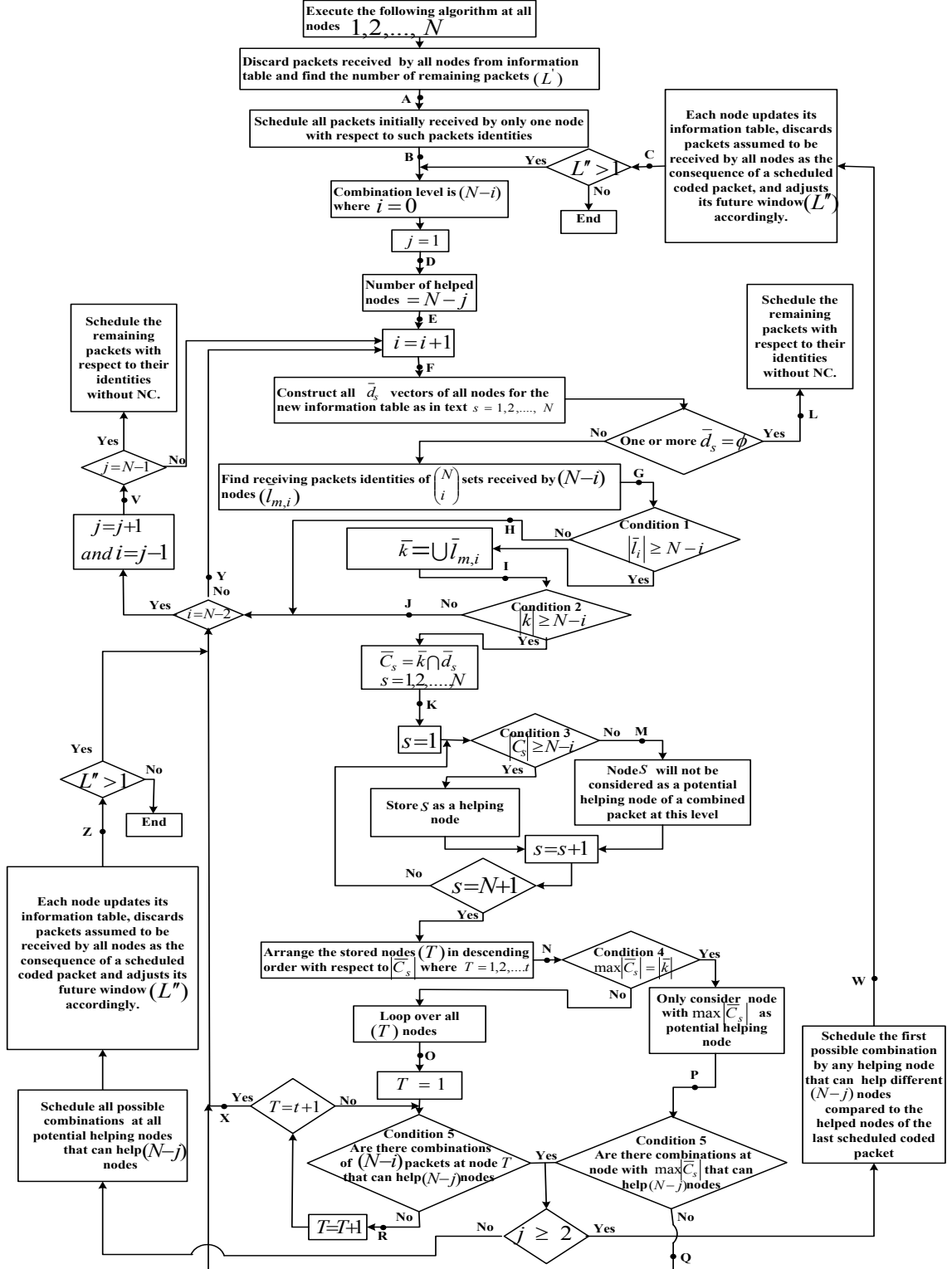


Figure 2.2 Flow chart of the scheduling algorithm

First, the scheduling algorithm tries to find combinations of packets that would favor the maximum number of nodes ( $j = 1$ ). The scheduling algorithm commences with the highest combination level ( $i = 1$ ) where one combined packet will help as many nodes as possible. If there is no combination at this level that could help  $(N - 1)$  nodes, the scheduling algorithm decrements the combination level by one ( $i$  becomes 2) and tries to find a combination of packets at any node that will help  $(N - 1)$  nodes, and so on for all ( $i \geq j$ ). Motivated by efficiency considerations, we maintain that  $(N - i) \leq (N - j)$ , i.e. the combination level is less than or equal to the number of helped nodes. To do the contrary would be a waste of bandwidth since a higher combination level should not be used to help fewer nodes. All possible combinations of packets that will help  $(N - 1)$  nodes at any combination level, by any node, will be scheduled for transmission and the repair window adjusted accordingly (point Z in the flow chart). If the scheduling algorithm schedules all the possible packet combinations at all the potential helping nodes, at all combination levels that can help  $(N - 1)$  nodes and there are still some packets that have not been scheduled (as evidenced by  $L'' > 1$  in the flow chart), the scheduling algorithm increments  $j$  to 2 (point V on the flow chart) and searches for combination that will help  $(N - 2)$  nodes commencing from the highest combination level. The highest combination level for  $j = 2$  is  $(N - 2)$  packets, since XORing  $(N - 1)$  packets to help  $(N - 2)$  nodes is inefficient in terms of process and bandwidth. The first combination that will help  $(N - 2)$  nodes, discovered at any combination level, will be scheduled for transmission and the repairing window adjusted accordingly (point C in the flow chart).

After scheduling a combination that can help  $(N - j)$  nodes ( $j \geq 2$ ), the algorithm assumes that all the  $(N - j)$  nodes have received such a combined packet, and the XOR operation decodes the packet and their  $\bar{d}_s$  are accordingly adjusted (point C in the flow chart). Then the scheduling algorithm tries to find other combinations that will help  $(N - 1)$  nodes, starting from the highest combination level. After each scheduling that helps  $(N - 2)$  or fewer nodes, the scheduling algorithm loops again to search for combinations that can help  $(N - 1)$  nodes (point D in the flow chart). If there is no combination that can help  $(N - 1)$  nodes after scheduling a coded packet that can help  $(N - 2)$  or fewer nodes, the algorithm will schedule the first combination that can help other  $(N - 2)$  or fewer nodes, compared to the nodes that were helped by the last scheduled coded packet (point W in the flow chart). If the scheduling algorithm schedules all the possible combinations that will help  $(N - 1)$  nodes and there is no possible combination that can help  $(N - 2)$  nodes, and there are still some packets that have not been scheduled, the algorithm searches for a combination that will help  $(N - 3)$  nodes, i.e.  $j = 3$ , starting from the highest combination level which is  $(N - 3)$  packets, i.e.  $i = 3$  in this case, and so on. A node that has been selected to create a combined packet formed from the XORing of  $(N - i)$  packets will include in this XOR operation only those packets commonly received by at least  $(N - i)$  nodes, because the number of helped nodes  $(N - j)$  is always greater than or equal to the combination level  $(N - i)$ . The main steps of the proposed scheduling algorithm can be presented as follows:

*Step1:* The algorithm first will find the involved packets to create the coded packets at the current combination level. We have  $\binom{N}{i}$  different possible sets of nodes  $(\bar{n}_{v,i})$  where the



$v^{\text{th}}$  set has  $(N - i)$  nodes where  $v = 1, 2, \dots, \binom{N}{i}$ . As an example, if  $i = 1$  in table 2-1, then we have  $\binom{4}{1} = 4$  sets where each set is composed of three nodes, as follows:

$$\bar{n}_{1,1} = \{\text{node 1, node 2, node 3}\},$$

$$\bar{n}_{2,1} = \{\text{node 1, node 2, node 4}\},$$

$$\bar{n}_{3,1} = \{\text{node 1, node 3, node 4}\},$$

$$\bar{n}_{4,1} = \{\text{node 2, node 3, node 4}\},$$

Then the algorithm finds the identities of the packets commonly received by all nodes in each set of nodes. Let  $\bar{l}_{v,i}$  denote the identities of the commonly packets received in each set. For instance, in table 2-1, for  $i = 1$ ,  $\bar{l}_{1,1} = \emptyset$ ,  $\bar{l}_{2,1} = \{3, 6\}$ ,  $\bar{l}_{3,1} = \{1, 9\}$ , and  $\bar{l}_{4,1} = \{2\}$ . Only these packets are involved to be XORed together to create the coded packets at this combination level which in turn minimizes the total transmission time of the subsequent repair process.

*Step 2:* Denote  $|\bar{l}_i|$  as the number of nonempty vectors of all the  $\bar{l}_{v,i}$  vectors. The algorithm checks if  $|\bar{l}_i|$  is less than  $(N - i)$ . If  $|\bar{l}_i| < (N - i)$ , then a packet combination at this level that helps  $(N - j)$  nodes is not possible, since the algorithm picks only one packet from each set to create a coded packet. The algorithm decrements the combination level by one:  $i = i + 1$ . Thus, the algorithm will not proceed at this combination level since it knows, from the value of  $|\bar{l}_i|$ , that it is not possible to create a coded packet at this combination level that can help  $(N - j)$  nodes. From table 2-1, for  $i = 1$ ,  $|\bar{l}_i|$  is three sets which is equal to  $(N - i)$ , and so the algorithm will proceed to try to create coded packets at this combination level.

*Step 3:* If  $|\bar{l}_i| \geq (N - i)$ , we denote  $\bar{k}_i$  as the union set of all the  $\bar{l}_{v,i}$  sets of packets identities.

$$\bar{k}_i = \bigcup_{v=1}^{\binom{N}{i}} \bar{l}_{v,i} \quad (2.1)$$

$\bar{k}_i$  is actually a vector of each packet's identity that will be considered at the  $(N - i)$  combination level for the information table at given  $i$ . For example, from table 2-1, for  $i = 1$ ,  $\bar{k}_i = \{1, 2, 3, 6, 9\}$ . Denote  $|\bar{k}_i|$  as the number of packets identities of vector  $\bar{k}_i$ . If  $|\bar{k}_i| < N - i$ , then it is not possible that a packet combination at this combination level can help  $(N - j)$  nodes. Therefore, the algorithm will decrement the combination level by one, to become  $(N - i - 1)$ , and try for the new combination level.

*Step 4:* If  $|\bar{k}_i| \geq N - i$ , then we do find the identities of packets received by each node  $s$  that intersect with  $\bar{k}_i$ .

$$\bar{c}_{s,i} = \bar{k}_i \cap \bar{d}_s \quad (2.2)$$

For example, from table 2-1, for  $i = 1$ ,  $\bar{c}_{1,1} = \{1, 3, 6, 9\}$ ,  $\bar{c}_{2,1} = \{2, 3, 6\}$ ,  $\bar{c}_{3,1} = \{1, 2, 9\}$ , and  $\bar{c}_{4,1} = \{1, 2, 3, 6, 9\}$ . Denote  $|\bar{c}_{s,i}|$  as the number of packets identities of vector  $\bar{c}_{s,i}$ . If  $|\bar{c}_{s,i}| < N - i$ , then the  $s^{th}$  node cannot combine  $(N - i)$  packets that would help  $(N - j)$  nodes. Nodes with  $|\bar{c}_{s,i}| \geq N - i$  are sorted in descending order with respect to  $|\bar{c}_{s,i}|$ . A node with the maximum  $|\bar{c}_{s,i}|$  will be considered first to create a coded packet then a node with lower  $|\bar{c}_{s,i}|$ , and so on. Therefore, the node with the maximum

$|\bar{C}_{s,i}|$  will have the highest chance to be scheduled for transmission, in order to help other nodes by its combined packet transmission.

For example, from table 2-1, for  $i = 1$ ,  $|\bar{C}_{1,1}| = 4$ ,  $|\bar{C}_{2,1}| = 3$ ,  $|\bar{C}_{3,1}| = 3$  and  $|\bar{C}_{4,1}| = 5$ . Thus, all nodes can create coded packets. Consequently, all nodes will be considered as possible helping nodes with respect to their  $|\bar{C}_{s,i}|$ , where the scheduling algorithm considers node 4 as a first possible helping node.

*Step 5:* For the given number of nodes to be helped ( $N - j$ ), and the possible combination level of ( $N - i$ ) packets, if  $\max(|\bar{C}_{s,i}|) = |\bar{k}_i|$ , we only consider the node with  $\max(|\bar{C}_{s,i}|)$  as a helping node and proceed to try to find combined packets at level ( $N - i$ ). If no ( $N - j$ ) nodes can benefit from the selected helping node, the algorithm will go to a lower combining level ( $N - i - 1$ ) without considering any other possible combined packets from any other node. The reasoning is that this node has all of the  $|\bar{k}_i|$  packets that can help, and so if the combined packets of such a node cannot help ( $N - j$ ) nodes, no other nodes will either. For example, from table 2-1, for  $i = 1$ ,  $\max(|\bar{C}_{s,i}|) = |\bar{C}_{4,1}| = 5 = |\bar{k}_1|$  and as a result, only node 4 will be considered as a helping node at this combination level. However, if the  $\max(|\bar{C}_{s,i}|) < |\bar{k}_i|$ , all nodes with  $|\bar{C}_{s,i}| \geq N - i$  will be considered as helping nodes. The algorithm investigates the possibility of forming combined packets from the node with the highest  $|\bar{C}_{s,i}|$  that can help ( $N - j$ ) nodes, and then it goes to the node with the next-highest  $|\bar{C}_{s,i}|$ . If, after investigating all the nodes with  $|\bar{C}_{s,i}| \geq N - i$  no combined packets from any node yields benefits to ( $N - j$ ) nodes, then the algorithm reverts to the next lower combination level.

*Step 6:* Let  $\bar{U}$  be the set of the identities of potentially transmitting helping nodes  $s$ , whose  $|\bar{C}_{s,i}| \geq N - i$  have been sorted in descending order with respect to the  $|\bar{C}_{s,i}|$  values. Denote the transmitting node under consideration to combine  $(N - i)$  packets as node  $x$ . Node  $x$  will pick  $(N - i)$  packets whose identities are part of  $\bar{C}_{x,i}$ . These packets are arranged in ascending order with respect to their identities, and the scheduling algorithm will check if these packets can help  $(N - j)$  nodes. For  $(j = 1)$ , if the combined packet can help  $(N - 1)$  nodes, this specific coded packet will be scheduled and the algorithm will pick other different  $(N - i)$  packets from the same helping node whose identities are part of  $\bar{C}_{x,i}$ , and so on. However, if the combined packet cannot help  $(N - 1)$  nodes, the algorithm will pick other  $(N - i)$  packets from the same helping node whose identities are part of  $\bar{C}_{x,i}$ , and so on. After scheduling all the possible coded packets from a certain helping node, which can help  $(N - 1)$  nodes, the algorithm selects another node with lower  $|\bar{C}_{s,i}|$  as a potential helping node provided that  $\max(|\bar{C}_{s,i}|) < |\bar{k}_i|$ . Packets that are part of  $\bar{C}_{x,i}$  where  $x$  is the selected helping node that have not been scheduled by any other considered helping node will be involved in creating a coded packet. If  $\max(|\bar{C}_{s,i}|) = |\bar{k}_i|$ , only a node with  $\max(|\bar{C}_{s,i}|)$  will be considered as a potential helping node. After scheduling all the possible coded packets at all the potential helping nodes that can help  $(N - 1)$  nodes, the algorithm updates the information table and the combination level will be decremented by one. On the other hand, if  $(j \geq 2)$  and the combined packet can help a group of  $(N - j)$  nodes this specific coded packet is scheduled. If not, the algorithm tries to find another group of  $(N - j)$  nodes that can be helped by this specific coded packet. Finally, if this specific coded packet cannot help any group of  $(N - j)$  nodes, the algorithm picks another possible combined packet from

the same helping node, and so on. If none of the combined packets of the particular helping node can benefit any group of  $(N - j)$  nodes, the algorithm selects another node with lower  $|\bar{C}_{s,i}|$  as a potential helping node provided that  $\max(|\bar{C}_{s,i}|) < |\bar{k}_i|$ . If  $\max(|\bar{C}_{s,i}|) = |\bar{k}_i|$  and none of the combined packets of the node with  $\max(|\bar{C}_{s,i}|)$  can benefit any group of  $(N - j)$  nodes, the combination level will be decremented by one and the algorithm will try to create a coded packet for this new combination level.

*Step 7:* To check if node  $x$  will be able to help  $(N - j)$  nodes at the  $(N - i)$  combination level, the algorithm will compare the  $(N - i)$  chosen packets of  $\bar{C}_{x,i}$  with all the  $\bar{C}_{s,i}$  packets. At each node, we have to find the availability of the constituting packets of each coded packet. If the availability number is  $(N - i - 1)$ , this node can decode the encoded packet and find its missing packet. We also have to find the identities of the unavailable packets at each node corresponding to a certain potential combined packet. If the total number of unavailable packets at all  $(N - j)$  potentially helped nodes corresponding to a certain coded packet is  $(N - i)$ , then this coded packet can help all the  $(N - j)$  helped nodes. However, for better and faster processing we propose the following equivalent and more efficient implementation. Denote the candidate  $(N - i)$  packets to be combined at node  $x$  as  $p_{x,e}$  where  $e = 1, 2, \dots, N - i$ . To make sure that these  $(N - i)$  packets combined by node  $x$  will help  $(N - j)$  nodes, the following three conditions must be satisfied for the  $(N - j)$  nodes to be helped:

$$\sum_{s=1, s \neq x}^{N-j} p_{I(s),1} \times p_{I(s),2} \dots \times p_{I(s),N-i} = 0 \quad (2.3)$$

where  $I(s), s = 1, 2, \dots, N - j$  denotes the identities of the nodes potentially helped by this combined packet. The number of such nodes to be helped is  $(N - j)$ . This condition guaranties that each node is missing at least one packet from the packets constituting the current subject combined packet.

For example, using the information in table 2-2, suppose that node 5 is trying to combine packets (1, 2, 3, and 4) to help four nodes (1, 2, 3, and 4). By applying (2.3)

$$\sum_{s=1, s \neq x}^4 p_{I(s),1} \times p_{I(s),2} \times \dots \times p_{I(s),N-i} = 0 + 0 + 0 + 0 \implies 0$$

Thus, the first condition is satisfied and the algorithm will apply the second condition.

Table 2-2 Information table at each node describing the received and missed packets of all users

Packet identity	1	2	3	4
Nodes identities				
1	1	0	0	1
2	1	1	0	1
3	1	1	1	0
4	0	1	1	1
5	1	1	1	1
6	1	0	0	1

The second condition is:

$$\sum_{s=1, s \neq x}^{N-j} \sum_{e=1}^{N-i} p_{I(s),e} = (N - j) \times (N - i - 1) \quad (2.4)$$

Equation (2.4) effectively says that each of the potentially helped  $(N - j)$  nodes has already received all but one of the constituent packets of the subject packet, i.e. they have received  $(N - i - 1)$  packets if (2.3) is satisfied. The total number of corresponding indicators in the table for all of the  $(N - j)$  potentially helped nodes would be  $(N - j) \times (N - i - 1)$ , as per (2.4). The result of applying this condition at node 5, that is trying to combine four packets to help four nodes, is

$$\sum_{s=1, s \neq x}^4 \sum_{e=1}^4 p_{I(s),e} = 3 + 3 + 2 + 3 \Rightarrow 11 \Rightarrow \neq (N - j) \times (N - i - 1) = 4 \times 3 \Rightarrow$$

$$= 12$$

Thus, this combined packet will not be scheduled for transmission since only three nodes can benefit from this combined packet (node 2, 3 and 4). However, if node 1 in table 2-2 has received packet 2 or 3, as shown in table 2-3, then the second condition is satisfied and the algorithm will apply the third condition.

The third condition is:

$$\sum_{s=1, s \neq x}^{N-j} p_{I(s),e} < N - j \quad \text{for all } e \quad (2.5)$$

Equation (2.5) guarantees that no packet of the constituting  $(N - i)$  packets has been received by all of the potentially helped  $(N - j)$  nodes. For example, using table 2-3, apply the third condition at node 5.

$$\sum_{s=1, s \neq x}^4 p_{I(s),1} = 3 < N - j, \quad \sum_{s=1, s \neq x}^4 p_{I(s),2} = 4 = N - j,$$

$$\sum_{s=1, s \neq x}^4 p_{I(s),3} = 2 < N - j, \quad \sum_{s=1, s \neq x}^4 p_{I(s),4} = 3 < N - j$$

Table 2-3 Information table at each node describing the received and missed packets of all users

Packet identity	1	2	3	4
Nodes identities				
1	1	1	0	1
2	1	1	0	1
3	1	1	1	0
4	0	1	1	1
5	1	1	1	1
6	1	0	0	1

Hence, this combined packet will not be scheduled for transmission since only three packets will be received at all four helped nodes and the algorithm will combine packets 1, 3, and 4 instead. But, if node 1 in table 2-3 has received packet 3 and is missing packet 2, then all the conditions are satisfied and the algorithm will schedule this combined packet.

For a specific group of  $(N - j)$  nodes to be helped the computed values of  $|\bar{C}_{s,i}|$ ,  $s = 1, 2, \dots, N$  may have two or more equal values for different  $s$ . The scheduling algorithm arranges the stored nodes that have  $|\bar{C}_{s,i}| \geq (N - i)$  in descending order with respect to values of  $|\bar{C}_{s,i}|$ . Whenever it faces more than one node with the same  $|\bar{C}_{s,i}|$ , it first picks the node with the lower node identity value as the potential helping node.



Finally, in the flow chart, it may happen that a node misses all of the packets after some scheduling, and so NC will not help with these leftover packets and single broadcast transmission should be used.

## 2.4 Performance Analysis of the Proposed Algorithm

In this section, an upper bound on the improvement factor and analysis of the proposed algorithm are introduced. We assume that packets within a file are independently received at each node. Also as we have mentioned, the channels between BS and nodes are erroneous and have the same quality. On the other hand, transmission channels between nodes are error free.

For a certain number of nodes  $N$  and given probability of packet success  $P_c$  and file size  $L$ , the probability of  $M$  nodes having received a particular packet is

$$P_M = \binom{N}{M} P_c^M (1 - P_c)^{N-M} \quad (2.6)$$

Where  $M = 0, 1, 2, \dots, N$

For convenience, we start with the upper bound on the improvement factor of the proposed scheduling algorithm.

### 2.4.1 The upper bound performance

For analysis convenience each packet is assumed to have been received at least by one node. To guarantee this, the BS will retransmit packets that were not received by any node from previous transmissions until it receives at least one acknowledgment for each packet. Such acknowledgments and repeated BS transmissions details are not discussed further here. For a certain packet, the probability that  $M$  nodes will receive it after the first transmission is  $P_M$ . The probability that no node will receive it after the first transmission is  $P_0$ . If no node has received the packet after the first transmission, the BS will retransmit it and the probability that  $M$  nodes will receive it after the second transmission is  $P_0 \times P_M$ . If no node has received the packet after the second transmission, the BS will retransmit it and the probability that  $M$  nodes will receive it after the third transmission is  $P_0^2 \times P_M$ , and so on. The probability of a packet eventually being received by all  $M$  nodes is:

$$\begin{aligned}
 P'_M &= P_M + P_0 \times P_M + P_0^2 \times P_M + P_0^3 \times P_M + \dots \\
 &= P_M(1 + P_0 + P_0^2 + P_0^3 + \dots) \\
 P'_M &= \frac{P_M}{(1 - P_0)} \\
 P'_M &= \frac{P_M}{1 - (1 - P_c)^N} \quad \text{for } M = 1, 2, \dots, N
 \end{aligned} \tag{2.7}$$

where  $P_0$  is the probability that a certain packet was not received by any node which can be computed using (2.6). For a certain packet, the average number of nodes ( $R$ ) that will receive the packet can be found as follows:

$$R = P'_1 + 2 \times P'_2 + 3 \times P'_3 + \dots + N \times P'_N \quad (2.8)$$

Thus, the average number of received packets ( $k$ ), (the total number of ones in the information table, for example from table 2-1,  $k = 26$  packets) can be found as follows:

$$k = L \times R \quad (2.9)$$

For a given  $P_c$ , it is possible that some packets of the file are received by all nodes with a certain probability. Packets that are received by all nodes will not be considered for the local repair process. So, we find the distribution of the window size utilized for local repair ( $L'$ ) for a given  $P_c$  as follows:

$$P(L' = D) = \binom{L}{D} (1 - P'_N)^D P_N^{(L-D)} \quad (2.10)$$

In (2.10),  $P'_N$  is the probability that a packet is received by all  $N$  nodes. Let  $T$  denote the required number of transmissions for a given  $L'$ . The maximum number of transmissions ( $T_{max}$ ) is  $L'$ , which represents the case where no NC is possible. This scenario occurs with certain probability, depending on the values of  $P_c$ ,  $L$  and  $L'$ . Another possible scenario is ( $T = L' - 1$ ), which also occurs with a certain probability, and so on. In the best scenario, the algorithm will be able to create coded packets for the whole repair window. Each coded packet that is created will be able to help  $(N - 1)$  nodes. This scenario presents the minimum required number of transmissions. The minimum number of transmissions is:

$$T_{min} = \left\lceil \frac{N \times L - k}{N - 1} \right\rceil \quad (2.11)$$

The improvement factor is defined as the ratio between the required number of transmissions without NC ( $N + L'$ ) and the required number of transmissions using NC ( $N + T$ ). In order to compute the upper bound on the improvement factor ( $\Psi_{up}$ ), we assume that the occurrence of the minimum number of transmissions is certain. Thus, the upper bound of the improvement factor is

$$\Psi_{up} = \sum_{L'=1}^L \frac{(N + L')}{(N + T_{min})} \times P(L') \quad (2.12)$$

### 2.4.2 Analysis

In the first step of this analysis, we exclude packets that have been received by all nodes and packets not received by any node. By applying (2.6) where  $M = 0$  and  $N$ , the average number of packets that will be excluded is:

$$\text{excluded packets} = [L \times (P_0 + P_N)]$$

$$L' = L - \text{excluded packets} \quad (2.13)$$

$[q]$  means that we take the integer number closest to  $q$  and  $L'$  is the effective window size. Let  $(B_{N-i,j,h})$  presents the average number of possible coded packets where the subscripts  $(N - i, j, h)$  present the combination level, the number of nodes that will not

benefit from the coded packet including the transmitted node and the number of scheduled packets for  $j \geq 2$ , respectively.

The probability that a particular packet is received by one node ( $M = 1$ ) is

$$P_1 = \binom{N}{1} P_c (1 - P_c)^{N-1}$$

The average number of such packets ( $A_M$ ) where  $M = 1$  is

$$A_1 = [L \times P_1]$$

Since these packets will be transmitted individually, the average number of transmissions of such packets is

$$B_{1,1,0} = A_1$$

All packets constituting each of  $B_{N-i,1,h}$  coded packets are assumed to be scheduled and will be received by all nodes. Thus, all packets constituting any of  $B_{N-i,1,h}$  will be discarded from the repair window. After scheduling all packets received by one node, the algorithm tries to combine the maximum number of packets ( $N - 1$ ) packets ( $i = 1$ ) that can help ( $N - 1$ ) nodes ( $j = 1$ ). Each packet to be combined at this level should exist at ( $N - 1$ ) nodes. By using (2.6) where  $M = N - 1$ , we calculate the probability that a packet has been received by ( $N - 1$ ) nodes. Then we find the average number of such packets ( $A_{N-1}$ )

$$A_{N-1} = [L \times P_{N-1}]$$

The average number of possible coded packets at this combination level ( $B_{N-1,1,0}$ ), each constituted of  $(N - 1)$  individual packets, is

$$B_{N-1,1,0} = \left\lfloor \frac{A_{(N-1)}}{(N-1)} \right\rfloor$$

Where  $\lfloor q \rfloor$  means that we take only the integer number of  $q$ . The remaining nonscheduled packets ( $R_{N-1,i,h}$ ) received by  $(N - 1)$  nodes can be found as the remainder of the division above, i.e.

$$R_{N-1,1,0} = A_{N-1} - B_{N-1,1,0} \times (N - 1) \quad (2.14)$$

Next, we update the number of packets to be repaired by removing packets that have been scheduled. Then we find the average number of packets that can be combined in the next lower combination level, i.e.  $(N - 2)$  packets that can help  $(N - 1)$  nodes. Only the nonscheduled packets that are received by  $(N - 1)$  nodes, i.e. ( $R_{N-1,1,0}$ ) and packets received by  $(N - 2)$  nodes will be considered. The average number of packets received by  $(N - 2)$  nodes is

$$A_{N-2} = [L \times P_{N-2}] \quad (2.15)$$

In order to combine  $(N - 2)$  packets that can help  $(N - 1)$  nodes we need to have  $(N - 3)$  packets received by  $(N - 1)$  nodes and 1 packet received by  $(N - 2)$  nodes as a single combination. Table 2-4 shows a possible scenario corresponding to this case. In table 2-4,  $N = 6$  nodes and node 1 tries to combine  $(N - 2)$  packets (4 packets) that can help  $(N - 1)$  nodes (5 nodes). For this combination to help 5 nodes; each helped node must have initially received 3 packets of these combined 4 packets. Thus, the summation

of the number of packets missed by all helped nodes is equal to  $(N - 1)$ . So, the only possible combination of 4 packets that can help 5 nodes (for  $N = 6$ ) is to combine 3 packets received by 5 nodes and 1 packet received by 4 nodes.

Table 2-4 A possible scenario to combine  $(N-2)$  packets that can help  $(N-1)$  nodes

Nodes and packets identities	1	2	3	4
Node 1	1	1	1	1
Node 2	1	1	0	1
Node 3	1	1	1	0
Node 4	1	0	1	1
Node 5	0	1	1	1
Node 6	1	0	1	1

The number of possible coded packets at  $(N - 2)$  combination level is

$$B_{N-2,1,0} = \begin{cases} \left\lfloor \frac{R_{N-1,1,0}}{N-3} \right\rfloor & \text{if } \left\lfloor \frac{R_{N-1,1,0}}{N-3} \right\rfloor \leq A_{N-2} \\ A_{N-2} & \text{if } \left\lfloor \frac{R_{N-1,1,0}}{N-3} \right\rfloor \geq A_{N-2} \end{cases} \quad (2.16)$$

The number of nonscheduled packets that are received by  $(N - 1)$  nodes after scheduling all combinations of  $(N - 2)$  packets is  $(R_{N-1,2,0} = R_{N-1,1,0} - B_{N-2,1,0}(N - 3))$  packets. Similarly, the number of nonscheduled packets that are received by  $(N - 2)$  nodes after scheduling all possible combinations of  $(N - 2)$  packets is  $(R_{N-2,2,0} = A_{N-2} - B_{N-2,1,0})$  packets. Next, we update the number of packets to be repaired by removing packets that have been scheduled, i.e.  $B_{N-2,1,0}$ . Similarly, we find the average number of packets for the next combination level  $(N - 3)$  packets that can help  $(N - 1)$  nodes. Only nonscheduled packets that are received by  $(N - 1)$  and  $(N - 2)$  nodes, i.e.

$R_{N-1,2,0}$  and  $R_{N-2,2,0}$  and packets received by  $(N - 3)$  nodes will be under consideration for this combination level. The average number of packets received by  $(N - 3)$  nodes is

$$A_{N-3} = [L \times P_{N-3}] \quad (2.17)$$

In order to combine  $(N - 3)$  packets that can help  $(N - 1)$  nodes we need to have  $(N - 4)$  packets received by  $(N - 1)$  nodes and 1 packet received by  $(N - 3)$  nodes (table 2-5 shows a possible scenario corresponding to this case). Or  $(N - 5)$  packets received by  $(N - 1)$  nodes and 2 packets received by  $(N - 2)$  nodes as a single combination.

Table 2-5 A possible scenario to combine  $(N-3)$  packets that can help  $(N-1)$  nodes

Nodes and packets identities	1	2	3
Node 1	1	0	1
Node 2	1	0	1
Node 3	1	1	0
Node 4	1	1	1
Node 5	0	1	1
Node 6	1	0	1

In table 2-5,  $N = 6$  nodes; node 4 tries to combine  $(N - 3)$  packets (3 packets) that can help  $(N - 1)$  nodes (5 nodes). For this combination to help 5 nodes; each helped node must have initially received 2 packets of these combined 3 packets. Thus, the summation of the number of packets missed by all helped nodes is equal to  $(N - 1)$ . So, one possible combination of 3 packets that can help 5 nodes is to combine 2 packets  $((N - 4)$  packets)



received by 5 nodes ( $(N - 1)$  nodes) and 1 packet received by 3 nodes ( $(N - 3)$  nodes). Similarly, for the case of combining  $(N - 5)$  packets received by  $(N - 1)$  nodes and 2 packets received by  $(N - 2)$  nodes can be explained using a corresponding table latter case. The number of possible coded packets at  $(N - 3)$  combination level is:

$$B_{N-3,1,0} = \begin{cases} A_{N-3} + \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor & \text{if } R_{N-1,2,0} \geq \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor (N-5) + A_{N-3}(N-4) \\ A_{N-3} + \lfloor Z_1 \rfloor & \text{if } 0 \leq \lfloor Z_1 \rfloor \leq \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor \text{ and } A_{N-3} < \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor \\ \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor + \lfloor Z_2 \rfloor & \text{if } 0 \leq \lfloor Z_2 \rfloor \leq A_{N-3} \text{ and } \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor < \left\lfloor \frac{R_{N-1,2,0}}{(N-5)} \right\rfloor \\ \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor + \lfloor Z_3 \rfloor & \text{if } A_{N-3} > \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor \text{ and } 0 \leq \lfloor Z_3 \rfloor \leq \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor \\ \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor + \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor & \text{if } A_{N-3} > \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor \text{ and } \lfloor Z_3 \rfloor \geq \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor \\ \left\lfloor \frac{R_{N-1,2,0}}{(N-5)} \right\rfloor & \text{if } \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor > \left\lfloor \frac{R_{N-1,2,0}}{(N-5)} \right\rfloor \end{cases} \quad (2.18)$$

$$\text{Where, } Z_1 = \frac{R_{N-1,2,0} - (N-4) \times A_{N-3}}{(N-5)}, Z_2 = \frac{R_{N-1,2,0} - (N-5) \times \left\lfloor \frac{R_{N-2,2,0}}{2} \right\rfloor}{(N-4)}$$

$$\text{and } Z_3 = \frac{R_{N-1,2,0} - (N-4) \times \left\lfloor \frac{R_{N-1,2,0}}{(N-4)} \right\rfloor}{(N-5)}. \text{ Next, we update the number of packets to be repaired}$$

by removing packets that have been scheduled, i.e.  $B_{N-3,1,0}$  from the repairing window and the algorithm continues to find all other possible coded packets for all combination levels that can help  $(N - 1)$  nodes, i.e.  $(N - 4)$ , and so on. The interested reader may continue to help  $(N - 1)$  nodes at further combination levels  $N - 4, N - 5$ , etc. For analysis convenience we take small value of  $N$  and we stopped at combination level  $(N - 3)$  as above. Now, after scheduling all possible combinations of all combination levels that can help  $(N - 1)$  nodes and we still have some packets not yet scheduled, the algorithm will try to help  $(N - 2)$  nodes via combined packets each constituting of

$(N - 2)$  packets. The remaining nonscheduled packets that are received by  $(N - 1)$  and  $(N - 2)$  nodes after scheduling all possible combinations of all combination levels that can help  $(N - 1)$  nodes ( $R_{N-1,N-2,0}, R_{N-2,N-2,0}$ ) will be involved for such combination. Table 2-6 shows a possible scenario corresponding to this case. In table 2-6,  $N = 6$  nodes; node 5 tries to combine  $(N - 2)$  packets (4 packets) that can help  $(N - 2)$  nodes (4 nodes). For this combination to help 4 nodes; each helped node must have initially received 3 packets of these combined 4 packets.

Table 2-6 A possible scenario of combining  $(N-2)$  packets that can help  $(N-2)$  nodes

Nodes and packets identities	1	2	3	4
Node 1	0	1	1	1
Node 2	1	0	0	0
Node 3	1	1	1	0
Node 4	1	0	1	1
Node 5	1	1	1	1
Node 6	1	1	0	1

Thus, the summation of the number of packets missed by  $(N - 2)$  helped nodes is equal to  $(N - 2)$ . So, one possible combination of 4 packets that can help 4 nodes out of six nodes is to combine 3 packets received by  $(N - 2)$  nodes and 1 packet received by  $(N - 1)$  nodes. To be able to combine  $(N - 2)$  packets that can help  $(N - 2)$  nodes

$$R_{N-1,N-2,0} + R_{N-2,N-2,0} \geq (N - 2) \quad (2.19)$$

The average number of possible coded packets at a  $(N - 2)$  combination level that can help  $(N - 2)$  nodes is:

$$B_{N-2,2,1} = \left\lfloor \frac{Z_4}{N-2} \right\rfloor \quad \text{if } Z_4 \geq (N-2) \quad (2.20)$$

Where  $Z_4 = R_{N-1,N-2,0} + R_{N-2,N-2,0}$ . We note  $h = l$  meaning now the algorithm is trying to help  $(N-2)$  or fewer nodes for the first time. The algorithm will schedule only one such combination and then will search for combinations that can help  $(N-1)$  nodes and so on. If  $Z_4 < (N-2)$ , the algorithm will try to find combination of  $(N-3)$  packets that can help  $(N-2)$  nodes and so on. After scheduling a coded packet that can help  $(N-2)$  nodes, the constituting packets of the coded packet that are initially received by  $(N-1)$  nodes will be discarded from the repairing window and the constituting packets of the coded packet that were initially received by  $(N-2)$  nodes will be considered as packets received by  $(N-1)$  nodes. In general, after scheduling a coded packet that can help  $(N-Q)$  nodes where  $Q = 2, 3, 4, \dots, N-2$ , the constituting packets of the coded packet that are initially received or assumed to be received by less than  $(N-Q)$  nodes will be considered as packets received by  $(N-Q+1)$  nodes. Also, the constituting packets of the coded packet that are initially/assumed received by  $(N-U)$  nodes where  $(U = 1, 2, 3, \dots, Q)$  will be considered as packets received by  $(N-U+1)$  nodes.

After scheduling all missing packets and making sure that all nodes will be able to get all their missing packets then the scheduling process is done and nodes that have been scheduled for transmissions will transmit. The total number of transmissions ( $G$ ) is the summation of the receptions' report transmissions ( $N$ ) and total number of scheduled coded packets is

$$G = N + B_{1,1,0} + \sum_{h=0}^{L'} \left[ \left[ \sum_{i=1}^{N-2} B_{N-i,1,h} \right] + D' \right] \quad (2.21)$$

$$\text{Where } D' = \begin{cases} 1 & \text{if any } B_{N-i,j,h+1} \neq 0 \text{ for } j \geq 2 \text{ and } N-2 \geq i \geq j \\ 0 & \text{if all } B_{N-i,j,h+1} = 0 \text{ for } j \geq 2 \text{ and } N-2 \geq i \geq j \end{cases}$$

$G$  and  $D'$  values can be explained as follow. First the algorithm will sum  $N$  transmissions for receptions reports,  $B_{1,1,0}$  transmissions for packets received by only one node and all possible coded packets that can help  $(N-1)$  nodes for all combination levels for  $(h=0)$ . Then if any  $B_{N-i,j,h+1} \neq 0$  for  $j \geq 2, h=0$  and  $N-2 \geq i \geq j$ , the algorithm will schedule only one coded packet ( $D'=1$ ) that will help  $(N-2)$  nodes or less. Next, the algorithm will schedule all possible coded packets that can help  $(N-1)$  nodes ( $B_{N-i,1,1}$ ) for  $1 \leq i \leq N-2$ . After that, if any  $B_{N-i,j,h+1} \neq 0$  for  $j \geq 2, h=1$  and  $N-2 \geq i \geq j$ , the algorithm will schedule only one coded packet ( $D'=1$ ) that will help  $(N-2)$  nodes or less. Subsequently, the algorithm will schedule all possible coded packets that can help  $(N-1)$  nodes ( $B_{N-i,1,2}$ ) for  $1 \leq i \leq N-2$ , and so on. However, if all  $B_{N-i,j,h+1} = 0$  for  $j \geq 2, N-2 \geq i \geq j$  and  $h \geq 0$ , the algorithm will not schedule any coded packet that will help  $(N-2)$  nodes or less ( $D'=0$ ) for that  $h$ .

The improvement factor ( $\Psi$ ) is defined as the ratio between the required number of transmissions without NC ( $N+L'$ ) and the required number of transmissions using NC  $G$ . Thus, the improvement factor can be presented as follows:

$$\Psi = \frac{N+L'}{G} \quad (2.22)$$

## 2.5 Results

In this section we show the improvement factor provided by the proposed XOR based scheduling algorithm for NC in cooperative local repair in star network, through computer simulations, analysis, and upper bound on the improvement factor. We assume that the channels between nodes are error free (nodes are very close to each other). However, the channels between the nodes and the BS are error prone, which leads to the need for repairs. In addition, we assume that the channels between the BS and the nodes are of the same quality which is a reasonable assumption since the nodes are close to each other. During simulation, the improvement factor of the proposed scheduling algorithm (case 1) is compared with two cases where no NC is applied. In case 2, nodes cooperate together, i.e. exchange initial packets availability to create an information table at each node. Since no NC is used, each missing packet will be transmitted individually by one user. Thus, the total number of transmissions is  $(L' + N)$ , where  $L'$  is the file size after removing the packets received by all nodes, and  $N$  is the number of packets necessary for transmitting packets' availability by all nodes. The latter assumes that one packet is enough to convey the IDs of the packets missing at each node and a node's identity. In case 3, nodes do not cooperate with each other and each node will transmit all its received packets (no packets availability exchange, no coordination and no NC). For our scheduling approach (case 1) we take different  $N, L$  and  $P_c$ 's, and for each set of values we compute the improvement factor, processing time, etc. For case 1, we count the total number of transmissions until all users are able to find all of the file packets.

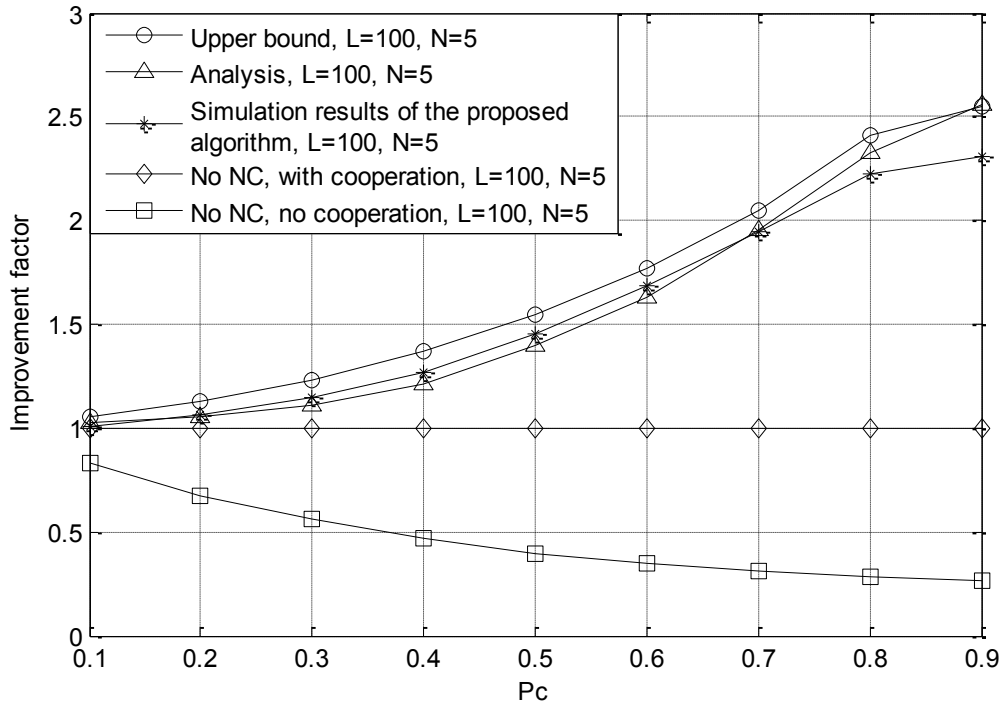


Figure 2.3 the improvement factors of: the proposed scheduling algorithm, with no NC and cooperation between nodes, with no NC and no cooperation between nodes, the upper bound, and the analysis.

To this number, we add the  $N$  packets necessary for the reception report transmissions, to get the total number of transmissions. Uniform random variables are used to determine the IDs of the packets initially received from the BS at every node. The number of packets initially received from the BS at each node is obtained by calling a binomially distributed random variable for a given file size and a given probability of packet success. Figure 2.3 shows the simulation, the upper bound and the analysis results of the proposed algorithm including case 2 and case 3. From figure 2.3, the new scheduling algorithm provides the best improvement factor compared to non-NC systems (cases 2 and 3). We also notice that the simulation and the analysis results are comparable.

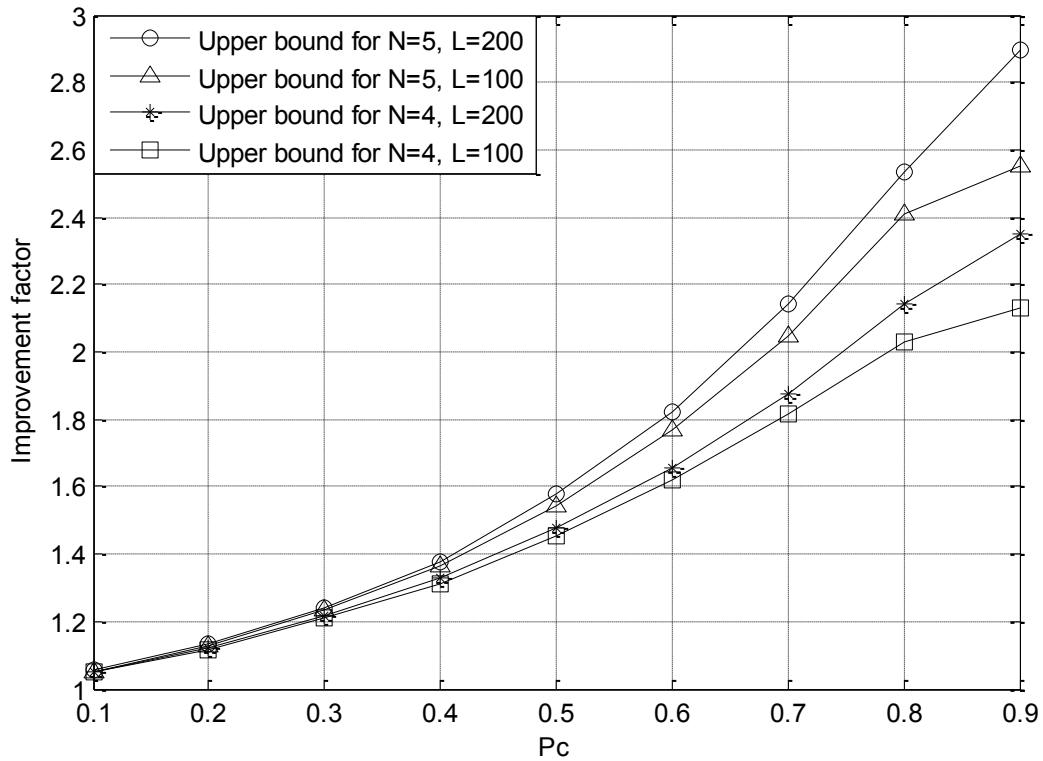


Figure 2.4 The upper bound on the improvement factors

From figure 2.3 we can observe that the improvement factor of the new scheduling algorithm is close to that for the upper bound. This is because the algorithm prioritizes the nodes with high packet availability to create coded packets that achieve maximum benefit. Figure 2.3 also shows that the higher the packet's availability, the better the improvement factor, since the number of packets received by  $(N - 1)$  nodes increases when the packet availability increases.

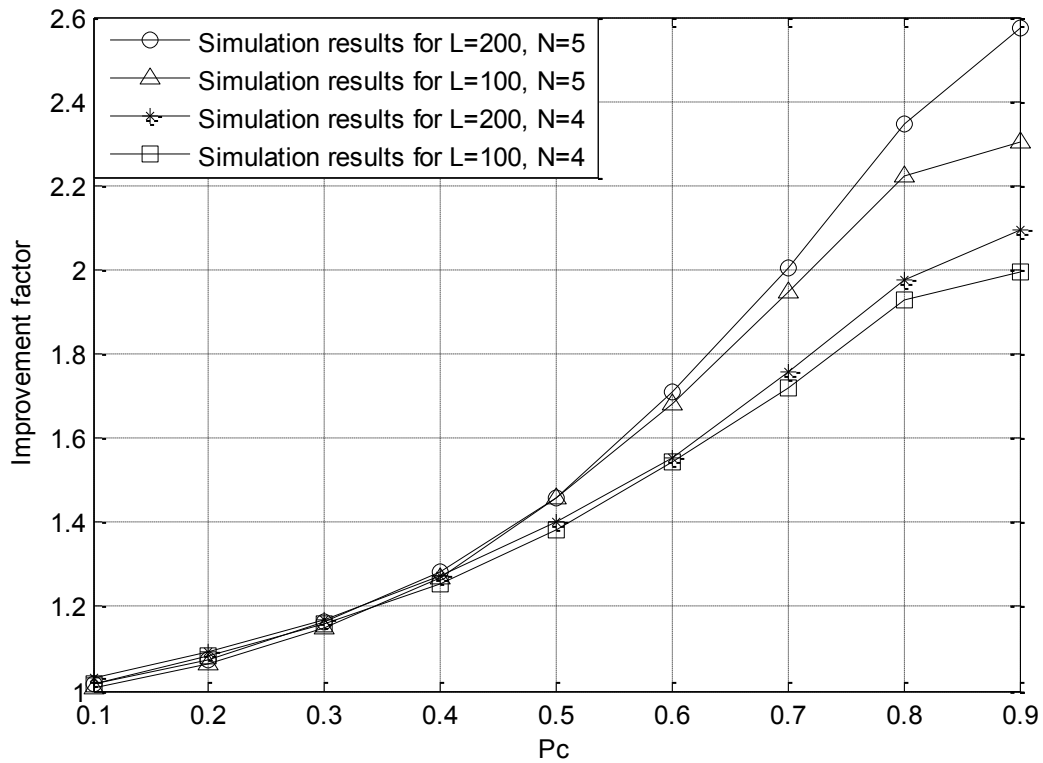


Figure 2.5 Simulation results

Figures 2.4 and 2.5 show the tradeoff between the number of nodes, the file size and the improvement factor. For ( $P_c > 0.4$ ), as the file size or the number of nodes increases, the improvement factor increases. However, increasing the number of nodes provides a better improvement factor than increasing the file size. This is because as the number of nodes increases, the maximum combination level will increase. Therefore, the maximum number of packets that could be combined together increases. On the other hand, increasing the file size will lead to increasing the likelihood of having many coded packets, which will have a smaller effect as the file size becomes larger. Also from figures 2.4 and 2.5, we can observe that the effects of the file size and number of nodes



for low  $P_c$  ( $P_c < 0.4$ ) are negligible. The reason is the small number of possible generated coded packets for low  $P_c$ . The proposed scheduling algorithm always tries to find the best node that can XOR a set of packets that will help maximum number of nodes. for that reason, the performance of the proposed scheduling algorithm is close to the upper bound which can be seen in figures 2.4 and 2.5.

The processing delay introduced by the scheduling process is an important issue to consider, since it determines the time required for local repair processes. In practice, cooperative local repair needs to be done in a certain amount of time depending on the delay tolerance of the application.

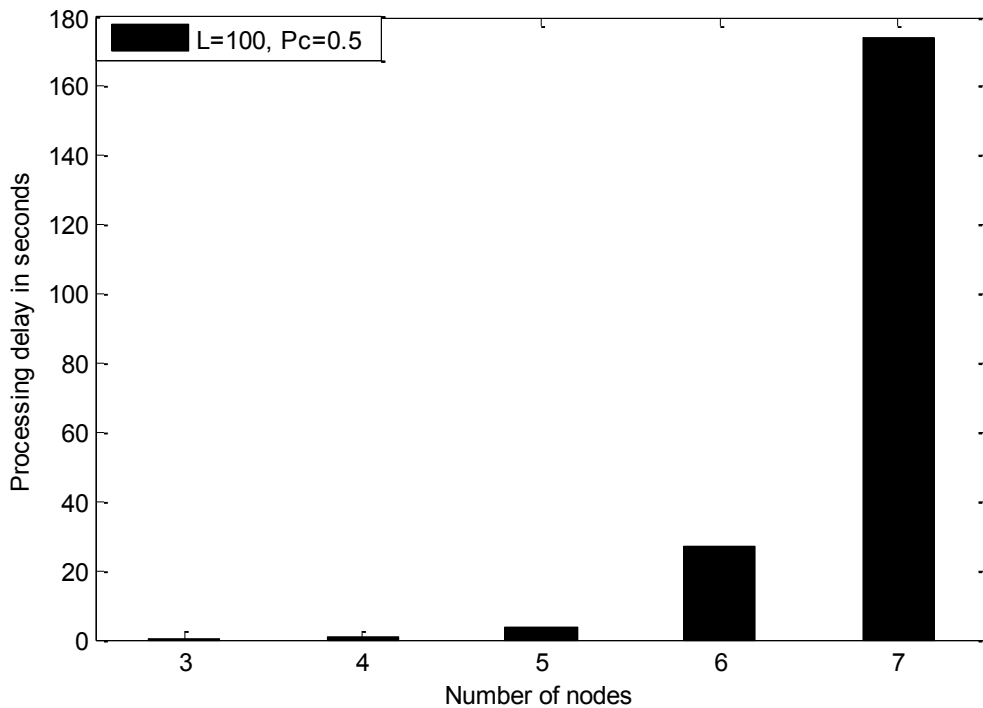


Figure 2.6 Processing delay (in seconds) vs. number of nodes in the new three-phase algorithm

The processing time was found by employing certain real time simulation indicators on a general purpose IBM PC (1.8 GH Intel(R) Core(TM)2 processor and 1 GB RAM ). This is a worst case indicator since in a real environment all the general functionality and processing related to performance measurement and curving would not be executed. In this subsection, we study the tradeoff between processing delay and number of nodes at certain file size and packet availability. Figure 2.6 shows that the lower the number of nodes, the lower the processing delay. The difference in processing delay for different number of nodes is not linear. For example, the processing delays in the case of  $N = 3, 4$  and 5 are almost the same, however it increases dramatically for  $N = 7$  for these given parameters (file size and packet's availability).

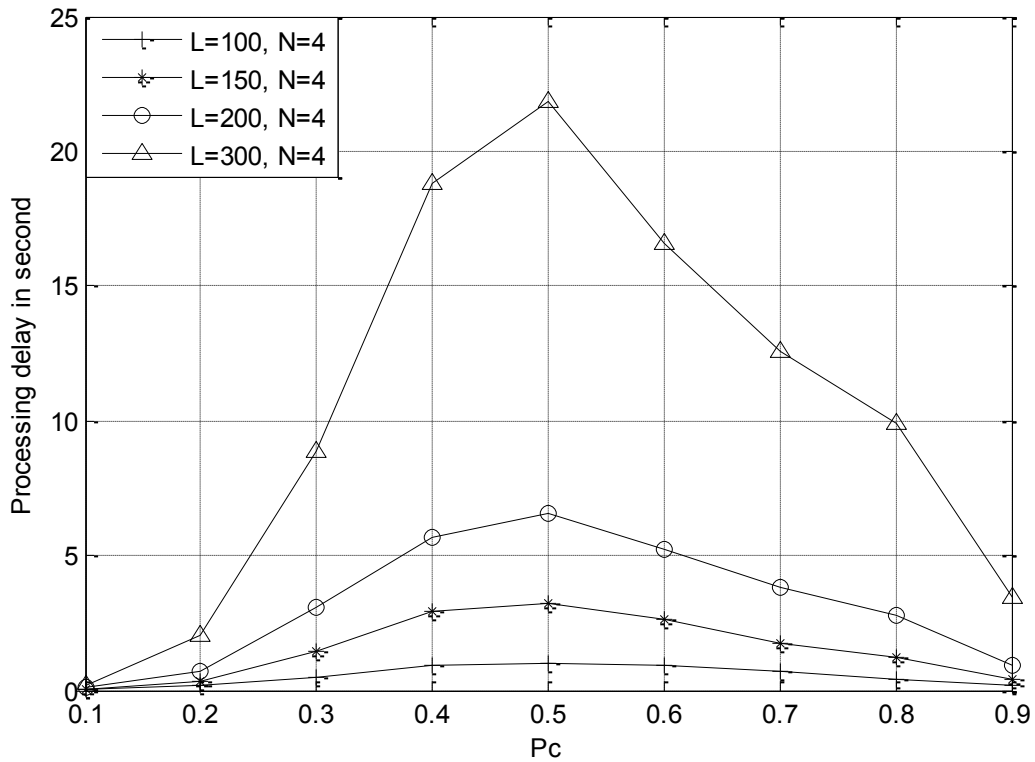


Figure 2.7 Processing delay (in seconds) vs. file size in the new three-phase algorithm

Tradeoffs between processing delay, packet availability and file size are shown in figure 2.7. Figure 2.7 shows that, as  $P_c$  increases, the processing delay increases up to a certain value of  $P_c$ . In addition to scheduling coded packets that will help  $(N - 2)$  or fewer nodes for  $(0.1 \leq P_c \leq 0.5)$ , as  $P_c$  increases the chance of coded packets that can help  $(N - 1)$  nodes at different combination levels will increase, which leads to increased processing delay. For high  $P_c$  values (0.8 and 0.9), processing delay is reduced since NC will mostly prevail at  $(N - 1)$  combined packets that can help  $(N - 1)$  nodes. Also, the processing delay for  $P_c = 0.9$  is less than the delay for  $P_c = 0.8$  because the number of the packets considered for local repair operation for  $P_c = 0.9$  is less than for  $P_c = 0.8$ . Figure 2.7 shows that, for a certain number of nodes, increasing the file size will lead to increasing the processing delay. The relation between file size and processing delay is not linear. For instance, the processing delay is very small for  $L = 100$  packets. However, it increases significantly for  $L = 300$  packets. Also, from figure 2.7 we see that the required scheduling time is at a maximum when nodes have received 50 % of the file. This is because most of the nodes will be involved at each combination level and the algorithm will look for the best combination with respect to the number of helped nodes  $(N - j)$  and the number of combined packets  $(N - i)$ . On the other hand, the processing delay is very low for  $P_c = 0.1, 0.2$  and  $0.9$ , since most of the packets are either received by most of the nodes ( $P_c = 0.9$ ) or received by a few nodes ( $P_c = 0.1$  and  $0.2$ ). Therefore, most of the scheduled transmissions will be from one combination level that can help  $(N - 1)$  nodes. For example, for  $P_c = 0.9$ , the most scheduled coded packets are combinations of  $(N - 1)$  packets that would help  $(N - 1)$  nodes.

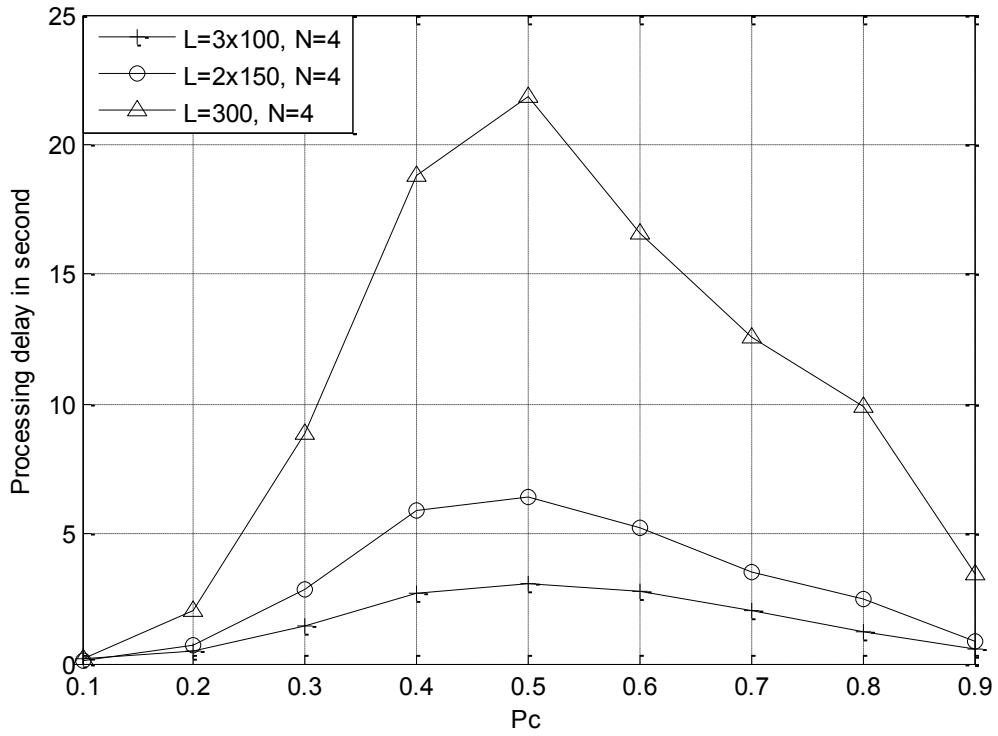


Figure 2.8 Segmentation effects on the processing delay in the new three-phase algorithm

Due to the high processing delay for the large file sizes in figure 2.7, file segmentation may be needed to reduce the processing delay. File segmentation will also reduce the improvement factor, as shown in figure 2.9, but this reduction will not be severe compared to the effects of the significant processing delays.

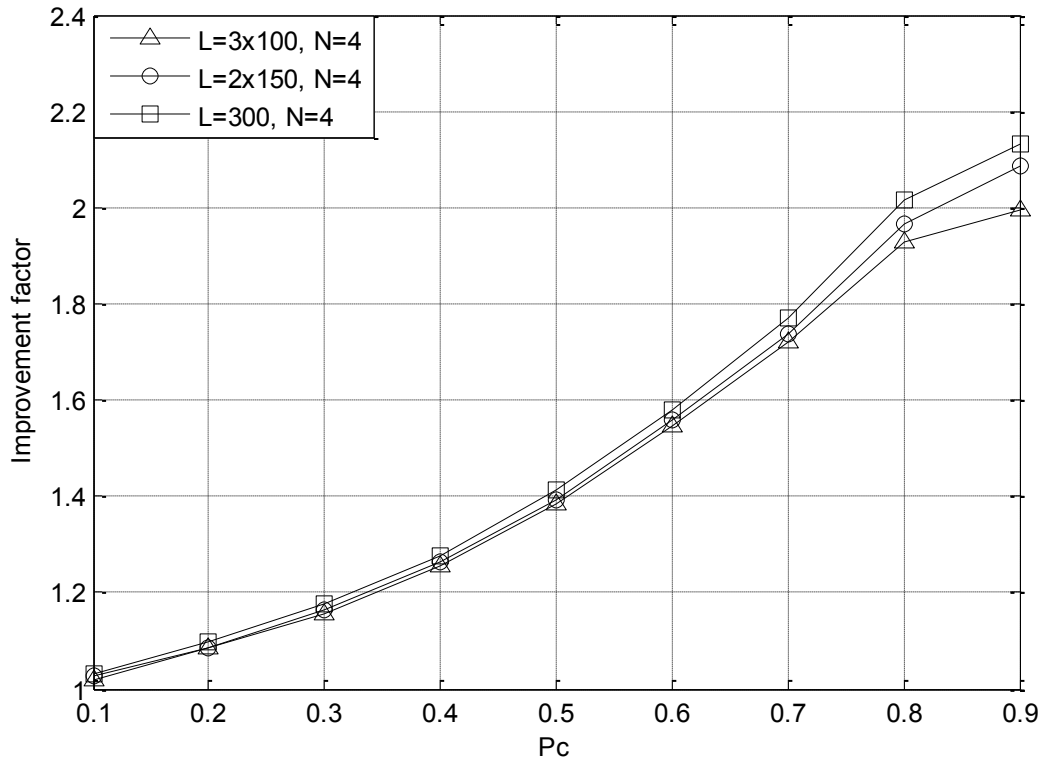


Figure 2.9 Segmentation effects on the improvement factor in the new three phase algorithm

Figure 2.8 shows the processing delays for  $L = 300$  packets and the required processing delay if we segment the same file into two or three files of sizes  $L = 150$  and  $100$  packets. The difference in processing delay between no segmentation and with segmentation of file of size 300 packets is noticeable. At the same time, the improvement factor is reduced by a small amount, as shown in figure 2.9.

## 2.6 Conclusion

In this chapter we have presented a new XOR based scheduling algorithm for NC in cooperative local repair for a group of nodes that can hear each other directly. The algorithm used the knowledge of the packet's availability at neighboring nodes to improve the overall network throughput. In our proposed algorithm, we used network coding to determine which node should transmit and in which time slot (sequential MAC) that would provide the best improvement. The proposed algorithm proceeds in three phases. First, the nodes exchange their packet's availability vectors. This is followed by a short period of distributed scheduling, during which the nodes execute the processing algorithm, developed to minimize the total transmission time. In the third phase, nodes transmit the encoded packets as per the decision of the scheduling algorithm. We have found that the improvement factor provided by the new XOR based scheduling algorithm is very close to the upper bound on the improvement factor. We also investigated the effects of the packet's availability, file size and the number of nodes. We found that as the packet's availability increases, the improvement factor increases. We also found that as the number of nodes or file size increases, the improvement factor and the processing delay will increase. Also, a large file requires more scheduling time than a short file, but it provides a better improvement factor. One solution for this problem is file segmentation which reduces the processing delay significantly while sacrificing a small amount of improvement factor.

## **Chapter 3 RANDOM NETWORK CODING FOR CDMA/TDD WIRELESS MESH NETWORKS**

In chapter 2, the fact that all nodes are within hearing distance from each other facilitates the knowledge of missing packets at each node. Chapter 2 shows consequently how the nodes would benefit deterministically from such knowledge. On the other hand, in WMNs where nodes are organized in multi-hop network, the BS or the sender will have to issue synchronization and other control signals to help in such network. Moreover, in such multi-hop networks, a simple XOR deterministic NC may not be feasible and we would have to resort to random network coding. This chapter studies the effect of RNC on CDMA/TDD WMNs.

### **3.1 Introduction**

Code division multiple access/time division duplex (CDMA/TDD) access has been a subject of great interest in cellular mobile systems [10], [11]. Recently, research efforts have extended the CDMA/TDD technique to backbone WMNs [13], [57]. In [13], the transmitter and receiver structure of a CDMA/TDD system for WMNs was introduced and the delay and delay jitter were computed. NC, a technique to improve the system throughput, was initiated by Ahlswede [1]. Without NC, intermediate nodes merely replicate and forward their received packets to their neighbors. However, with NC, intermediate nodes code/combine a number of received packets and then send the resulting coded packet to their neighbors. The simplest way to create a coded packet is to XOR a certain number of the original packets [3]. A neighbor node, upon receipt of one

such coded packet, will then be able to find an individual component packet, provided that it has all the other original packets within the coded packet. In a different technique, packets received at a certain intermediate node are multiplied by a set of coefficients. These coefficients are fixed at all nodes [5]. This LNC is sufficient for small and static networks. For large and dynamic networks, a RNC is preferable, where each intermediate node chooses its coefficients randomly and independently from a finite field (Galois Field ( $2^m$ )) [6] whenever a coded packet is formulated. The received packets are weighted according to the chosen coefficients and combined to create the coded packet to be transmitted. These coefficients are also transmitted as headers in the coded packet for decoding purposes at the receiving node, as shown in figure 3.4. For large files, the header size becomes significant, which wastes bandwidth. In addition, the destination node will not be able to decode the coded packet until it receives a sufficient number of independent copies of the coded packet. These are the two most challenging problems caused by the use of RNC. However, RNC provides two significant advantages. Firstly, if nodes in the network are mobile, thus the number of neighbors and the neighbors are dynamic. Therefore, LNC in such a case is not possible. Secondly, if the channels between nodes are poor, then the redundant transmissions from different neighbors will provide some improvement. Many works in NC have proved that applying NC to networks could provide significant improvement in overall system throughput [58-62]. In [61], RNC is proposed for TDD channels to optimize the mean time to complete the transmission of a block of packets by determining the number of transmissions that the sender should transmit back-to-back before waiting for acknowledgment. In [63], the authors have suggested that peer to peer networks are the best application for NC. A cross



layer scheme that incorporate NC and subcarrier assignment in OFDMA system is proposed in [62].

The outline of this chapter is as follows. In section 3.2, the system model is presented, and system analysis is presented in 3.3. Numerical and simulation results are presented in section 3.4. The conclusion is in section 3.5.

## **3.2 System Model**

We consider a WMN with a single source node (s) which has a file of length (L) packets to be sent to all of the nodes in the WMN, as shown in figure 3.1. Nodes in consecutive hops may not hear each other if there is an obstacle between them, and so the existence of a line-of-sight channel between nodes that can hear each other is assumed. Nodes are scattered away from the single source node in multi-hop fashion, thus constituting a WMN. Half-duplex communication links are assumed, which implies that nodes can only receive or transmit at any given time. Topology configuration is done by Hello messages. Whenever a node wants to become a member of a multicast group or network, it sends a Hello message to its neighbors. Every node hearing the Hello message sends a Hello message to its own neighbors, and so on, similar to the hand shaking and clustering techniques used in ad hoc wireless LANs [64] (which we do not discuss in this work). These Hello messages will be repeated until each node knows its neighbors, and its hop identity (even or odd), the scrambling code and the assigned Walsh functions at each neighbor. Each node dynamically builds its receiver and determines its transmitting and receiving time slots accordingly. Nodes that have just joined the network will listen

to Hello messages from their neighbors. These received Hello messages contain the hop identity (even or odd) of the transmitting node, scrambling code and the Walsh functions used at the transmitting node. Accordingly, the newly joined node will determine its hop, scrambling code and its Walsh functions. Scrambling codes and Walsh functions are integral parts of the CDMA transmitter and receiver, as will follow shortly. Nodes having the same hop identity cannot hear each other, and transmit simultaneously for half of the time (e.g. in even number slots) and receive during the other half. For instance, in figure 3.1, nodes (R1, R2) and (R6, R7) will simultaneously transmit half of the time (i.e. during even numbered slots) and receive during the other half.

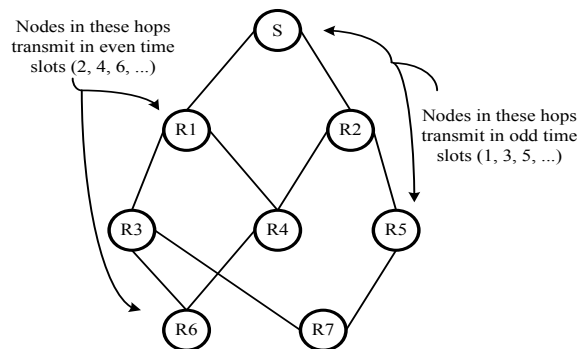


Figure 3.1 System topology as an example

The number of Walsh functions used at each node is fixed. Users only hear some nodes of their immediate hops (hops before and after the users), so the utilization of the same Walsh function can take place at further hops. To avoid excessive interference that will degrade the performance, the total number of assigned Walsh functions at all of the neighbors transmitting to a particular node should not exceed the Processing Gain. All nodes in the network have the same transmission power (divided over the number of

Walsh functions used at the transmitting node). The number of receivers at each node can be determined by knowing the number of neighbors and the number of Walsh functions used at each transmitting neighbor. We assume the existence of a strong error detection code (e.g. CRC) to each network coded packet (NCP). At the receiver, the network decoding process commences only if the CRC indicates a valid NCP. The same is repeated for each of the simultaneously received NCP from the neighbors at any time, such details are not the subject of this thesis.

In this chapter, two scenarios are compared in terms of their efficiency. In scenario 1, the source node uses more Walsh functions than other nodes. In addition, the source node does not generate an NCP. In scenario 2, no network coding is used and the number of assigned Walsh functions is the same at all nodes. These two scenarios are described in more detail below.

### **3.2.1 Scenario 1**

Figure 3.2 shows the transmitter block diagram at the intermediate nodes in scenario 1. In figure 3.2, a number of  $g$  orthogonal Walsh functions enable parallel transmission of different data streams from one transmitter to its neighbors. These Walsh functions also spread the data such that the spread spectrum bandwidth divided by the data bandwidth yields the Processing Gain. The Processing Gain is between 8 - 64 in the applicable WMNs. The scrambling codes in turn randomize the chip identities of

different users so as to avoid lock situations where different nodes transmit the same chip identities for a long duration.

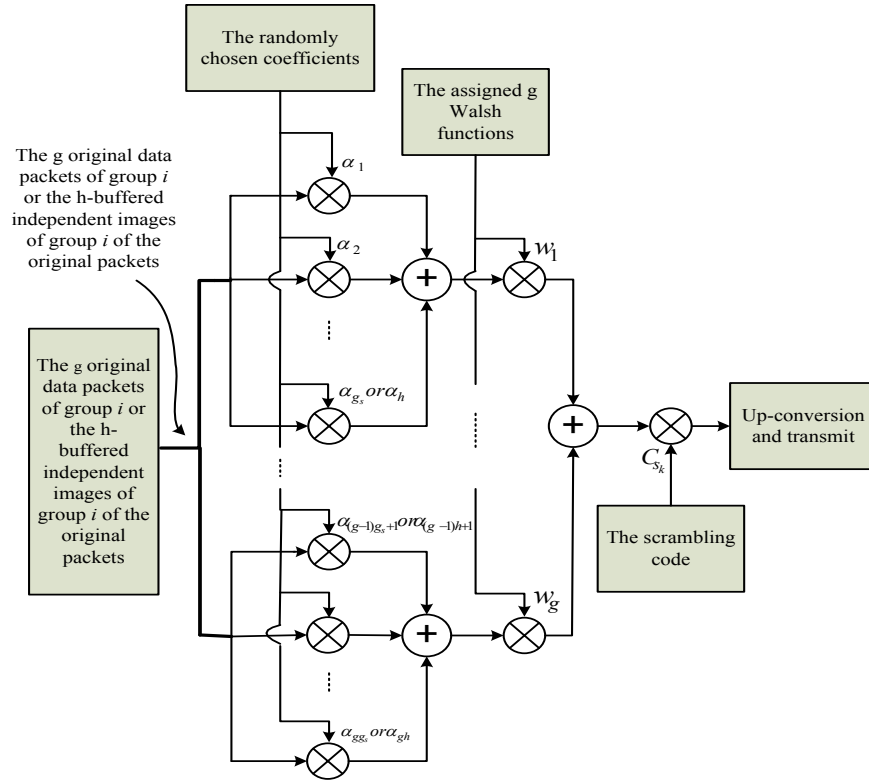


Figure3.2 Block diagram of the transmitter at the intermediate nodes (scenario 1)

As indicated above, the number of assigned Walsh functions at the source node ( $g_s$ ) is greater than the number of assigned Walsh functions at the other nodes ( $g$ ). Data is divided into groups, each group consists of  $g_s$  packets. The source node in scenario 1 sends  $g_s$  original packets using the assigned  $g_s$  Walsh functions, the original data packets being used implies no RNC is applied at the source node. The Walsh coded packet, thereby created by the source node that related to the  $i^{th}$  group of the original data packets, can be presented as follows:

$$Y_{i,s} = \sum_{j=g_s \times (i-1) + 1}^{i \times g_s} w_{I(j)} \times X_j \quad (3.1)$$

where  $I(j)$ ,  $j = 1, 2, \dots, g_s$  is the identity of the assigned Walsh functions at the source node for parallel data transmissions,  $X_j$  is the  $j^{\text{th}}$  original data packet and  $Y_{i,s}$  is the Walsh coded packet related to group  $i$  of the original data packets generated by node  $s$  (the source node). For example, assume  $g_s = 4$  and  $i = 3$ , denotes the group of original data packets (packets 9, 10, 11 and 12 transmitted by the source node). The total number of the NCPs is  $\frac{L}{g_s}$ , Where  $L$  is the file size in data packets.

Depending on the channel quality, some nodes in the first hop receive the transmitted source packet. Nodes that have received the source coded packet decode it using the corresponding scrambling and Walsh functions to get the original  $g_s$  data packets. The number of assigned Walsh functions at each intermediate node ( $g$ ) is the same and is fixed. All intermediate nodes, including nodes in the first hop, utilize RNC to generate an NCP. The generated  $l^{\text{th}}$  image of an NCP related to group  $i$  of the original data packets by a node  $k$ , which is assumed to have received the  $i^{\text{th}}$  source coded packet, can be presented as follows:

$$Y_{i,l,k} = \sum_{j=g_s \times (i-1) + 1}^{i \times g_s} \alpha_{j,l,k} \times X_j \quad (3.2)$$

where  $\alpha_{j,l,k}$  is the  $j^{\text{th}}$  randomly chosen coefficient from a finite field to generate the  $l^{\text{th}}$  image of the  $i^{\text{th}}$  NCP by node  $k$ .  $X_j$  is the  $j^{\text{th}}$  original data packet and  $g_s$  is the number of assigned Walsh functions at the source node and also it presents the number of the

original data packets constituting the generated NCP. Utilizing the parallel transmission capability of the CDMA system, the number of images of an NCP transmitted by an intermediate node in each time slot is  $g$ . The signal generated by node  $k$  after Walsh spreading can be presented as follows:

$$Z_k = w_1 \times Y_{i,1,k} + w_2 \times Y_{i,2,k} + \dots + w_g \times Y_{i,g,k} \quad (3.3)$$

where  $w_g$  is the  $g^{\text{th}}$  Walsh function. The Walsh encoded packet ( $Z_k$ ) is scrambled by the scrambling code ( $C_{s_k}$ ), then it is up-converted and transmitted. All of the nodes in the network use the same scrambling code, but each one uses different shifts. The signal transmitted by node  $k$  can be presented as follows:

$$V_k = C_{s_k} \times Z_k \times \cos \omega_c t \quad (3.4)$$

The preceding analysis (in 3.2, 3.3, and 3.4) is made under the assumption that the transmitting node has received the  $g_s$  original data packets of group  $i$ . If it is not the case, all of the nodes that have not received the constituting packets of the  $i^{\text{th}}$  NCP (the original data packets of group  $i$ ), and depending on the availability of the independent images of the  $i^{\text{th}}$  NCP, combine the received images that belong to the  $i^{\text{th}}$  NCP using RNC. Assume there are  $h$  independent images that belong to the group  $i$  of the original data packets at a node  $u$ . The  $l^{\text{th}}$  image of the  $i^{\text{th}}$  NCP generated by node  $u$  is:

$$Y_{i,l,u} = \sum_{a=1}^h \alpha_{a,l,u} \times Y_{i,a,u} \quad (3.5)$$

where  $Y_{i,a,u}$  is the  $a^{\text{th}}$  encoded image of the  $i^{\text{th}}$  NCP that already exists at node  $u$ . The chosen NCP is determined by the scheduling algorithm. Using the assigned Walsh

functions, each node combines  $g$  different images belonging to the NCP, as per (3.6).

The Walsh coded packet generated by node  $u$  can be presented as follows:

$$Z_u = \sum_{j=1}^g w_j \times Y_{i,j,u} \quad (3.6)$$

in which  $w_j$  is the  $j^{\text{th}}$  assigned Walsh function at node  $u$ , and  $Y_{i,j,u}$  is the  $j^{\text{th}}$  generated image of the  $i^{\text{th}}$  NCP by node  $u$ . This Walsh encoded packet ( $Z_u$ ) is scrambled by the scrambling code ( $C_{s_u}$ ) and then up-converted and transmitted. The signal transmitted by node  $u$  can be represented as:

$$V_u = C_{s_u} \times Z_u \times \cos \omega_c t \quad (3.7)$$

To distinguish between the different NCPs at intermediate nodes, the source node tags each original data packet with a certain number, called a sequence number (SN). The chosen random coefficients and the sequence number are transmitted as a network coding header, as shown in figure 3.4. The receiver block diagram is shown in figure 3.3. At the receiver, the received signal is first down-converted (multiplied by the carrier and filtered) and de-scrambled using the corresponding scrambling codes (the assigned scrambling code at each neighbor), as shown in figure 3.3. The resulting signals from all the neighbors are then de-spread using the assigned Walsh functions at the neighbors. Perfect acquisition and synchronization of the scrambling and Walsh codes at any receiver are assumed. After the CDMA de-spreading process, the resulting NCPs are processed according to their SN, as shown in figure 3.3. If the new received image of an NCP is independent of the previously buffered images of the NCP (found by the rank testing as will follow in the simulation section), then the new received image will be

buffered together with previously stored NCP images. Otherwise, the received node will discard it. When a node buffers  $g_s$  or more independent images of an NCP, the node decodes the packet using Gaussian elimination and the node piggybacks the acknowledgment signal in the next transmission.

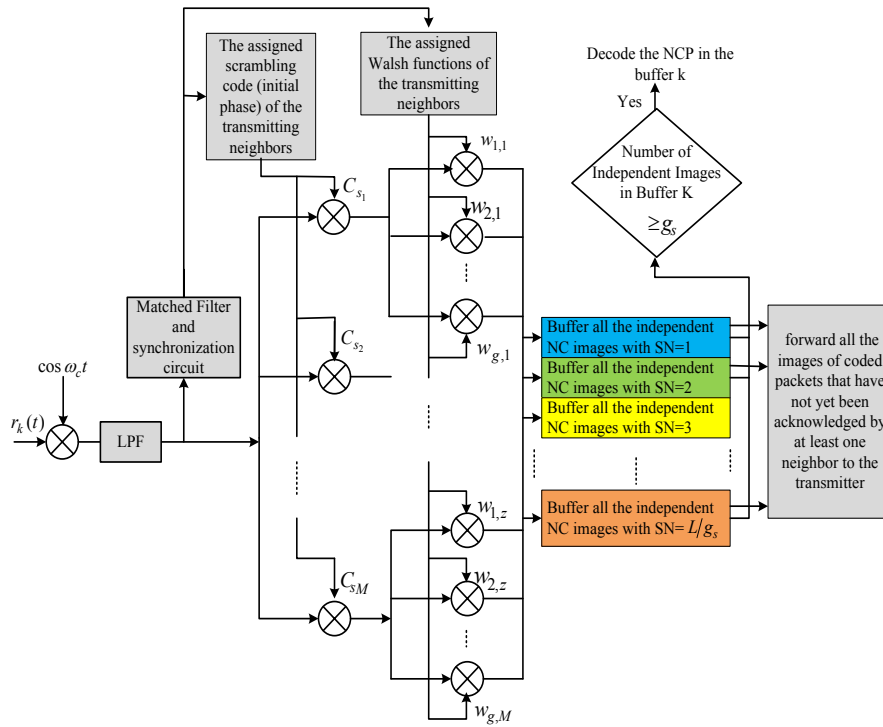


Figure 3.3 Block diagram of the receiver (scenario 1)

Acknowledgment is sent by a node only if the node is able to decode the NCP. The packet format is similar to the one in [13], except for the added network coding and the Ack headers, as shown in figure 3.4. A matched filter at the receiver, i.e. matched to the short synchronization preamble code. In figure 3.4, the short synchronization preamble code field detects the packet start of the received signal from a neighbor. This enables the detection of the identities of the scrambling code and the Walsh functions from the



corresponding field in figure 3.4, since the scrambling code and the Walsh functions are spread by the synchronization preamble field. Then the identity of the scrambling code field helps the receiver to synchronize its local scrambling code to the scrambling code received from the neighboring transmitter. The identity of the Walsh function code field helps the receiver to determine the Walsh functions used at the neighboring transmitter. The network coding header field conveys the randomly chosen coefficients and the sequence number of the NCP transmitted by the neighbor, both of which are necessary for appropriate decoding of the NCP. The NCP field is the NCP created by the neighbor. To have a dynamic network we may allow dynamic changes of the identities of the scrambling and Walsh codes for future development. For that reason, the identity of the Walsh function code field and the identity of the scrambling code field in figure 3.4 are needed in each packet.

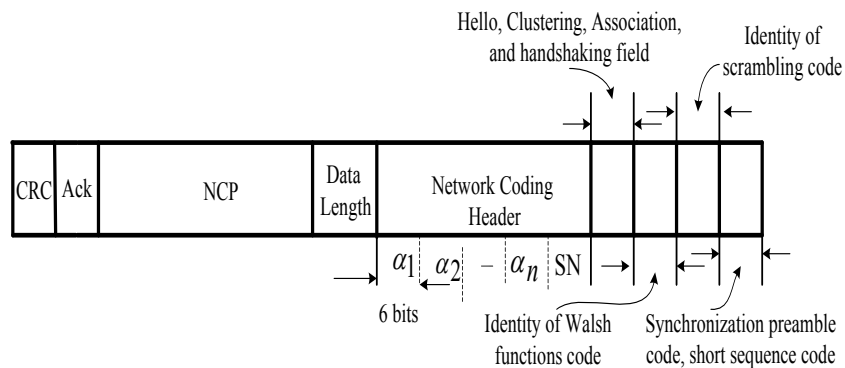


Figure 3.4 The transmitted packet format

Handshaking, Hello, Clustering and association form a field, which is used for topology configuration, determining the transmitting and receiving time slots and Walsh function assignments. The Ack field is for the acknowledgement signal.

### 3.2.2 Scenario 2 (No network coding)

No network coding is used at all in scenario 2--only plain CDMA/TDD. All nodes, including the source node, use the same number of Walsh functions. The number of assigned Walsh functions at each node is  $g$ . All nodes in scenario 2, depending on the availability of the unacknowledged original data packets received by the nodes, send  $g$  original data packets using the assigned  $g$  Walsh functions. The generated Walsh coded packet related to group  $i$  of the original data packets by a node  $u$  ( $Y_{i,u}$ ) can be presented as follows:

$$Y_{i,u} = \sum_{j=g \times (i-1) + 1}^{i \times g} w_{I(j)} \times X_j \quad (3.8)$$

where  $I(j)$ ,  $j = 1, 2, \dots, g$  is the identity of the assigned Walsh functions at node  $u$ , and  $X_j$  is the  $j^{\text{th}}$  original data packet. The Walsh coded packet ( $Y_{i,u}$ ) is scrambled by the scrambling code ( $C_{s_u}$ ) and then up-converted and transmitted. The signal transmitted by node  $u$  can be presented as follows:

$$V_u = C_{s_u} \times Y_{i,u} \times \cos \omega_c t \quad (3.9)$$

In both scenarios, for reliable packets transfer, a Selective Repeat ARQ protocol is used. In the Selective Repeat ARQ protocol, the send window ( $w_s$ ) at the transmitter consists of a certain number of packets. A transmitting node transmits only one of the NCPs (scenario 1) that are within  $w_s$ . For example, if  $w_s = 4$  and the smallest unacknowledged NCP (scenario 1) at the transmitting node is related to group 6 of the original data

packets. The transmitting node transmits an NCP that is related to groups 6 to 9. Similarly, at the receiver side, a receiving node accepts only NCPs (scenario 1) that are within the receive window ( $w_r$ ). For instance, if  $w_r = 4$  and the smallest undecodable NCP (scenario 1) is related to group 8 of the original data packets. Then the receiving node accepts only the NCPs that are related to groups 8 to 11. Nodes use a round robin scheduling algorithm. Let us assume the number of neighbors of a receiving node  $d$  is  $M_d$ . The received signal at the intermediate node  $d$  for both proposed scenarios is the summation of the signals transmitted by all of the neighbors, and which can be presented as follows:

$$r_d(t) = \sum_{e=1}^{M_d} V_{I(e)} + AWGN \quad (3.10)$$

where  $I(e), e = 1, 2, \dots, M_d$  is the identity of the neighbors and AWGN is the additive white Gaussian noise. At the CDMA/TDD receiver, the synchronization of scrambling codes and Walsh functions are assumed to be perfect operations. Orthogonal Walsh functions are assigned at each node for packets transmitted in parallel during the same time slot. Thus, the interference between coded packets that belong to the same transmitted node at the receiver side is negligible. On the other hand, packets received from different neighbors will interfere with each other, since the propagation delay of each neighbor varies. Let us assume a neighbor  $q$  of the receiving node  $d$  uses  $w_q$  Walsh functions, and the total transmission power at each neighbor of the receiving node  $d$  is  $S$ . Each signal transmitted by node  $q$ , which is spread by a certain Walsh function, is considered a Walsh signal. The power of each Walsh signal transmitted by the neighbor  $q$

is  $(\frac{S}{w_q})$ . The composite signal received at node  $d$  consists of the signals transmitted by all of the  $M_d$  neighbors. One of these  $M_d$  signals is intended for a receiving branch in the receiving node  $d$ , as in figure 3.3, and the remaining  $(M_d - 1)$  received signals act as interference. We assume the total power of the AWGN after CDMA spreading is  $N'$ . The received signal to interference plus noise ratio  $(\frac{S}{I})$  at the receiving node  $d$  of a Walsh signal transmitted by the neighbor  $q$  is:

$$\left(\frac{S}{I}\right) = \frac{\frac{S}{w_q}}{(M_d - 1) \times S + N'} \quad (3.11)$$

The energy per bit ( $E_b$ ) is the ratio between the Walsh signal power and the bit rate ( $R_b$ ).

$$E_b = \frac{S}{R_b \times w_q} \quad (3.12)$$

The power spectral density of the receiver's AWGN is the same with and without CDMA spreading, thus

$$I_0 + N_0 = \frac{(M_d - 1) \times S}{W} + N_0 \quad (3.13)$$

where  $N_0$  is the noise power spectral density. The Processing Gain is the ratio between the total bandwidth ( $W$ ) and the bit rate ( $R_b$ ). The energy/bit to interference plus noise power spectral density  $(\frac{E_b}{I_0 + N_0})$  can be found as follows:

$$\left(\frac{E_b}{I_0 + N_0}\right) = \frac{1}{w_q \times \frac{(M_d - 1)}{P_G} + \frac{N_0}{E_b}} \quad (3.14)$$

$\left(\frac{N_0}{E_b}\right)$  in the denominator of (3.14) is the inverse of the thermal noise and  $P_G$  in (3.14) is the Processing Gain. By using  $\left(\frac{E_b}{I_0 + N_0}\right)$  and the selected channel coding, the bit error rate ( $P_b$ ) can be computed (using turbo coding,  $P_b$  on page 252, figure 7.32 in [65]). Assuming  $n_b$  independent bits per packet, the probability of packet success ( $P_c$ ) is:

$$P_c = (1 - P_b)^{n_b} \quad (3.15)$$

For scenario 1, let us denote the total number of nodes in the network as  $N$  and the number of time slots required until all nodes receive the entire file ( $L$  original data packets) and all nodes receive acknowledgment signals for all  $L$  original packets from all of their neighbors as  $F$  (found from the simulation results).

The system efficiency of scenario 1 ( $\eta_{NC}$ ) is:

$$\eta_{NC} = \left(\frac{2 \times \left(\frac{L}{g}\right)}{F}\right) \times \left(\frac{(N - 1) \times R_b}{W}\right) \times \left(\frac{n_b - g_s \times m}{n_b}\right) \quad (3.16)$$

Where the first part of (3.16) presents the RNC effects, the second part of (3.16) is the typical CDMA transmission efficiency and the third part of (3.16) is the effects of RNC header. For scenario 2, let us denote the number of time slots required until all nodes receive the entire file ( $L$  original data packets) and all nodes receive acknowledgment signals for all  $L$  original packets from all of their neighbors as  $F'$  (found from the simulation results).

The system efficiency for scenario 2 ( $\eta_{NNC}$ ) is:

$$\eta_{NNC} = \left( \frac{2 \times \left( \frac{L}{g} \right)}{F'} \right) \times \left( \frac{(N-1) \times R_b}{W} \right) \quad (3.17)$$

The difference between (3.16) and (3.17) is the effect of the RNC header. We assume a real time data transfer application. The delay of the  $n^{th}$  original data packet at a node  $x$  for a certain generated topology  $H$  is defined as the difference between the time slot when node  $x$  has received the original data packet ( $T_{n,x,H}$ ) and the time slot when the  $n^{th}$  original data packet should have been synchronously transmitted from the source node ( $T_{n,H}$ ). The delay of the  $n^{th}$  original data packet at node  $x$  for topology  $H$  is:

$$D_{n,x,H} = T_{n,x,H} - T_{n,H} \quad (3.18)$$

The average delay of the original data packet  $n$  averaged over all intermediate nodes for topology  $H$  is:

$$D_{n,H} = \frac{\sum_{x=1}^{N-1} D_{n,x,H}}{N-1}$$

The average delay of all packets at all intermediate nodes for topology  $H$  is:

$$D_H = \frac{\sum_{n=1}^L D_{n,H}}{L} \quad (3.19)$$

Denote the number of generated topologies as  $\delta$ . The average delay of all packets at all nodes for all topologies is:

$$D = \frac{\sum_{H=1}^{\delta} D_H}{\delta} \quad (3.20)$$

The delay jitter between packets  $n$  and  $n+1$  at node  $x$  for topology  $H$  can be presented as follows:

$$\bar{J}_{n,x,H} = |D_{n+1,x,H} - D_{n,x,H}| \quad (3.21)$$

The delay jitter for the data file at node  $x$  for topology  $H$  is:

$$\bar{J}_{x,H} = \sum_{n=1}^{L-1} \frac{\bar{J}_{n,x,H}}{L-1}$$

The delay jitter is:

$$\overline{\sigma}_{x,H} = \sqrt{\frac{\sum_{n=1}^{L-1} (\bar{J}_{x,H} - \bar{J}_{n,x,H})^2}{L-2}} \quad (3.22)$$

The average delay jitter is:

$$\sigma = \frac{\sum_{x=1}^{N-1} \sum_{H=1}^{\delta} \overline{\sigma}_{x,H}}{(N-1)\delta} \quad (3.23)$$

The delay and delay jitter in (3.18) to (3.23) will be computed during simulation, as will follow.

### 3.3 Performance Analysis

In this section we present the performance analysis of the scenario 1 where RNC is applied at all intermediate nodes but not at the source node, and the number of the assigned Walsh functions at the source node ( $g_s$ ) is more than the number of the assigned

Walsh functions at intermediate nodes ( $g$ ). A subject node is defined as an intermediate node that cannot hear the source node directly (any intermediate node that is not located in the first hop). Let us assume that the number of neighbors of the subject node is  $M$ , and the number of assigned Walsh functions at the subject node and at its  $M$  neighbors is  $g$ . Equation (3.14) is used to find the energy/bit to interference plus noise power spectral density ( $\frac{E_b}{I_0+N_0}$ ). Using the result from equation (3.14) and the selected channel coding, the bit error rate ( $P_b$ ) can be computed (using for example turbo coding,  $P_b$  on page 252, figure 7.32 in [65]). Then (3.15) is used to find the probability of packet success at the subject node for a given  $M$  neighbors ( $P_{c,M}$ ).

Let us denote the maximum number of neighbors of any node in the network as  $Q$ , and the minimum number of neighbors of any node in the network as  $J$ . For further development network connectivity ( $\xi$ ) is defined as the ratio between the summation of the neighbors of all nodes and the number of nodes. For example, in figure 3.1, the network connectivity is  $(2+3+3+3+3+2+2+2)/8 = 2.5$ .

For a given network connectivity ( $\xi$ ), the total number of nodes ( $N$ ) in the network,  $Q$  and  $J$ , the total number of possible topologies that may exist is:

$$\begin{aligned}
 q = & \sum_{\epsilon_1=0}^N \sum_{\epsilon_2=0}^N \dots \sum_{\epsilon_{Q-J+1}=0}^N \binom{N}{\epsilon_1} \times \binom{N-\epsilon_1}{\epsilon_2} \times \dots \\
 & \times \binom{N-\epsilon_1-\epsilon_2-\dots-\epsilon_{Q-J}}{\epsilon_{Q-J+1}} \quad (3.24)
 \end{aligned}$$

For  $J \times \epsilon_1 + (J+1) \times \epsilon_2 + \dots + Q \times \epsilon_{Q-J+1} = N \times \xi$ ,  
and  $\epsilon_1 + \epsilon_2 + \dots + \epsilon_{Q-J+1} = N$



where  $\mathcal{C}_1$  in equation (3.24) represents the number of nodes that have  $J$  neighbors,  $\mathcal{C}_2$  in equation (3.24) represents the number of nodes that have  $(J + 1)$  neighbors, and so on. If one node out of the  $N$  nodes has  $M$  neighbors, then the total possible topologies that may exist is:

$$\Omega = \sum_{\varphi_1=0}^{N-1} \sum_{\varphi_2=0}^{N-1} \dots \sum_{\varphi_{Q-J+1}=0}^{N-1} \binom{N-1}{\varphi_1} \times \binom{N-\varphi_1-1}{\varphi_2} \times \dots \times \binom{N-\varphi_1-\varphi_2-\dots-\varphi_{Q-J}-1}{\varphi_{Q-J+1}} \quad (3.25)$$

For  $J \times \varphi_1 + (J + 1) \times \varphi_2 + \dots + Q \times \varphi_{Q-J+1} = N \times \xi - M$ ,  
 And  $\varphi_1 + \varphi_2 + \dots + \varphi_{Q-J+1} = N - 1$

where  $\varphi_1$  in equation (3.25) represents the number of nodes that have  $J$  neighbors,  $\varphi_2$  in equation (3.25) represents the number of nodes that have  $(J + 1)$  neighbors, and so on. We note the subtraction of one from  $N$  nodes in various terms in Eq. (3.25) is due to the exclusion of the subject node with a given number of neighbors ( $M$  neighbors). Thus the probability that the subject node has  $M$  neighbors is

$$P(M/\xi) = \frac{\Omega}{\varrho} \quad (3.26)$$

Two methods to find the average achievable NC benefit per time slot at the subject node for the performance analysis of the system are derived.

### 3.3.1 Analysis by enumeration

For a given network connectivity ( $\xi$ ), the total number of nodes ( $N$ ), the number of neighbors of the subject node ( $M$ ),  $Q$  and  $J$ , and the  $M$  neighbors of the subject node having  $U$  neighbors, the total possible topologies that we can have is given by:

$$\begin{aligned} \Phi = & \sum_{I_1=J}^Q \sum_{I_2=J}^Q \dots \sum_{I_M=J}^Q \sum_{\varphi_1=0}^{N-M-1} \sum_{\varphi_2=0}^{N-M-1} \dots \\ & \dots \sum_{\varphi_{Q-J+1}=0}^{N-M-1} \binom{N-M-1}{\varphi_1} \times \binom{N-M-1-\varphi_1}{\varphi_2} \times \dots \\ & \times \binom{N-M-1-\varphi_1-\varphi_2-\dots-\varphi_{Q-J}}{\varphi_{Q-J+1}} \end{aligned} \quad (3.27)$$

$$\begin{aligned} \text{for } J \times \varphi_1 + (J+1) \times \varphi_2 + \dots + Q \times \varphi_{Q-J+1} &= N \times \xi - M - U, \\ \varphi_1 + \varphi_2 + \dots + \varphi_{Q-J+1} &= N - M - 1 \\ \text{and } I_1 + I_2 + \dots + I_M &= U \end{aligned}$$

where  $\varphi_1$  in the equation (3.27) represents the number of nodes that have  $J$  neighbors,  $\varphi_2$  in the equation (3.27) represents the number of nodes that have  $(J+1)$  neighbors, and so on, and  $I_1$  represents the number of neighbors of the first neighbor of the subject node,  $I_2$  represents the number of neighbors of the second neighbor of the subject node,  $I_M$  represents the number of neighbors of the  $M^{\text{th}}$  neighbor of the subject node, and so on. The probability that the  $M$  neighbors of the subject node have  $U$  neighbors is:

$$P(U/M, \xi) = \frac{\Phi}{\Omega} \quad (3.28)$$

For a given values of  $Q$ ,  $J$ ,  $U$  and  $M$ ,  $P(U/M, \xi)$  presents many different scenarios. For example, if  $M = 2$ ,  $U = 7$ ,  $Q = 6$  and  $J = 2$ , then  $P(7/2, \xi)$  presents scenarios where one of the neighbors of the subject node has two neighbors and the other has five, or vice versa, or one of the neighbors of the subject node has three neighbors and the other has four, or vice versa. As the neighbors of the subject node ( $M$ ) have more of their own neighbors, the chance that these  $M$  neighbors help the subject node increases. Also, since there is only a single source node in the network, we assume in the best case, for large  $U$ , on average the subject node could receive  $g_s$  independent images per each receiving time slot. In addition, in the worst case, for small  $U$ , on average the subject node receives  $g$  independent images per each receiving time slot. Thus, the average number of independent images received by the subject node per each receiving time slot for a given  $U$  and  $M$  can be assumed to have a linear relation with  $U$ ,  $g$  and  $g_s$  as follows:

$$\mu_U = g + \frac{(g_s - g) \times (U - J)}{(Q - J)} \quad (3.29)$$

The average number of independent images received by the subject node per each receiving time slot for a given  $M$  and  $\xi$  is:

$$B_{M,\xi} = \sum_{U=M*J}^{M*Q} P(U/M, \xi) \times \mu_U \quad (3.30)$$

The average number of independent images received by the subject node per each receiving time slot for a given  $\xi$  is:

$$B_{\xi} = \sum_{M=J}^Q B_{M,\xi} \times P(M/\xi) \times P_{c,M} \quad (3.31)$$

The average number of time slots needed for the subject node to receive the entire file ( $L$  original packets) for a given  $\xi$  is:

$$F_{\xi} = \frac{2 \times L}{B_{\xi}} \quad (3.32)$$

Shortly after, we use (3.32) to find the system efficiency in (3.46).

### 3.3.2 Markov Chain Analysis

Let us consider  $w_t$  to be the window length of NCPs that have been received by all the neighbors of the subject node but have not been acknowledged by at least one neighbor of each neighboring node. In this analysis we assume all neighbors of the subject node have received all the NCPs within  $w_t$ . Also, denote the number of the outstanding NCPs that have been received by all of the neighbors of the subject node but have not yet been acknowledged by the subject node as  $w_r$ . The size of  $w_r$  and  $w_t$  are proportional to  $P_{c,M}$ . At each transmission time slot designated to a neighboring node of the subject node, the neighboring node transmits one of its received NCP within  $w_t$  with equal probability. The subject node will benefit from the neighboring transmissions only if the neighboring node transmits an NCP within the  $w_r$ . The average number of the

decodable NCP at the subject node per time slot can be presented by the Markov chain as shown in figure 3.5. Recall,  $g_s$  is the minimum number of independent images necessary for an NCP to be decodable at the receiver. We denote the ratio between the number of assigned Walsh functions at the source node ( $g_s$ ) and the number of Walsh functions at the intermediate nodes ( $g$ ) as  $\Delta$ .

$$\Delta = \frac{g_s}{g} \quad (3.33)$$

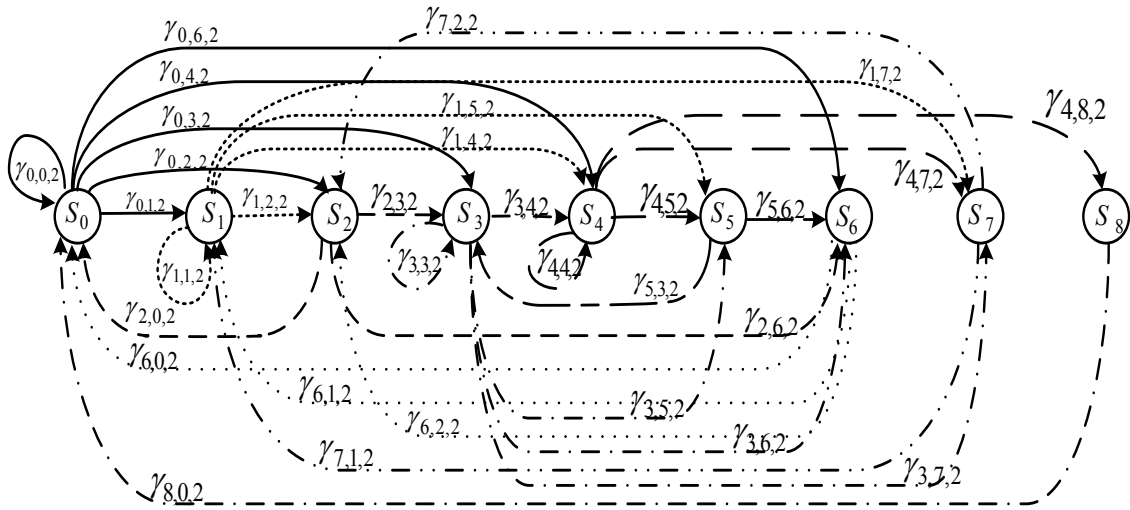


Figure 3.5 Markov chain representation of a node having two neighbors, and  $w_r = 2$ , and

$$\Delta = 2.$$

Each state in figure 3.5 is a vector of  $w_r$  components representing  $w_r$  NCPs expected by the subject node each laying in the range between 0 to  $\Delta$ . For example, let  $w_r = 2$  NCPs and  $\Delta = 2$ , then the states in figure 3.5 can be presented as follows:

$S_0 = [0 \ 0]$ ,  $S_1 = [0 \ 1]$ ,  $S_2 = [0 \ 2]$ ,  $S_3 = [1 \ 0]$ ,  $S_4 = [1 \ 1]$ ,  $S_5 = [1 \ 2]$ ,  $S_6 = [2 \ 0]$ ,  $S_7 = [2 \ 1]$ ,  
 $S_8 = [2 \ 2]$ .

Since the number of assigned Walsh functions at each neighbor is  $g$ , the subject node can only receive a multiple of  $g$  images of an NCP. Zeros indicate that the subject node has not received any images of the NCP, ones - the subject node has received  $g$  independent images of the NCP and twos - the subject node has received  $2 \times g$  independent images of the NCP, and so on. If  $\Delta \times g$  independent images of an NCP have been received at the subject node, then it is assumed to be able to decode the NCP. The number of states of the Markov chain ( $\Gamma$ ) depends on  $\Delta$  and  $w_r$ .

$$\Gamma = (\Delta + 1)^{w_r} \quad (3.34)$$

For instance, if  $w_r = 2$  and  $\Delta = 2$ , then  $\Gamma = 9$ . Each  $S_i$  for  $i = 0, 1, \dots, \Gamma - 1$ , represents the availability of the NCPs at the subject node within  $w_r$ .

When a neighboring node transmits an NCP, the subject node may not benefit from the neighbor transmission in two cases. First, if the transmitted NCP is lost due to the channel effects. Second, if the transmitted images of the NCP are dependent on the already existent images at the subject node. The transition probability to move from state  $i$  to state  $j$  at the subject node for a certain number of neighbors  $M$  is:

$$\gamma_{i,j,M} = \sum_{\lambda=d}^M \binom{M}{\lambda} \times (1 - P_{c,M})^{M-\lambda} \times (P_{c,M})^\lambda \times \left(\frac{1}{\Delta \times w_t}\right)^\lambda \times \psi_{i,j} \quad (3.35)$$

where  $\psi_{i,j}$  is in equation (3.36) and representing the total possibility of combinations of transmitted NCPs from  $\lambda$  neighbors such that the availability of packets at the subject node moves from state  $i$  to state  $j$ ,  $\lambda$  is the number of neighboring nodes from which the subject node has correctly received the transmissions at a certain time slot, and  $(\frac{1}{\Delta \times w_t})^\lambda$  is the probability that  $\lambda$  neighboring nodes transmit  $g$  independent images of a specifically needed NCPs to the subject node within  $w_t$ , which changes the packet availability at the subject node and leads the subject node to move from state  $i$  to state  $j$ . We assume that the probability of transmitting any  $g$  independent images of any NCP within  $w_t$  by any neighboring node is equally likely and equal to  $(\frac{1}{\Delta \times w_t})$ .  $\psi_{i,j}$  in (3.36) is the number of possible transmissions by the  $\lambda$  neighboring nodes that causes the subject node to move from state  $i$  to state  $j$ , and  $d$  in (3.35 and 3.36) is the Euclidean distance between state  $i$  and state  $j$ . For example, if  $w_r = 2$ ,  $M = 2$ ,  $S_i$  (current state) is  $S_3 = [1 \ 0]$  and  $S_j$  (next state) is  $S_5 = [1 \ 2]$ , then  $d = |S_j - S_i| \Rightarrow d = |[1 \ 2] - [1 \ 0]| \Rightarrow d = 2$ . However, decodable NCPs at current state are not considered for Euclidean distance calculations. For example, if  $w_r = 2$ ,  $M = 2$ ,  $S_i$  (current state) is  $S_2 = [0 \ 2]$  and  $S_j$  (next state) is  $S_5 = [1 \ 0]$ , then  $d = |S_j - S_i| \Rightarrow d = |[1] - [0]| \Rightarrow d = 1$ . In order for the subject node to move from state  $i$  to state  $j$  at a certain time slot, at least  $d$  received neighboring nodes transmissions have to be independent images of the currently investigated NCPs in the current received window which will lead the subject node to move to state  $j$ . The other transmissions received from other neighboring nodes could be independent images of the currently investigated NCPs in the current received window which will lead the subject node to move to state  $j$ , or images of the currently investigated NCPs that are

dependent on the ones already existent at the subject node or images of the NCPs that are outside  $w_r$ . Let us denote the cardinality of the current state ( $S_i$ ) as  $|i|$ . For example, if  $S_i$  is  $[1\ 2]$ , then  $|i| = 3$ .  $\psi_{i,j}$  in equation (3.35) can now be presented as follows:

$$\psi_{i,j} = \begin{cases} \tau \times \sum_{\vartheta_1=1}^{\mu} \sum_{\vartheta_2=1}^{\mu} \dots \sum_{\vartheta_\lambda=1}^{\mu} \begin{cases} 1 & \text{if } |\bar{\vartheta} \cap \bar{D}| = d \\ 0 & \text{elsewhere} \end{cases} & \text{if } \lambda = 0 \\ 1 & \text{if } \lambda = 0 \end{cases} \quad (3.36)$$

where  $\bar{\vartheta} = [\vartheta_1\ \vartheta_2\ \vartheta_3 \dots \vartheta_\lambda]$ ,  $\bar{D} = [1\ 2\ 3 \dots d]$

$\mu = |i| + d + \Delta \times (w_t - w_r)$  and

$$\tau = \prod_{A=1}^{w_r} \binom{\Delta - S_{i,A}}{S_{j,A} - S_{i,A}} \quad (3.37)$$

where  $S_{i,A}$  is the number in the current state of the  $A^{\text{th}}$  NCP within  $w_r$  at the subject node, and  $S_{j,A}$  is the number in the next state of the  $A^{\text{th}}$  NCP within  $w_r$  at the subject node. For example, in figure 3.5, assume  $S_i$  (current state) is  $[0\ 1]$  and  $S_j$  (next state) is  $[2\ 1]$ , then  $S_{i,1} = 0, S_{j,1} = 2, S_{i,2} = 1, S_{j,2} = 1$ . If the availability of a certain NCP at the subject node is  $g_s$  or more, then we assume that this coded packet is decodable and thus  $w_r$  is advanced to handle the next expected NCP. Since the Markov chain presents the average number of decodable coded packets per time slot and the effects of sliding  $w_r$  to new NCP will appear only in next time slot, thus the effects of the new NCP will not be included in the same time slot when the older NCPs were completely decoded. For example, if current state is  $[0\ 2]$  and  $M = 2$ , then the possible next states are  $[0\ 0]$ ,  $[1\ 0]$  and  $[2\ 0]$  but not  $[0\ 2]$  or  $[1\ 2]$  or  $[2\ 2]$  as shown in figure 3.5. When we reset a certain decoded packet location to zero, this make the transition diagram applicable to fixed and sliding window cases.



The summation of all outgoing transition probabilities of any given state in a Markov chain must equal one. Thus,

$$\sum_{j=0}^{\Gamma-1} \gamma_{i,j,M} = 1 \quad 0 \leq i \leq \Gamma - 1 \quad (3.38)$$

where  $\gamma_{i,j,M}$  is the transition probability from state  $i$  to state  $j$ . For example, in figure 3.5, assume  $S_i = [0 \ 1]$ . From figure 3.5, where  $M = 2$ , the possible next states are  $S_1, S_2, S_4, S_5, S_7$ . So  $\gamma_{1,1,2} + \gamma_{1,2,2} + \gamma_{1,4,2} + \gamma_{1,5,2} + \gamma_{1,7,2}$  must be equal to one. In figure 3.5, where  $M = 2$ , assume  $g = 4$ ,  $g_s = 8$ ,  $w_r = 2$ ,  $w_t = 2$ ,  $P_G = 32$ ,  $\frac{N_0}{E_b} = 0.01$  and  $n_b = 1000$ . By using equation (3.14) and the selected channel coding,  $P_{c,2} = 0.9906$ . For  $S_i = S_1$  and  $S_j = S_1$   $d = |S_j - S_i| \Rightarrow d = |[0 \ 1] - [0 \ 1]| \Rightarrow d = 0$ . Then by using (3.35), (3.36) and (3.37)

$$\begin{aligned} \gamma_{1,1,2} &= \sum_{\lambda=0}^2 \binom{2}{\lambda} (1 - 0.9906)^{M-\lambda} \times (0.9906)^\lambda \times \left(\frac{1}{2 \times 2}\right)^\lambda \times \prod_{A=1}^2 \binom{2 - S_{i,A}}{S_{j,A} - S_{i,A}} \times \\ &\sum_{\vartheta_1=1}^\mu \sum_{\vartheta_2=1}^\mu \cdots \sum_{\vartheta_\lambda=1}^\mu 1 \\ &= \binom{2}{0} \times (1 - 0.9906)^2 \times \binom{2-0}{0-0} \times \binom{2-1}{1-1} \times 1 \\ &+ \binom{2}{1} \times (1 - 0.9906) \times (0.9906) \times \left(\frac{1}{4}\right) \times \binom{2-0}{0-0} \times \binom{2-1}{1-1} \times 1 \\ &+ \binom{2}{2} \times (0.9906)^2 \times \left(\frac{1}{4}\right)^2 \times \binom{2-0}{0-0} \times \binom{2-1}{1-1} \times 1 \\ &= 0.0661 \end{aligned}$$

Where the first part of  $\gamma_{1,1,2}$  above presents the case where the subject node does not receive any neighboring node transmissions, the second part presents the case where the subject node receives correctly only from one neighboring node and the third part presents the case where the subject node receives correctly from two neighboring nodes.

Again by using (3.35), (3.36) and (3.37),  $\gamma_{1,2,2} = 0.1886$ ,  $\gamma_{1,4,2} = 0.3773$ ,  $\gamma_{1,5,2} = 0.2453$  and  $\gamma_{1,7,2} = 0.1227$ . If  $\gamma_{1,1,2}$ ,  $\gamma_{1,2,2}$ ,  $\gamma_{1,4,2}$ ,  $\gamma_{1,5,2}$  and  $\gamma_{1,7,2}$  are added together the result should be equal to one. After finding all the transition probabilities for a given number of neighbors of the subject node, the probability of each state ( $P(s_i/M, \xi)$ ) is computed as follows:

$$P(s_i/M, \xi) = \sum_{y=0}^{\Gamma-1} s_y \times \gamma_{y,i,M} \quad (3.39)$$

Next we solve the simultaneous equations resulting from (3.39) using the Cramer Rule technique to find the probability of each state. Only probabilities of the states where the subject node could decode one or more NCPs are used for computing the average number of decodable NCPs per each receiving time slot. For example, from figure 3.5, states  $S_2$ ,  $S_5$ ,  $S_6$ ,  $S_7$  present the case where the subject node is able to decode one NCP ( $\beta = 1$ ), and state  $S_8$  presents the case where the subject node is able to decode two NCPs ( $\beta = 2$ ). The average number of decoded NCPs at the subject node per each receiving time slot for a given  $M$  is:

$$C_M = \sum_{\beta=1}^{w_r} \beta \times \sum_{i=1}^{\Gamma} P(s_i/M, \xi) \quad (3.40)$$

for states where the number of decodable NCPs is equal to  $\beta$

Averaging over the neighbor's distribution we obtain the average number of decodable NCPs at the subject node per each receiving time slot as follows:

$$C = \sum_{M=J}^Q C_M \times P(M/\xi) \quad (3.41)$$

where  $P(M/\xi)$  is the probability that the subject node has  $M$  neighbors for the given network connectivity ( $\xi$ ). The average number of time slots needed for the subject node to receive the entire file ( $L$  original packets) for a given  $\xi$  is:

$$F_\xi = \frac{2 \times L}{g_s \times C} \quad (3.42)$$

The probability of a source packet success at a node in the first hop for a given number of neighbors of the node ( $P_{c_s, M}$ ) can be found using (3.14) and (3.15), where  $w_q$  in (3.14) is  $g_s$ . Then, the probability that a certain node in the first hop receives the source packet for a given  $\xi$  is:

$$P'(s) = \sum_{M=J}^Q P_{c_s, M} \times P(M/\xi) \quad (3.43)$$

Let us denote the number of hops in the network as  $\partial$ . The average number of nodes per hop is:

$$\omega = \frac{N - 1}{\partial} \quad (3.44)$$

The probability that at least one node in the first hop receives the source packet is:

$$P(s) = 1 - (1 - P'(s))^\omega \quad (3.45)$$

Then the system efficiency ( $\eta$ ) is:

$$\eta_{NC} = \left( \frac{2 \times \left( \frac{L}{g} \right)}{F_\xi} \right) \times \left( \frac{N - 1}{P_G} \right) \times \left( \frac{n_b - g_s \times m}{n_b} \right) \times P(s) \quad (3.46)$$

### 3.4 Results

In this section we show the effects of RNC on the CDMA/TDD system performance through analysis and computer simulations. A comparison between the performances of the two scenarios is introduced. In simulations, number crunching and discrete event utilizing MATLAB were employed. In addition to using NC and other techniques, we have also utilized turbo coding [65] for forward error correction at the physical layer level so as to improve the probability of bit and hence packet transmission error. In the analysis and the simulation, without going into too much detail, a turbo code that uses identical BCH component code (BCH (31, 21, 5) & BCH (31, 21, 5)) with a 0.51 code

rate was used. Also, the total used bandwidth ( $W$ ) is 100MHz. The maximum possible number of neighbors of any node in the network is fixed to avoid working in very low SNIR. To have accurate and general results in our simulation, 50 different topologies for each given parameter, such as network connectivity, file size, packet size, etc. were generated. Uniform random variables are used to determine the number of nodes in each hop, the number of neighbors of each node, and the existing links in each topology. The value of  $F$  was computed for each topology. We used (3.14) and the curve in page 252, figure 7.32 in [65] to compute  $P_b$ . Then we used (3.15) to compute  $P_c$ . Uniform random variates ( $\theta$ ) were used to determine if the transmitted packet was received correctly. For example, if the uniformly distributed random generator yields 0.8 for a packet, where  $P_c$ , the packet success probability (3.15), is 0.9, then  $\theta < P_c$  and a packet success is recorded. The randomly generated coefficients for network coding were chosen from GF ( $2^6$ ) and transmitted as a header in the NCP. Each coefficient adds 6 bits to the NCP size. The network coding header in the NCP is  $6 \times g_s$  bits.

For scenario 1, if a new image of an NCP is received at node  $d$ , we buffer the coefficients of the image with the coefficients of the previously received images of the NCP. We then compute the rank of the coefficients of all the buffered images of the NCP. If the rank is equal to the number of buffered images, the last image is considered to be independent of the previously buffered images, otherwise node  $d$  discards the last received image. If  $g_s$  or more independent images belonging to an NCP are buffered at node  $d$ , then we assume that node  $d$  is able to decode the NCP and get the original data packets, and so an acknowledgment signal is broadcast by node  $d$  in the next transmission time slot. Also, if the node has received enough images of an NCP ( $g_s$  or more) and is still receiving a new

image of the NCP, the node piggybacks an acknowledgment signal again to make sure that all neighbors have been acknowledged. For scenario 1, when it is time for an intermediate node to transmit (except those, nodes that have received the constituting packets of the NCP), it selects one unacknowledged NCP depending on the scheduling algorithm (round robin). That intermediate node then generates  $g$  new images of the selected NCP using the buffered images of the NCP, as per equation (3.5). These  $g$  new images then spread using the  $g$  assigned Walsh functions at the node. The average number of transmissions required for all of the generated topologies is then computed.

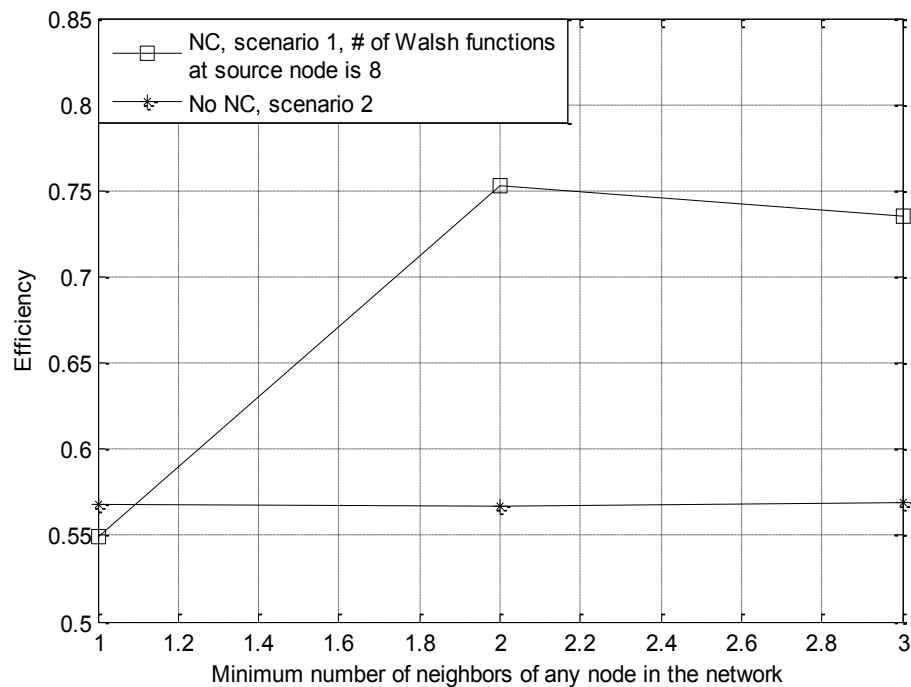


Figure 3.6 The efficiency of the scenarios vs. the minimum number of neighbors. Connectivity = 4,  $N = 20$  nodes  $g = 4$ ,  $g_s = 8$ , number of hops = 6,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $I/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$  and the maximum number of neighbors = 6.

To compute the system efficiency for the given set of parameters, we substitute the average required number of transmissions in (3.16) for scenario 1 and in (3.17) for scenario 2. Figure 3.6 shows the relation between the efficiency of the systems and the minimum number of neighbors for the two scenarios. From figure 3.6 we can observe that if any node in a network has only one neighbor, then scenario 1 (using RNC) provides the worst performance, because of the RNC header in scenario 1. However, if the minimum number of neighbors of any node in the network is two or more, then scenario 1 provides the best efficiency. With scenario 1, nodes in the first hop use original data packets to generate an NCP, and so most of the time intermediate nodes transmit independent images of the NCP. For that reason, nodes with two or more neighbors receive many independent images of the NCP. Hence, the performance of scenario 1 improves. The minimum number of neighbors does not affect the performance of scenario 2, since scenario 2 transmits original data packets, and an acknowledgment signal is piggybacked for each received data packet, without NC. Figure 3.7 shows the effects of the network connectivity on the efficiency of the systems. Figure 3.7 shows that for lower connected networks, scenario 1 provides the worst performance for the given set of parameters ( $g_s$ ,  $g$ , network connectivity, thermal noise, etc.). This is because of the small number of neighbors for each node, which slows the spreading of NCPs through the network. In addition, the low SNIR of the source coded packets at nodes in the first hop, due to the number of Walsh functions used at the source node, degrades system performance. Also, in scenario 1, a node piggybacks an acknowledgement signal for an NCP only if the node has received  $g_s$  or more independent images of the NCP, while in scenario 2, a node piggybacks an acknowledgement signal for each original data packet.

As a result, some nodes may receive images that depend on the previously received images of an NCP. Furthermore, the network coding header, which is  $6 \times g_s$  bits in each NCP, degrades the performance of scenario 1. Figure 3.7 shows that for the given set of parameters, the efficiency of scenario 1 is maximized at a certain connectivity level.

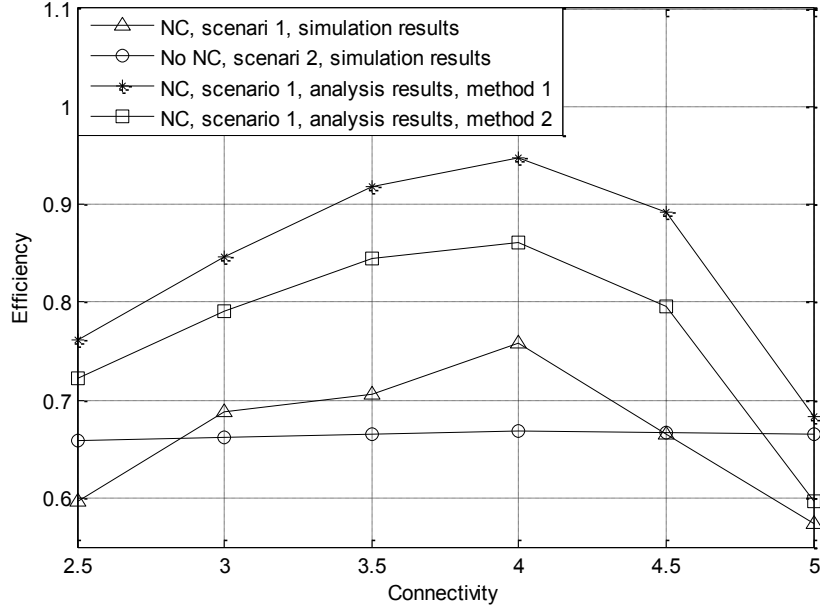


Figure 3.7 The efficiency of the systems vs. the network connectivity, where  $N = 24$  nodes,  $g = 4$ ,  $g_s = 8$ , number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2.

From figure 3.7 we can observe that for network connectivity between 3 and 4, scenario 1 provides the best system performance, thanks to the increase in the network connectivity. However, with scenario 1, increasing the network connectivity does not always lead to better performance, which can be seen in figure 3.7 for network connectivity = 5. Increasing the network connectivity leads to increasing the number of the nodes' neighbors in the network (any node including nodes in the first hop). Increasing the



number of neighbors of a node in the first hop leads to a lower SNIR of the source coded packets at that node.

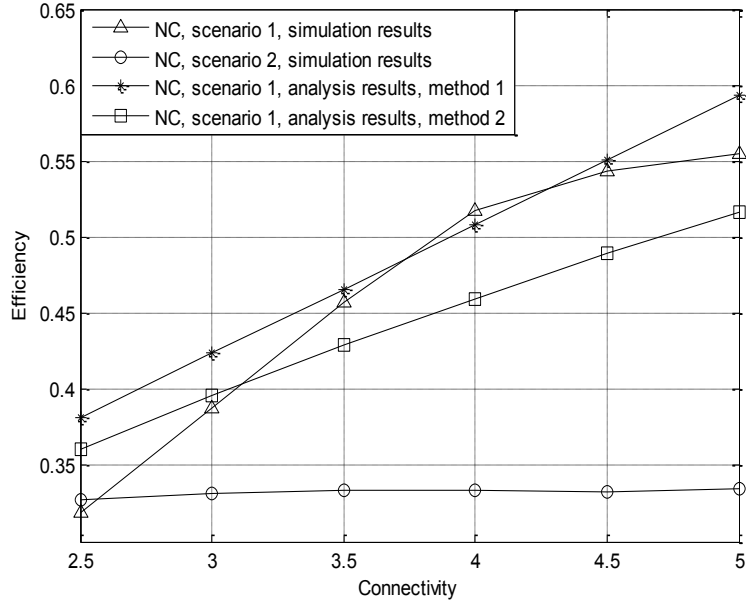


Figure 3.8 The efficiency of the systems vs. the network connectivity, where  $N = 24$  nodes,  $g = 4$ ,  $g_s = 8$ , number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 64$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2.

Consequently, fewer nodes in the first hop receive the source coded packet and thus more transmissions from the source node are required, dragging down the system performance.

Figure 3.8 shows the effects of the Processing Gain on the efficiency of the systems for the same set of parameters as those used in figure 3.7. From figures 3.7 and 3.8 we can observe that increasing the Processing Gain dominates the effects of the high network connectivity but does not necessarily lead to increasing the efficiency. Figure 3.9 shows the relation between the packet size and the efficiency of the systems. From figure 3.9, we observe that the effect of the packet size in scenario 1 is more noticeable than in

scenario 2. From (3.15), as packet size increases the probability of packet success decreases.

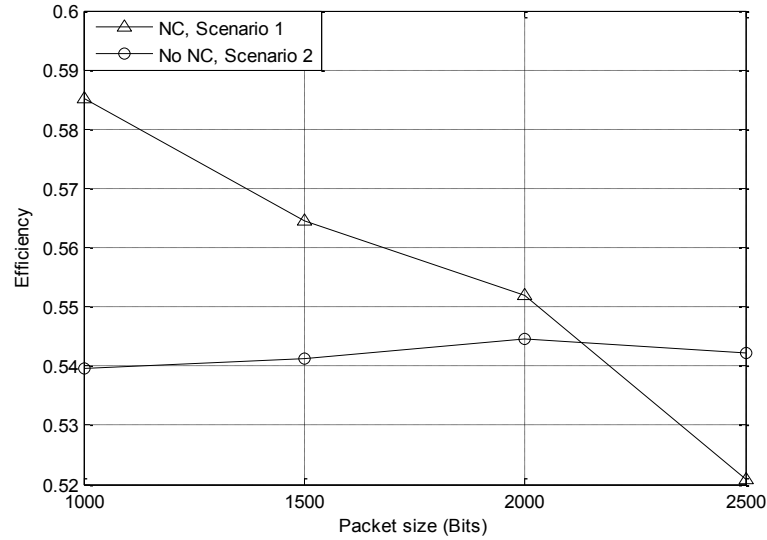


Figure 3.9 The systems efficiency of the scenarios vs. packet size, where  $N = 20$ ,  $g = 4$ ,  $g_s = 8$ ,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ , connectivity = 3,  $L = 120$  packets, number of hops = 5,  $m = 6$  bits,  $w_s = 4$ ,  $w_r = 4$ , minimum number of neighbors = 2 and the maximum number of neighbors = 6.

Also, in scenario 1 the number of Walsh functions assigned at the source node is higher than in scenario 2. For these two reasons, the effect of the packet size in scenario 1 is more noticeable than in scenario 2. Figure 3.10 shows the effects of the size of the GF on the efficiency of scenario 1. In scenario 1, the independency among generated images of an NCP by an intermediate node depends on the size of the GF. From figure 3.10 we note that  $m = 6$  is good enough for the given set of parameters. In figure 3.10, the reduction in the efficiency for  $m = 8$  and 10 is due to the increasing in the size of the network coding header, which increases as  $m$  increases.

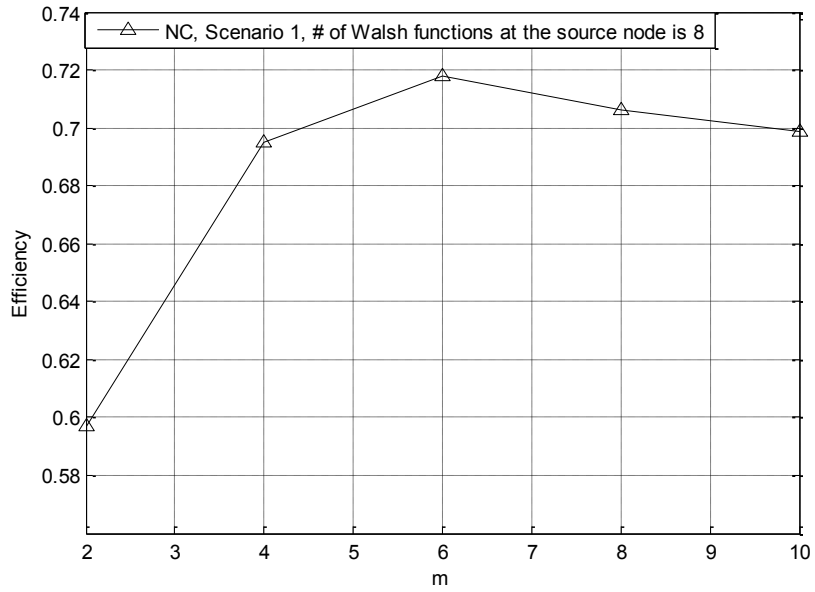


Figure 3.10 The systems efficiency of the scenarios vs.  $m$ , where  $N = 24$ ,  $g = 4$ ,  $g_s = 8$ ,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ , connectivity=3.5,  $L = 120$  packets, number of hops = 5,  $n_b = 1000$  bits,  $w_s = 4$ ,  $w_r = 4$ , minimum number of neighbors = 2 and the maximum number of neighbors = 6.

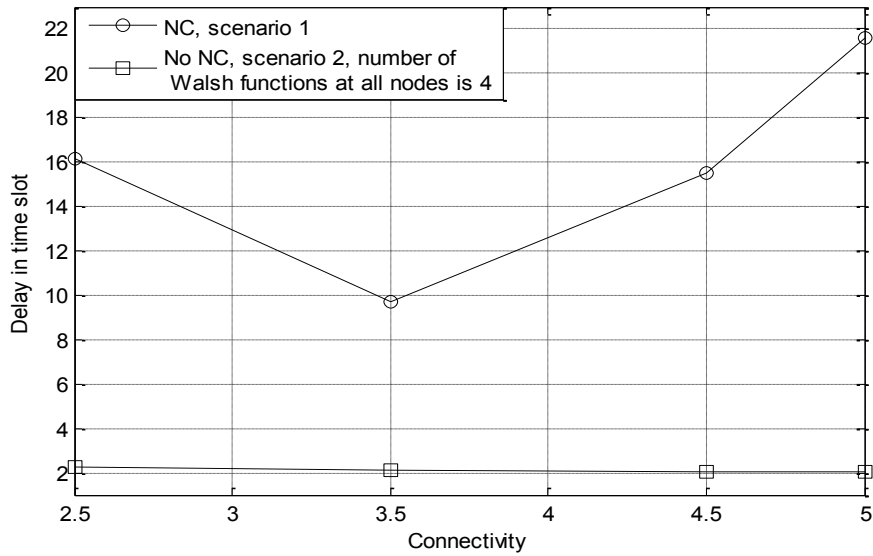


Figure 3.11 The delay vs. connectivity, where  $N = 20$  nodes,  $g = 4$ ,  $g_s = 8$ , number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2.

Figure 3.11 shows the effects of network connectivity on the delay. Lower network connectivity means having fewer neighbors for a node in the network. Therefore, the lower the network connectivity, the lower the number of source retransmissions. For this reason, from figure 3.11, we note that the delay for some low network connectivity (connectivity = 2.5 to 4.5) for scenario 1 is less than the delay for higher connected networks (connectivity = 5). From figure 3.11, we note that the delay for scenario 1 is higher than the delay for scenario 2 for all network connectivity levels. The reason is that the source node was assumed to be able to generate  $g_s$  and  $g$  original data packets, at each transmitting time slot, for scenario 1 and scenario 2, respectively.

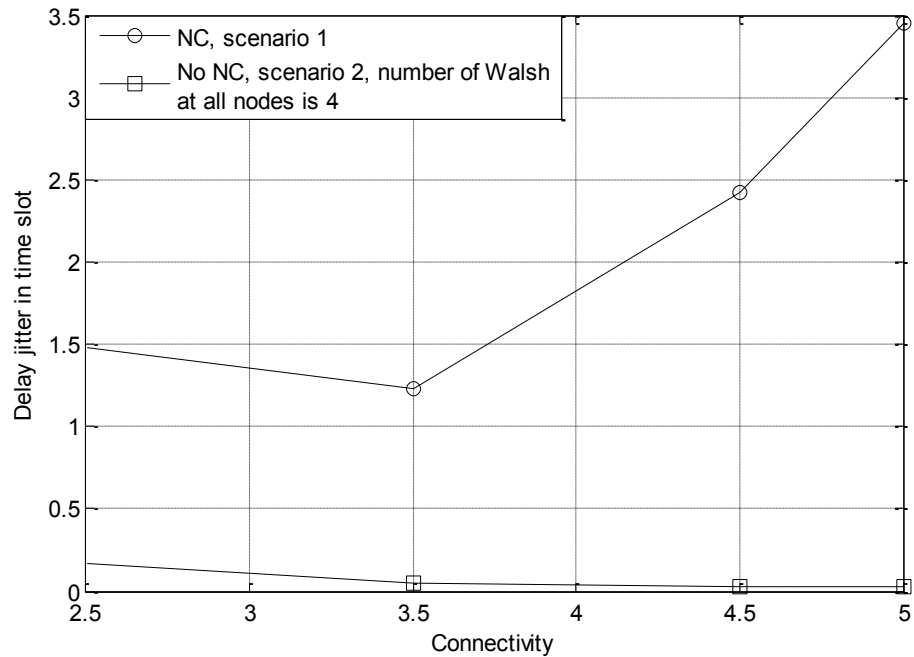


Figure 3.12 The delay jitter vs. connectivity, where  $N=20$  nodes,  $g = 4$ ,  $g_s= 8$ , number of hops =5,  $L =120$  packets,  $n_b =1000$  bits,  $P_G =32$ ,  $1/SNR_{th} =0.01$ ,  $m = 6$ , maximum number of neighbors=6, and the minimum number of neighbors =2.

Figure 3.12 shows the effects of the network connectivity on the delay jitter. From figure 3.12, we note that the delay jitter for scenario 1 is higher than the delay jitter for scenario 2. This is because of the network connectivity, the NC and the number of assigned Walsh functions at the source node for scenario 1.

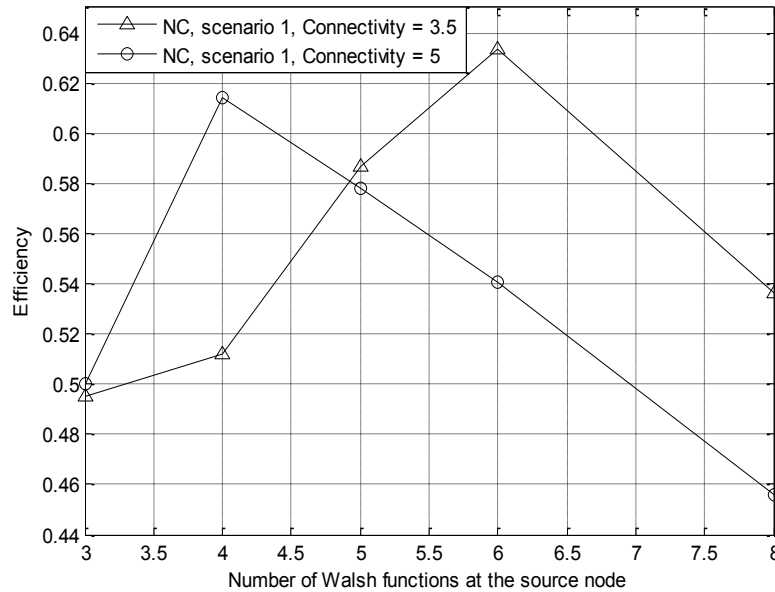


Figure 3.13 Number of Walsh functions at the source node vs. the efficiency, where  $N = 24$  nodes, number of Walsh functions at intermediate nodes is 3, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$ , maximum number of neighbors = 6, and the minimum number of neighbors = 2.

Figure 3.13 shows the effects of the number of Walsh functions at the source node on the system efficiency. From figure 3.13 we observe that the optimum number of Walsh functions at the source node that would maximize the system efficiency depends on the network connectivity.

### 3.5 Conclusion

This chapter studied the effects of RNC on the CDMA/TDD WMNs. Analysis and computer simulation results are presented. The study sheds light on the interaction between physical based error correction and CDMA parameters such as the Processing Gain and the number of Walsh functions on one side, and of network based coding parameters such as connectivity and field size on the other. Two scenarios were compared in terms of their efficiencies. Only scenario 1 applies RNC. For both scenarios, we have assumed the same number of assigned Walsh functions at all intermediate nodes. In scenario 1, if a certain node in the network has only one neighbor, then RNC will not improve the overall system performance. For a given set of parameters (number of assigned Walsh functions at the source node, number of assigned Walsh functions at intermediate nodes, Processing Gain, etc.), the efficiency of scenario 1 is maximized at a certain connectivity level. Increasing the Processing Gain dominates the effects of the high network connectivity. The effect of the packet size in scenario 1 on the system efficiency is more noticeable than in scenario 2. Also, in scenario 1, the optimum number of Walsh functions at the source node that would maximize the system efficiency was found to depend on the network connectivity. The delay and delay jitter in scenario 1 are higher than in scenario 2. In short, the possible improvement in the system performance depends on the minimum number of neighbors in the network, the network connectivity, the Processing Gain, the packet size, etc. Thus, a cross layer design between the network layer (topology) and the data link layer (number of assigned Walsh functions) at each node is needed to further improve the overall system performance.

## **Chapter 4 CROSS LAYER DESIGN FOR RANDOM NETWORK CODING CDMA/TDD WIRELESS MESH NETWORKS**

### **4.1 Introduction**

In chapter three an exhaustive study of the effects of RNC on the CDMA/TDD WMNs was investigated. In chapter three, all nodes in the WMNs use a certain number of Walsh functions during the file transfer. This number of assigned Walsh functions is determined independently of any criteria, such as number of neighbors of the node, the identities of unacknowledged packets, etc. Thus, the knowledge of the number of neighbors is not utilized for Walsh function assignment at a certain subject node. For example, if a node has many neighbors and all its neighbors use a certain number of Walsh functions. The SNIR of the received packets from all neighbors could be very low and thus, this could lead to a degraded performance for highly connected network as evidenced by the results of chapter 3. Therefore, the number of assigned Walsh functions at all nodes has a high impact on the system performance. This chapter presents some strategies for Walsh functions assignments in RNC-CDMA/TDD WMNs.

WMNs have many unique properties, such as self-configuration, self-organization [66], etc. The need of CLD for WMNs has been studied in [67]. In [52] effects of RNC on CDMA/TDD platforms are studied and evaluated. The authors of [68] have proposed a cross layer optimization framework for WMNs with NC. On the other hand this chapter investigates the effects of the cross layering between network and data link layers for the RNC- CDMA/TDD platforms. In this chapter, the adjustments of the number of Walsh

functions at all nodes are the main apparatus for cross layer design. Different Walsh function assignments strategies at nodes are presented and tested. A multi-hop mesh network with a single source and multiple receiving nodes is assumed. For reliable data transfer, a Selective Repeat ARQ protocol is used. Two scenarios are evaluated for their efficiency. In scenario 1, RNC is applied to CDMA/TDD system. In scenario 2 RNC is not utilized. The delay and delay jitter for both scenarios are computed.

The outline of this chapter is as follows. In section 4.2, the system model and the Walsh assignment are presented. Simulation results are presented in section 4.3. The conclusion is in section 4.4.

## **4.2 System Model**

As in chapter three, a WMN with a single source node (s) which has a file of length ( $L$ ) packets to be sent to all of the nodes in the network is considered, as shown in figure 3.1. For more detail about the system model refer to section 3.3 in the third chapter. In this chapter, various nodes utilize in a cross layer fashion the information about network topology contained within the network layer. This information is used by the data link layer which then adjusts the number of Walsh functions depending on such information received from network layer. Three strategies for the number of the assigned Walsh functions at the data link layer of the source node are compared. The three strategies can be explained as follows:



#### **4.2.1. The source fixed assignment:**

The source fixed assignment was presented in [52], where the number of assigned Walsh functions at the source node is chosen independently of network topology and is fixed. Accordingly, in the source fixed assignment no CLD is implemented.

#### **4.2.2. The source worst case assignment:**

Hello messages aid the source node to know the number of neighbors of each neighbor which implies a possible cross layer mechanism i.e. information obtained by Hello messages at the network layer will be used to adjust the number of Walsh functions and hence improve the overall performance. The source node determines the number of its assigned Walsh functions depending on the neighbor with the maximum number of neighbors, which we call the worst case. The source node chooses the number of its assigned Walsh functions so as to guarantee that the probability of packet success of its transmitted signal at a neighbor having the maximum number of neighbors is above a given threshold. The exact mechanism for such guarantees will follow shortly.

#### **4.2.3. The source average assignment:**

Here we follow the same Hello messages and inherent cross layer as above. However, now the source node computes the average number of its neighbor's neighbors excluding neighbors with a single neighbor. Then the source node chooses the number of its assigned Walsh functions so as to guarantee that the probability of packet success of its transmitted signal at a neighbor having the average number of the neighbors' neighbors is above a given threshold.

At the intermediate nodes, three strategies to determine the number of assigned Walsh functions are compared. The three strategies can be explained as follows:

#### **4.2.4. The intermediate fixed assignment:**

The intermediate fixed assignment was presented in [52], where the number of assigned Walsh functions at all intermediate nodes is chosen independently on the network topology and is fixed, again this is not cross layer design.

#### **4.2.5. The intermediate worst case assignment:**

From Hello messages and inherent cross layer mechanism, the intermediate nodes know the number of neighbors of each of their neighbors. An intermediate node determines the number of its assigned Walsh functions depending on the neighbor with the maximum number of neighbors, which we call the worst case (similar to source node strategy above). The intermediate node chooses the number of its assigned Walsh functions which guarantees that the probability of packet success of its transmitted signal at the neighbor with the maximum number of neighbors is above a given threshold. The mechanism for such guarantee will be detailed shortly.

#### **4.2.6. The intermediate dynamic assignment:**

Figure 4.1 is the flow chart of the intermediate dynamic assignment. Again Hello messages help the intermediate node to know the number of neighbors of each of their neighbors. The intermediate node initially determines the number of its assigned Walsh functions depending on the neighbor with the maximum number of neighbors, which we call the worst case. The intermediate node initially chooses the number of its assigned

Walsh functions which guarantees that the probability of packet success of its transmitted signal at the neighbor with the maximum number of neighbors is above a given threshold. The intermediate node schedules an NCP to be transmitted. If the initial number of assigned Walsh functions at the transmitting node is greater than the number of assigned Walsh functions at the source node, then the initial number of assigned Walsh functions at the transmitting node is changed to be equal to the number of assigned Walsh functions at the source node. If one or more neighbors of the transmitting intermediate node have only one neighbor and it is the first time that the transmitting intermediate node transmits this NCP, then the number of the assigned Walsh functions at the transmitting intermediate node will be chosen to be equal to the number of the independent images of the transmitted NCP. The reason is to help the neighbor with a single neighbor as much as possible.

On the other hand, if the following conditions are true

- one or more neighbors of the transmitting intermediate node have only one neighbor
- it is not the first time that the transmitting intermediate node transmits this NCP and the number of the independent images of the scheduled NCP at the transmitting node is less than the initially assigned number at the transmitting node,

Then the number of the assigned Walsh functions at the transmitting intermediate node will be determined to be equal to the number of independent images of the scheduled NCP at the transmitting intermediate node.

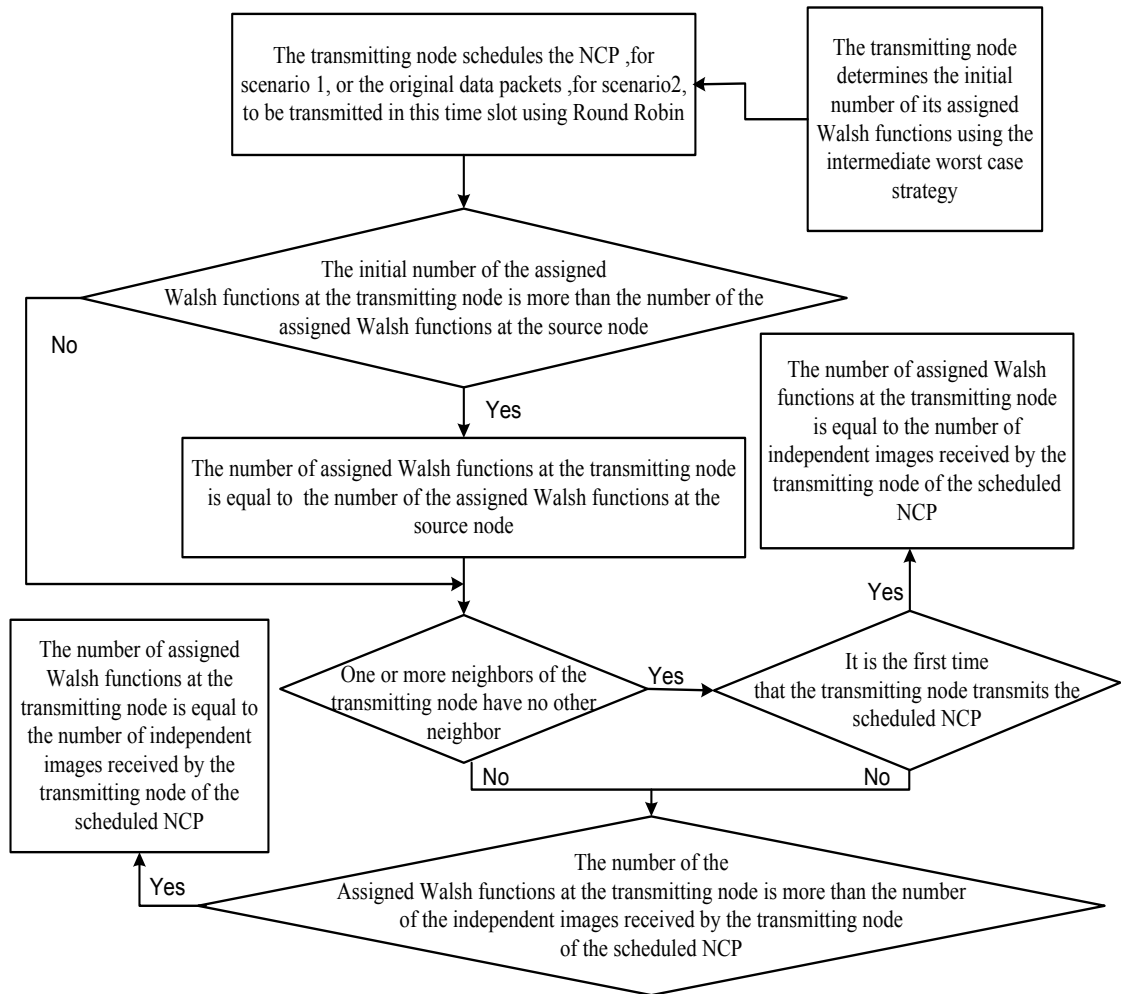


Figure 4.1 The flow chart of the intermediate dynamic assignment presented in section 4.2.6.

However, if the following conditions are true

- one or more neighbors of the transmitting intermediate node have only one neighbor
- it is not the first time that the transmitting intermediate node transmits this NCP

- the number of the independent images of the scheduled NCP at the transmitting intermediate node is more than the number of the initially assigned Walsh functions at the transmitting intermediate node

Then the number of the assigned Walsh functions at the transmitting intermediate node will be chosen to be equal to the initially assigned number (i.e. function of maximum number of neighbors). If all neighbors of the transmitting intermediate node have two or more neighbors and the number of the independent images of the scheduled NCP is less than the initially assigned number at the transmitting node, then the number of the assigned Walsh functions at the transmitting intermediate node will be determined to be equal to the number of independent images of the scheduled NCP at the transmitting intermediate node.

In this chapter, the Walsh function assignments presented above are applied to the two scenarios (using and not using RNC) presented in sections 3.2.1 and 3.2.2. Considering the two possible scenarios and the aforementioned Walsh function assignments for the source and the intermediate nodes one may get at least 12 design cases. The results section will discuss some of these.

By using 3.14 and the same selected channel coding in third chapter, the bit error rate ( $P_b$ ) can be computed (using turbo coding,  $P_b$  on page 252, figure 7.32 in [65]). Assuming  $n_b$  independent bits per packet, the probability of packet success ( $P_c$ ) is:

$$P_c = (1 - P_b)^{n_b} \quad (4.1)$$

As discussed in the Walsh functions assignment strategies for source node and intermediate nodes above, the number of Walsh functions at intermediate nodes is not the

same. Moreover, the number of assigned Walsh functions at a node could vary from one transmission time slot to another, as explained in 4.2.6. Thus, the smallest number of assigned Walsh functions at any intermediate node in the network is considered for the system efficiency computation. For scenario 1, let us denote the total number of nodes in the network as  $N$  and the number of time slots required until all nodes receive the entire file ( $L$  original data packets) and all nodes receive acknowledgment signals for all  $L$  original packets from all of their neighbors as  $F$  (found from the simulation results).

The system efficiency of scenario 1 ( $\eta_{NC}$ ) is:

$$\eta_{NC} = \left( \frac{2 \times \left( \frac{L}{\min(g_k)} \right)}{F} \right) \times \left( \frac{N-1}{P_G} \right) \times \left( \frac{n_b - g_s \times m}{n_b} \right) \quad (4.2)$$

Where  $g_s$  is number of the assigned Walsh functions at the source node,  $P_G$  is the Processing Gain and  $m$  is the GF size. The first part of (4.2) presents the RNC effects, the second part of (4.2) is the typical CDMA transmission efficiency and the third part of (4.2) is the effects of the RNC header. For scenario 2, let us denote the number of time slots required until all nodes receive the entire file ( $L$  original data packets) and all nodes receive acknowledgment signals for all  $L$  original packets from all of their neighbors as  $F'$  (found from the simulation results). The system efficiency for scenario 2 ( $\eta_{NNC}$ ) is:

$$\eta_{NNC} = \left( \frac{2 \times \left( \frac{L}{\min(g_k)} \right)}{F'} \right) \times \left( \frac{N-1}{P_G} \right) \quad (4.3)$$

The difference between (4.2) and (4.3) is the effect of the RNC header. We assume a real time data transfer application. The delay of the  $n^{th}$  original data packet at node  $x$  for a

certain generated topology  $H$  is defined as the difference between the time slot when node  $x$  has received the original data packet ( $T_{n,x,H}$ ) and the time slot when the  $n^{\text{th}}$  original data packet should have been transmitted from the source node ( $T_{n,H}$ ). The delay of the  $n^{\text{th}}$  original data packet at node  $x$  for topology  $H$  is:

$$D_{n,x,H} = T_{n,x,H} - T_{n,H} \quad (4.4)$$

The average delay of the original data packet  $n$  averaged over all intermediate nodes for topology  $H$  is:

$$D_{n,H} = \frac{\sum_{x=1}^{N-1} D_{n,x,H}}{N-1} \quad (4.5)$$

The average delay of all packets at all intermediate nodes for topology  $H$  is:

$$D_H = \frac{\sum_{n=1}^L D_{n,H}}{L} \quad (4.6)$$

Denote the number of generated topologies as  $\delta$ . The average delay of all packets at all nodes for all topologies is:

$$D = \frac{\sum_{H=1}^{\delta} D_H}{\delta} \quad (4.7)$$

The delay jitter between packets  $n$  and  $n+1$  at node  $x$  for topology  $H$  can be presented as follows:

$$\bar{J}_{n,x,H} = |D_{n+1,x,H} - D_{n,x,H}| \quad (4.8)$$

The delay jitter for the data file at node  $x$  for topology  $H$  is:

$$\bar{J}_{x,H} = \sum_{n=1}^{L-1} \frac{\bar{J}_{n,x,H}}{L-1} \quad (4.9)$$

The delay jitter is:

$$\overline{\sigma}_{x,H} = \sqrt{\frac{\sum_{n=1}^{L-1} (\bar{J}_{x,H} - \bar{J}_{n,x,H})^2}{L-2}} \quad (4.10)$$

The average delay jitter is:

$$\sigma = \frac{\sum_{x=1}^{N-1} \sum_{H=1}^{\delta} \overline{\sigma}_{x,H}}{(N-1)\delta} \quad (4.11)$$

### 4.3 Simulation Results

In this section we show the effects of the Walsh functions assignments presented in section 4.2, on the RNC-CDMA/TDD WMNs performance through computer simulations. A comparison between the performances of the two scenarios (scenario 1 and scenario 2) and the different Walsh functions assignment strategies is introduced. To improve the physical layer, the probability of bit and hence packet transmission errors, turbo coding results [65] were used. In the simulation, without going into too much detail, a turbo code that uses identical BCH component code (BCH (31, 21, 5) & BCH (31, 21, 5)) with a 0.51 code rate was used [65]. Also, the assumed total used bandwidth (W) is 100MHz. The maximum possible number of neighbors of any node in the network is limited to avoid working in very low SNIR. To have accurate and general results in our



simulation, 50 different topologies for each given input parameter, such as network connectivity, file size, packet size, etc. were generated. As in chapter 3, uniform random variables are used to determine the number of nodes in each hop, the number of neighbors of each node, and the existing links in each topology. The value of  $F$  was computed by simulation for each topology. We used the SNIR in (3.14) and the curve in [65, pp. 252] to compute the standard  $P_b$ . Then we used (4.1) to compute  $P_c$ . Uniform random variates ( $\theta$ ) were used to determine if the transmitted packet was received correctly. For example, if the uniformly distributed random generator yields 0.8 for a packet, where  $P_c$ , the packet success probability from (4.1), is 0.9, then  $\theta < P_c$  and a packet success is recorded. Depending on which strategy of Walsh functions assignment is used, a node uses the maximum number of Walsh functions that guarantee the probability of packet success of its transmitted signal to be above a given certain threshold. In the simulation results, the threshold was 95%. For example, if the node uses the intermediate worst case assignment presented in 4.2.5, then the node will choose the maximum number of Walsh functions that guarantee the probability of packet success of its transmitted signal to neighbor with maximum number of neighbors is above 95%. The randomly generated coefficients for NC were chosen from GF ( $2^6$ ) and transmitted as a header in the NCP. Each coefficient adds 6 bits to the NCP size. The network coding header in the NCP is  $6 \times g_s$  bits.

For scenario 1, if a new image of an NCP is received at node  $d$ , we buffer the coefficients of the image with the coefficients of previously received images of the NCP. We then compute the rank of the coefficients of all the buffered images of the NCP. If the rank is equal to the number of buffered images, the last image is considered to be independent of

the previously buffered images, otherwise node  $d$  discards the last received image. If  $g_s$  or more independent images belonging to an NCP are buffered at node  $d$ , then we assume that node  $d$  is able to decode the NCP and get the original data packets, and so an acknowledgment signal is broadcast by node  $d$  in the next transmission time slot. Also, if the node has received enough images of an NCP ( $g_s$  or more) and is still receiving a new image of the NCP, the node piggybacks an acknowledgment signal again to make sure that all neighbors have been acknowledged. For scenario 1, when it is time for the intermediate node  $k$  to transmit (except those nodes that have received the constituting packets of the NCP), it selects one unacknowledged NCP depending on the scheduling algorithm (round robin). That intermediate node then generates  $g_k$  new images of the selected NCP using the buffered images of the NCP, as per equation (3.5). These  $g_k$  new images are then spread using the  $g_k$  assigned Walsh functions at the node. The average number of transmissions required for all of the generated topologies is then computed. To compute the system efficiency for the given set of parameters, we substitute  $F$ , i.e. the average required number of transmissions in (4.2) for scenario 1 and  $F'$  in (4.3) for scenario 2. Figure 4.2 shows that for lower connected networks, scenario 1 with the fixed Walsh function assignment at the source and the intermediate nodes provides the worst performances. This is because the low SNIR of the source coded packets at nodes in the first hop, due to the number of Walsh functions used at the source node, degrades system performance. Also, in scenario 1, a node piggybacks an acknowledgement signal for an NCP only if the node has received  $g_s$  or more independent images of the NCP, while in scenario 2, a node piggybacks an acknowledgement signal for each original data packet.

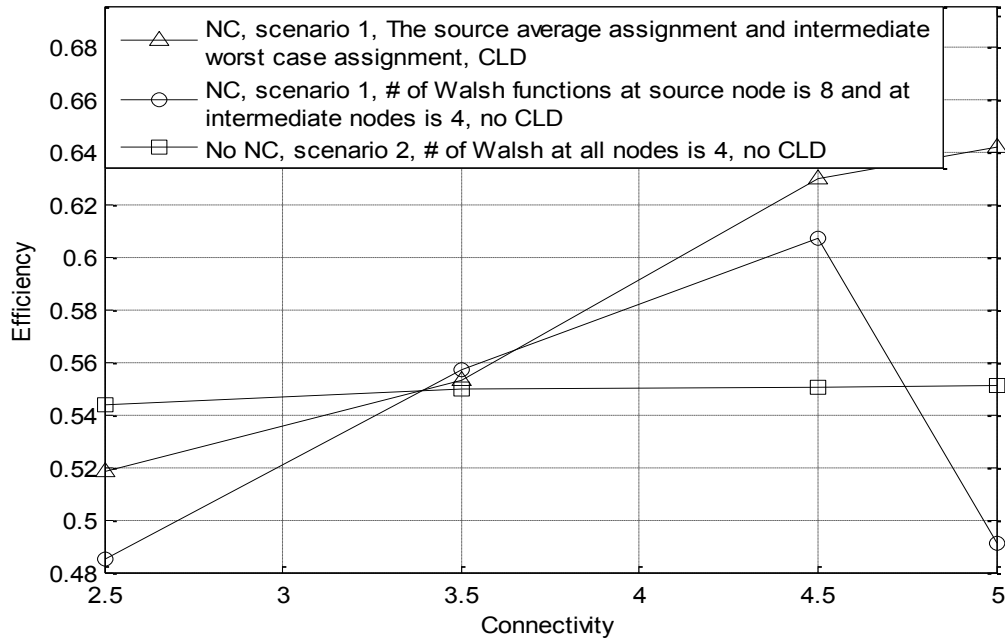


Figure 4.2 The efficiency of the scenarios vs. the network connectivity for some of the presented Walsh functions strategies.  $N = 20$  nodes number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$  and the maximum number of neighbors = 6.

As a result, in scenario 1, some nodes may receive images that depend on the previously received images of an NCP. Furthermore, the network coding header, which is  $6 \times g_s$  bits in each NCP, degrades the performance of scenario 1. Figure 4.2 shows that the efficiency of scenario 1 with fixed Walsh assignment is maximized at a certain connectivity level. From figure 4.2 we can observe that for network connectivity between 3.5 and 4, scenario 1 with fixed Walsh assignment provides a comparable performance compared with scenario 1 with cross layer design, thanks to the increase in the network connectivity. However, for scenario 1 with fixed Walsh assignment, increasing the network connectivity does not always lead to better performance, which can be seen in figure 4.2 for network connectivity = 5. Increasing the network connectivity leads to

increasing the number of the nodes' neighbors in the network (any node including nodes in the first hop). Increasing the number of neighbors of a node in the first hop leads to a lower SNIR of the source coded packets at that node.

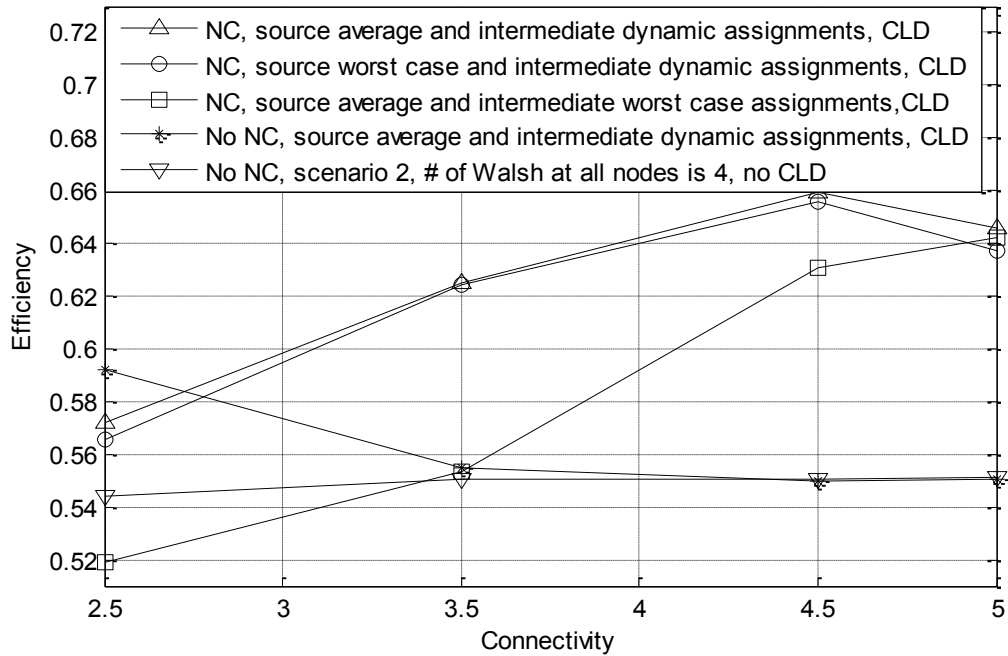


Figure 4.3 The efficiency of the systems vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ ,  $w_s = 4$ ,  $w_r = 4$  and maximum number of neighbors = 6.

Consequently, fewer nodes in the first hop receive the source coded packet and thus more transmissions from the source node are required, dragging down system performance. On the other hand, scenario 1 with cross layer design, the system performance does not decrease for high connected networks, as shown in figure 4.2 for network connectivity = 5, since the number of assigned Walsh functions is adjusted depending on the topology. Figure 4.3 shows the effects of the network connectivity on the efficiency of the systems.

Figure 4.3 shows that for lower connected networks, scenario 2 with the source average assignment and the intermediate dynamic assignment provides the best performance. This is because the network coding header, which is  $6 \times g_s$  bits in each NCP, degrades the performance of the scenario 1 for the same Walsh function assignment. Also, figure 4.3 shows that for lower connected networks, scenario 1 with the source average Walsh assignment and the worst case intermediate Walsh assignment provides the worst performances compared with the others that are presented in figure 4.3. Figure 4.4 shows the relation between the network connectivity and the delay for some Walsh function assignments.

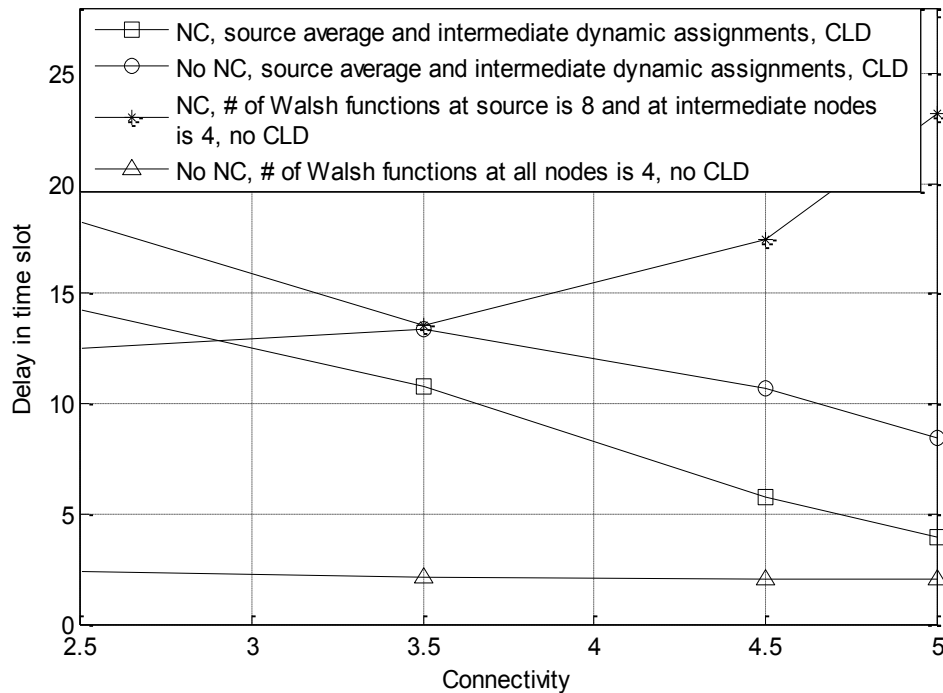


Figure 4.4 The delay vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ , and maximum number of neighbors = 6.

From figure 4.4, we note that the delay for scenario 1 with CLD (the source average Walsh assignment and the intermediate dynamic assignment) is less than the delay for scenario 1 without CLD (fixed assignment) and the delay for scenario 2 with CLD (the source average Walsh assignment and the intermediate dynamic assignment) schemes that are presented in figure 4.4 for some network connectivity (connectivity = 3.5 to 5). The higher the network connectivity the more neighbors a node can hear. As a result, for higher network connectivity a node receives from more neighbors per each time slot. Because of that, the node can decode an NCP within less number of time slots, less delay, compared with lower connected network(connectivity = 2.5).

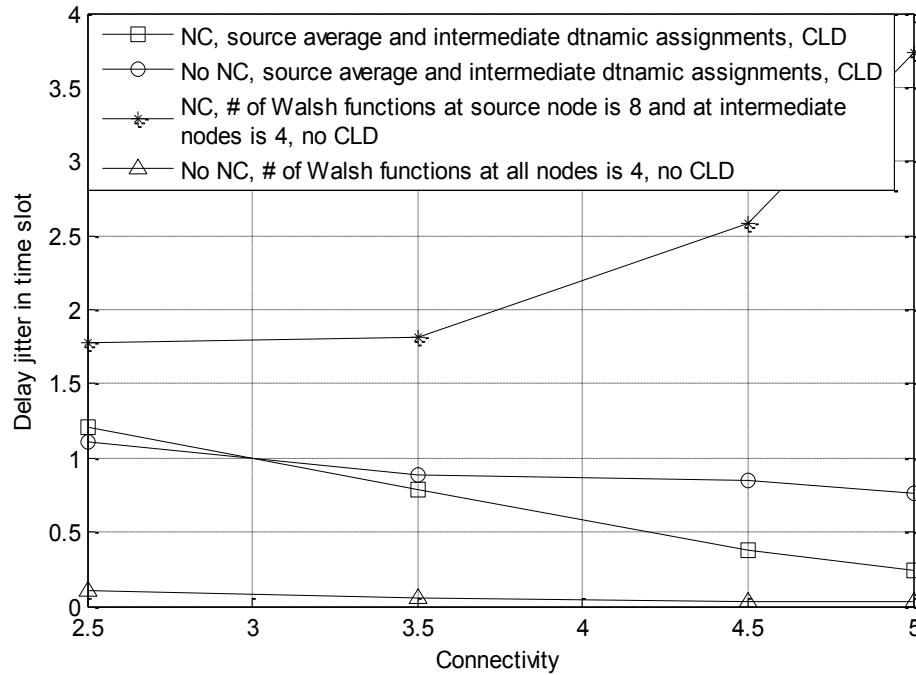


Figure 4.5 The delay jitter vs. the network connectivity for some of the presented Walsh functions strategies, where  $N = 20$  nodes, number of hops = 5,  $L = 120$  packets,  $n_b = 1000$  bits,  $P_G = 32$ ,  $1/SNR_{th} = 0.01$ ,  $m = 6$ , and the maximum number of neighbors=6.

Figure 4.5 shows the relation between the network connectivity and the delay jitter for some Walsh function assignments. In scenario 2, all nodes send the original data packets, no NC is applied. For that reason, figure 4.5 shows that scenario 2 has the smallest delay jitter. From figure 4.5 we observe that the delay jitter for scenario 1 with CLD is better than scenario 1 with fixed Walsh function assignment. CLD makes the delay jitter values comparable to those obtained in scenario 2 (no network coding).

#### **4.4 Conclusion**

This chapter studied the effects of the cross layer design (Walsh function assignments) on RNC-CDMA/TDD wireless mesh networks. Three different Walsh function assignments at the source node, including the fixed assignment presented in chapter 3, and three different Walsh function assignments at the intermediate nodes were presented and compared in terms of the achieved system efficiency, delay and delay jitter. Some computer simulation results were presented. Two scenarios were compared in terms of their efficiencies. Only scenario 1 applies RNC to CDMA/TDD WMN. For low connected networks, RNC may not help to improve the system efficiency, delay nor delay jitter. For scenario 1 with cross layer design the system performance, delay and delay jitter are better than scenario 1 without cross layer design. The suggested cross layer NC techniques necessitate the dynamic knowledge of the number of Walsh functions at each time slot and the consequent reconfiguration of transceiver. Such reconfiguration issues would be the subject of the near future research.

## Chapter 5 CONCLUSION

This thesis addresses the possible advantages of network coding, particularly to improve system performance for star-oriented and wireless mesh networks in certain applications, such as file sharing and video streaming.

Firstly, in this thesis a XOR based scheduling algorithm for star-oriented wireless networks for local repair was introduced. The proposed scheduling algorithm is for scenarios where BS (Wireless Wide Area Network (WWAN)) transmits a data file to nodes within its coverage area, and then a set of nodes that can hear each other directly use Wireless Local Area Network (WLAN) for local repair. The proposed scheduling algorithm commences in three phases. In the first phase, nodes exchange packet availability vectors. These vectors are functions of the probability of correct packet reception over the channel. This is followed by a short period of distributed scheduling where the nodes execute the processing algorithm which tries to minimize the total transmission time. In the third phase, nodes transmit the encoded packets as per the decision of the scheduling algorithm. Simulation and analysis results showed the possible improvement in the system performance for the proposed algorithm. Also a study of the trade-offs between file size, processing delay, number of users and packet availability were introduced. Also, the favorable effects of file segmentation on the performance of the proposed algorithm were presented.

Secondly, an exhaustive study about the effects of the random network coding on the CDMA/TDD wireless mesh networks was presented. A multi-hop wireless mesh



network with single source and multiple receiving nodes was assumed. For reliable data transfer, a Selective Repeat ARQ protocol was used. Two scenarios are evaluated for their efficiency. In scenario 1, random network coding is applied to CDMA/TDD system but not scenario 2. For both scenarios, the number of Walsh functions at each node is fixed. The delay and delay jitter for both scenarios are computed. The study also focuses on the effects of uncontrolled parameters such as the minimum number of neighbors and the network connectivity, and of controlled parameters such as the Galois Field (GF) size, the packet size and the Processing Gain. The analysis and simulation results show that applying random network coding to CDMA/TDD wireless mesh networks could provide a noticeable improvement in the overall efficiency. The possible improvement on the system efficiency for a given fixed number of Walsh functions at the nodes varies with the network connectivity. Moreover, delay and delay jitter for CDMA/TDD wireless mesh networks with random network coding for some given sets of parameters (network connectivity, the number of assigned Walsh functions at the source node, etc.) are higher than without using random network coding. In conclusion, to achieve high improvement in system efficiency and better delay and delay jitter, the number of assigned Walsh functions at the nodes should be chosen depending on the network topology.

Finally, a cross layer design for the Random Network Coded-Code Division Multiple Access/Time Division Duplex (RNC-CDMA/TDD) wireless mesh networks was presented. Nodes utilize the information about network topology contained in the network layer. This information is used by the data link layer to determine the number of assigned Walsh functions at each node. Two strategies of Walsh function assignments are proposed. In the first, nodes determine the number of their assigned Walsh functions

depending on the neighbor with the maximum number of neighbors, which we call the worst case assignments. In the second, nodes determine the number of their assigned Walsh functions depending on the need for each transmission. Simulation results showed the possible achievable improvement in the system performance, delay and delay jitter due to cross layer design.

## **5.1 Future Work**

In relation to this thesis, some suggested future works are as follows:

- In local repair, the proposed algorithm is for a group of nodes that can hear each other directly within the BS coverage area. This implies the clustering within the BS coverage area is required. Thus, internet-working among such clusters is one of the future works.
- Using FDD instead of TDD for CDMA/TDD with RNC in WMNs.
- To use Hybrid Automatic Repeat Request (HARQ) for reliable transmission instead of ARQ.
- The effects of the threshold values for Walsh functions assignments at nodes in WMNs on the system efficiency, and evaluation of the optimum value of the threshold in order to get the maximum efficiency.
- Extension of the NC concepts presented here to OFDM and MIMO systems

## References

- [1] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, pp. 1204-1216, 2000.
- [2] S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *Networking, IEEE/ACM Transactions on*, vol. 16, pp. 497-510, 2008.
- [3] J. Ben Saleh and A. K. Elhakeem, "A practical scheduling approach to network coding for wireless local repair," in *Communications (QBSC), 2010 25th Biennial Symposium on*, pp. 305-310.
- [4] N. Dong, T. Tuan, N. Thinh, and B. Bose, "Wireless Broadcast Using Network Coding," *Vehicular Technology, IEEE Transactions on*, vol. 58, pp. 914-925, 2009.
- [5] S. Y. R. Li, R. W. Yeung, and C. Ning, "Linear network coding," *Information Theory, IEEE Transactions on*, vol. 49, pp. 371-381, 2003.
- [6] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *Information Theory, IEEE Transactions on*, vol. 52, pp. 4413-4430, 2006.
- [7] Roger. E. Ziemer, Roger L. Peterson, and David E. Borth, *Introduction to spread-spectrum communications*: Prentice Hall, 1995.
- [8] C. Lee, R. Steele, and P. Gould, *GSM, cdmaOne and 3G Systems*, 2001.
- [9] R. Esmailzaddeh and. M. Nakagawa, *TDD-CDMA for wireless communications*: Artech House, 2003.

- [10] C. Kun-Nyeong, L. Ki-Dong, and K. Dongwoo, "Optimal timeslot and channel allocation considering fairness for multicell CDMA/TDD systems." vol. 33: Elsevier Science Ltd., 2006, pp. 3203-3218.
- [11] J. Dong Guen and J. Wha Sook, "Capacity of CDMA/TDD systems for mobile multimedia services," in *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, 1999, pp. 2606-2610 vol.5.
- [12] R. Esmailzadeh, M. Nakagawa, and A. Jones, "TDD-CDMA for the 4th generation of wireless communications," *Wireless Communications, IEEE*, vol. 10, pp. 8-15, 2003.
- [13] I. Al-wazedi and A. K. Elhakeem, "A CDMA/TDD approach for wireless mesh networks," *Computers & Electrical Engineering*, vol. 36, pp. 978-992.
- [14] A. Leon-Garcia and I. Widjaja, *Communication Networks* Second Edition ed.: Elizabeth A. Jones, 2004.
- [15] R. Koetter and M. Medard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on*, vol. 11, pp. 782-795, 2003.
- [16] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *Information Theory, IEEE Transactions on*, vol. 51, pp. 1973-1982, 2005.
- [17] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Information Theory, 2003. Proceedings. IEEE International Symposium on*, 2003, p. 442.

- [18] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proceeding of 41st Annual Allerton Conference on Communication, Control and Computing*, 2003.
- [19] H. Yomo and P. Popovski, "Opportunistic Scheduling for Wireless Network Coding," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 5610-5615.
- [20] H. Yomo and P. Popovski, "Opportunistic scheduling for wireless network coding," *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 2766-2770, 2009.
- [21] D. K. S. Katti, W. Hu, H. Rahul, M. Medard, "The importance of being opportunistic: Practical Network Coding for Wireless Environments," in *Proc. 43rd Annual Allerton Conference on Communication, Control and Computing*, 2005.
- [22] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network Coding-Aware Routing in Wireless Networks," *Networking, IEEE/ACM Transactions on*, vol. 18, pp. 1158-1170.
- [23] C. Kaikai, J. Xiaohong, and S. Horiguchi, "Network Coding Opportunity Analysis of COPE in Multihop Wireless Networks," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2008, pp. 2858-2863.
- [24] K. Chi, X. Jiang, and S. Horiguchi, "A More Efficient COPE Architecture for Network Coding in Multihop Wireless Networks," *IEICE Transactions on Communications*, vol. E92.B, pp. 766-775, 2009.

- [25] S. Omiwade, Z. Rong, and H. Cunqing, "Practical Localized Network Coding in Wireless Mesh Networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, 2008, pp. 332-340.
- [26] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network Coding-Based Broadcast in Mobile Ad-hoc Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1739-1747.
- [27] L. Jilin, J. Lui, and C. Dah Ming, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," in *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, 2008, pp. 462-469.
- [28] Y. Yan, Z. Baoxian, H. T. Mouftah, and M. Jian, "Practical Coding-Aware Mechanism for Opportunistic Routing in Wireless Mesh Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2871-2876.
- [29] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *Proc. ACM SIGCOMM*, 2007.
- [30] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 27, pp. 622-634, 2009.

- [31] Z. Jingyao, F. Pingyi, and K. B. Letaief, "Network Coding with Low Complexity in Wireless Ad-hoc Multicast Networks," in *Communications, 2006. ICC '06. IEEE International Conference on*, 2006, pp. 3699-3704.
- [32] Z. Jingyao, F. Pingyi, and K. Ben Letaief, "Network Coding for Efficient Multicast Routing in Wireless Ad-hoc Networks," *Communications, IEEE Transactions on*, vol. 56, pp. 598-607, 2008.
- [33] N. Bin, N. Santhapuri, C. Gray, and S. Nelakuditi, "Selection of Bit-Rate for Wireless Network Coding," in *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008. SECON Workshops '08. 5th IEEE Annual Communications Society Conference on*, 2008, pp. 1-6.
- [34] S. Zhang, S. C. Liew, and P. P. Lam, "Physical Layer Network Coding," in *Proc. of ACM MOBICOM 2006, Los Angeles, USA*, 2006.
- [35] K. Sachin, G. Shyamnath, and K. Dina, "Embracing wireless interference: analog network coding," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications* Kyoto, Japan: ACM, 2007.
- [36] Y. E. Sagduyu and A. Ephremides, "Crosslayer design for distributed MAC and network coding in wireless ad hoc networks," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, 2005, pp. 1863-1867.
- [37] Y. E. Sagduyu and A. Ephremides, "On Joint MAC and Network Coding in Wireless Ad Hoc Networks," *Information Theory, IEEE Transactions on*, vol. 53, pp. 3697-3713, 2007.

- [38] A. Khreishah, W. Chih-chun, and N. Shroff, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 27, pp. 606-621, 2009.
- [39] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "On MAC Scheduling and Packet Combination Strategies for Practical Random Network Coding," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 3582-3589.
- [40] L. Junning, D. Goeckel, and D. Towsley, "The Throughput Order of Ad Hoc Networks Employing Network Coding and Broadcasting," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, 2006, pp. 1-7.
- [41] L. Junning, D. Goeckel, and D. Towsley, "Bounds on the Gain of Network Coding and Broadcasting in Wireless Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 724-732.
- [42] A. Keshavarz-Haddadt and R. Riedi, "Bounds on the Benefit of Network Coding: Throughput and Energy Saving in Wireless Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 376-384.
- [43] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the throughput gain of network coding in unicast and multicast wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 27, pp. 582-592, 2009.
- [44] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, pp. 388-404, 2000.



- [45] L. Xin, S. Raza, C. Chen-Nee, and C. Gene, "Network Coding Based Cooperative Peer-to-Peer Repair in Wireless Ad-Hoc Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2153-2158.
- [46] L. Xin, G. Cheung, and C. Chen-Nee, "Rate-distortion optimized network coding for cooperative video stream repair in wireless peer-to-peer networks," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, 2008, pp. 1-6.
- [47] L. Xin, C. Gene, and C. Chen-Nee, "Structured network coding and cooperative local peer-to-peer repair for MBMS video streaming," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, 2008, pp. 456-461.
- [48] K. Nguyen, N. Thinh, and C. Sen-ching, "Peer-to-Peer Streaming with Hierarchical Network Coding," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 396-399.
- [49] L. Xin, G. Cheung, and C. Chen-Nee, "Structured Network Coding and Cooperative Wireless Ad-Hoc Peer-to-Peer Repair for WWAN Video Broadcast," *Multimedia, IEEE Transactions on*, vol. 11, pp. 730-741, 2009.
- [50] D. Qiu, j. Ben Saleh, and A. K. Elhakeem, "Performance of an Efficient Scheduling Approach to Network Coding For Wireless Local Repair," *Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT)*, pp. 49 - 58, 2011.
- [51] A. K. Elhakeem and J. Ben Saleh, "Analysis of a Practical Scheduling Approach to Network Coding for Wireless Local Repair," in *International Conference on computer, Electrical, and Systems Science, and Engineering*, 2010.

- [52] J. Ben Saleh and A. K. Elhakeem, "Random Network Coding Effects on CDMA/TDD Mesh Networks," in *Communication Networks and Services Research Conference (CNSR), 2011 Ninth Annual*, pp. 224-230.
- [53] S. Lin and D. J. Costello, *Error Control Coding :Fundamentals and Applications*, Second Edition ed.: Prentice Hall 2004.
- [54] T. Tuan, N. Thinh, and B. Bose, "A Joint Network-Channel Coding Technique for Single-Hop Wireless Networks," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, 2008, pp. 1-6.
- [55] S. Raza, L. Danjue, C. Chen-Nee, and G. Cheung, "Cooperative Peer-to-Peer Repair for Wireless Multimedia Broadcast," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 1075-1078.
- [56] N. Bin, N. Santhapuri, Z. Zifei, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, 2006, pp. 157-159.
- [57] D. Liao and A. K. Elhakeem, "Cross-layer optimization for CDMA/TDD wireless mesh networks," *Wireless Communications and Mobile Computing*, pp.1331-1340.
- [58] E. Fasolo, F. Rossetto, and M. Zorzi, "Network Coding meets MIMO," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, 2008, pp. 1-6.
- [59] D. E. Lucani, M. Medard, M. Stojanovic, and D. R. Karger, "Sharing information in time-division duplexing channels: A network coding approach," in

- Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, 2009, pp. 1403-1410.
- [60] D. E. Lucani, M. Medard, and M. Stojanovic, "Broadcasting in time-division duplexing: A random linear network coding approach," in *Network Coding, Theory, and Applications, 2009. NetCod '09. Workshop on*, 2009, pp. 62-67.
- [61] D. E. Lucani, M. Stojanovic, and M. Medard, "Random Linear Network Coding For Time Division Duplexing: When To Stop Talking And Start Listening," in *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, 2009, pp. 1800-1808.
- [62] Z. Xinyu and L. Baochun, "Joint Network Coding and Subcarrier Assignment in OFDMA-Based Wireless Networks," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, 2008, pp. 1-6.
- [63] L. Baochun and N. Di, "Random Network Coding in Peer-to-Peer Networks: From Theory to Practice," *Proceedings of the IEEE*, vol. 99, pp. 513-523.
- [64] J. Geier, *Wireless LANs Implementing Interoperable Networks*, 1st ed.: Jim Le Valley, 1999.
- [65] T. H. Liew, B. L. Yeap, and L. Hanzo, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*, 2002.
- [66] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445-487, 2005.
- [67] I. F. Akyildiz and W. Xudong, "Cross-Layer Design in Wireless Mesh Networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 1061-1076, 2008.

- [68] L. Kai and W. Xiaodong, "Cross-Layer Design of Wireless Mesh Networks with Network Coding," *Mobile Computing, IEEE Transactions on*, vol. 7, pp. 1363-1373, 2008.